HARNESSING USER GENERATED MULTIMEDIA CONTENT IN THE CREATION OF COLLABORATIVE CLASSIFICATION STRUCTURES AND RETRIEVAL LEARNING GAMES

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Otto Jerome Borchert

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Computer Science

July 2015

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Harnessing User Generated Multimedia Content in the Creation of Collaborative Classification Structures and Retrieval Learning Games

By

Otto Jerome Borchert

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Brian M. Slator
Chair

Anne Denton

Simone Ludwig

Erika Offerdahl

Approved:

9/2/2015
Date

Brian M. Slator
Department Chair

ABSTRACT

This paper describes a software tool to assist groups of people in the classification and identification of real world objects called the Classification, Identification, and Retrieval-based Collaborative Learning Environment (CIRCLE). A thorough literature review identified current pedagogical theories that were synthesized into a series of five tasks: gathering, elaboration, classification, identification, and reinforcement through game play. This approach is detailed as part of an included peer reviewed paper. Motivation is increased through the use of formative and summative gamification; getting points completing important portions of the tasks and playing retrieval learning based games, respectively, which is also included as a peer-reviewed conference proceedings paper. Collaboration is integrated into the experience through specific tasks and communication mediums.

Implementation focused on a REST-based client-server architecture. The client is a series of web-based interfaces to complete each of the tasks, support formal classroom interaction through faculty accounts and student tracking, and a module for peers to help each other. The server, developed using an in-house JavaMOO platform, stores relevant project data and serves data through a series of messages implemented as a JavaScript Object Notation Application Programming Interface (JSON API).

Through a series of two beta tests and two experiments, it was discovered the second, elaboration, task requires considerable support. While students were able to properly suggest experiments and make observations, the subtask involving cleaning the data for use in CIRCLE required extra support. When supplied with more structured data, students were enthusiastic about the classification and identification tasks, showing marked improvement in usability scores

and in open ended survey responses. CIRCLE tracks a variety of educationally relevant variables, facilitating support for instructors and researchers.

Future work will revolve around material development, software refinement, and theory building. Curricula, lesson plans, instructional materials need to be created to seamlessly integrate CIRCLE in a variety of courses. Further refinement of the software will focus on improving the elaboration interface and developing further game templates to add to the motivation and retrieval learning aspects of the software. Data gathered from CIRCLE experiments can be used to develop and strengthen theories on teaching and learning.

ACKNOWLEDGEMENTS

At the end of a fourteen year journey, it would be nearly impossible to thank every single person who has helped me get to this point. My biggest supporter has been my advisor: Dr. Brian Slator. His advice, editing, and guidance has been invaluable. In addition, my office mate, Guy Hokanson has been there every step of the way, testing software that wasn't quite perfect, offering suggestions for improvements that I may not have listened to right away even though I should have. Alekya Peruri developed the site map in CIRCLE and summarized student feedback from experiment one which led to a successful second experiment. Her support was greatly appreciated.

I'm grateful for the members of my committee: Brian Slator, Anne Denton, Simone Ludwig, and Erika Offerdahl. Their feedback throughout the process, especially during the preliminary defense guided the implementation and creation of this document.

I would also like to thank everyone in Journal Club. I would have never accomplished this without their support, feedback, and willingness to be my sounding board, guinea pigs, and exerting the informal social pressure to complete the dissertation. I'd especially like to thank Jennifer Momsen, for letting me use CIRCLE in her classroom and for testing four different versions of the software. Her feedback is present throughout the current prototype.

As odd as it seems, I'd also like to thank the professors who eventually didn't use CIRCLE in their classrooms too. They were both professional and gave feedback about initial bugs in to the software and the motivations of students. Sometimes the best information comes from your critics.

I would be remiss if I didn't thank the members of the World Wide Web Instructional Committee, an ad-hoc group of faculty, staff, and graduate students dedicated to the development

of immersive virtual environments for education. Their guidance, advice, and employment during my years as an undergraduate and graduate student at NDSU made this all possible.

I'd also like to thank the system administrators and administrative assistants of the Computer Science department and Center for Science and Mathematics Education: Nate, Robert, Ruth Ann, Stephanie, Carole, and everybody else. Their help through bureaucracy, server setup, form filling, and making sure I got paid were invaluable.

Other individuals have made an impact through the whole process that I'd like to thank: Harold "Cliff" Chaput, both for his comments while I was first formulating CIRCLE and for Dr. Slator's copy of his dissertation, which gave me strength during some rough patches. My friends, especially Josh, Melissa, and Heath, who have had to deal with my stress and being told no pretty consistently for almost three years. And finally, my family, who has always been there throughout the good and bad times in my life, being everything from a moral support to a good comedian to faith counselor. I love you all.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

INTRODUCTION

The Classification, Identification, and Retrieval-based Collaborative Learning Environment (CIRCLE) is an online, web-based application where users complete a series of steps to collaboratively classify and identify real world objects of their own choosing. Users referred to as gatherers begin by using their mobile devices to gather specimens: taking pictures and entering short descriptions. Their mobile devices then tag these specimens with their current location. Other users in the gatherer's group suggest experiments or observations to perform on the specimen. These experiments and observations are used to construct a classification structure. Each branch of the structure corresponds to a particular experimental or observational result. Paths on the tree define sets of experiments and observations and their results. At the end of these paths, at the leaves, are the specimens that embody these results. These classification structures inform student identification of the specimens, allowing them to distinguish between sets of experimental and observational results. Finally, the classification structures and positive identifications are used to automatically create games that reinforce student understanding.

CIRCLE offers a number of improvements over current state of the art. It implements a series of collaborative online tasks for identifying and classifying real world objects that are typically done by hand; applies formative and summative gamification through motivating constructivist user-generated content in contrast to flash cards or other memorization techniques, and provides an interface for instructors and researchers to view student progress and identify relevant variables associated with improving identification and classification outcomes.

Theoretical Background

This algorithm for teaching and learning taxonomy was reached through an exhaustive literature review, identifying current theories on how students learn. STEM educators seek to develop self-regulated learners, that is, learners who can create goals, are motivated to complete those goals, and know what they do and do not know (Zimmerman 1986, 1990). CIRCLE asks students to create their own goals when they choose what kinds of specimens they want to identify and motivate students through the power of game-based learning.

Individuals are motivated through two major modalities. Extrinsic motivation refers to external factors that encourage students to learn, through the use of rewards or punishments. Intrinsic motivation refers to internal factors that encourage students to learn, like the joy of learning or natural interest in the topic at hand. CIRCLE uses both of these methods. Students are intrinsically motivated by collecting, classifying, and identifying specimens of their own choosing and by playing entertaining games based on the information they collected. They gain extrinsic motivation through points given for completing tasks.

Current identification games ask students to memorize a set of specimens and their identities (App Shopper 2013; and others, see the literature review). CIRCLE builds on these methods through summative gamification - the playing of a game at the end of the series of tasks (Borchert, Hokanson, Peruri & Slator 2015). These games are generated by software based on the content gathered by the users. CIRCLE also implements gamification as described by (Deterding, Dixon, Khaled & Nacke 2011), which we call formative gamification, where points are given for completing tasks that are important to their goal of identifying and classifying objects.

After students have been motivated to learn, various techniques assist students in building their knowledge. The theory of Constructivism hypothesizes that students create their own knowledge based on prior understanding as opposed to simply being empty vessels in which knowledge is poured (Palincsar 1998). CIRCLE users build classification structures rather than memorizing them, an approach often referred to as “Learning by Doing”.

CIRCLE also implements state of the art in collaborative learning theory. Research has shown there are five primary factors correlated to group performance: positive interdependence, individual accountability, promotive interaction, social skills, and group processing (Johnson & Johnson 1989; Johnson, Johnson, & Holubec 1998). CIRCLE expresses these factors through a variety of means including real time chat, bulletin boards, role enforcement, a combination of developer and user specified tasks, and formative gamification.

Retrieval learning is a new theory in educational practice that postulates that students learn best when they attempt to retrieve knowledge as opposed to reviewing material multiple times (Karpicke 2012). Summative gamification supports retrieval learning. Rather than simply memorizing their constructed classification structure, students must retrieve and use this knowledge through the course of playing a game. For CIRCLE, this game involves students identifying the experiments needed to identify collected specimens.

These learning theories are cemented through computer science visualization techniques. CIRCLE experiences included two separate graphing solutions: D3.js and GoJS. D3.js was eventually replaced with GoJS for classification structure visualization, but D3.js continues to be used for statistics visualization. GoJS was also used in the creation of a visual site map, where nodes in the graph corresponded to pages within the CIRCLE interface that users could visit (Peruri, in preparation).

Implementation

CIRCLE is implemented as a client-server application with a series of messages passed between them in a Representative State Transfer (REST) protocol. The client, written in

Javascript, shows the user information related to their projects. The JavaMOO server stores information related to all users. JavaScript Object Notation (JSON) is used to communicate between the client and server.

The client has five main functional areas: Account Creation and Access, Project Setup and Selection, Project Interaction, Peer Help, and Instructor Pages. Account Creation and Access allows students and faculty to create accounts and connect to the game. Project Setup and Selection lets users create projects, invite users, and select a project to work on. Project Interaction allows students to complete the five main CIRCLE tasks: gathering, elaborating, classifying, identifying, and playing. Peer Help lets students assist with other projects. Finally, the instructor pages let educators and STEM researchers view statistics and track user progress.

The server stores information through individual domain objects. These domain objects refer to individual pieces of data important to CIRCLE functionality. These objects include classes corresponding to players, logging, projects, specimens, experiments, results, classification trees, groups of specimens and their hypotheses, game templates, synchronous and asynchronous communication, and course information. Each of these categories contains a number of helper classes to store information about user progress through CIRCLE.

The JSON Application Programming Interface (API) describes the messages sent between the client and the server. As a REST protocol, the client always begins by sending a message (called an Event) to the server, requesting information to be retrieved or stored. The server replies with a Response containing the requested information or flags noting whether the storage of information was successful or not.

Experiments

This implementation was refined and tested through two cycles of a beta test followed by an experiment. Beta tests involved using candy as the specimens to be identified and classified and were primarily designed to identify technical glitches in the software. After each beta test, an experiment was performed using a more realistic use case.

The first beta test asked participants to gather, suggest and perform observations, and identify a set of candy. Very few technical issues arose and players generally enjoyed the experience. This led to experiment one. In this experiment, CIRCLE was implemented in a General Ecology course. Students were asked to gather, elaborate, identify, and classify trees on a college campus. Usability scores during this experiment were quite poor, resulting in a variety of substantive changes which included a tutoring system, consistent navigation, a site map, and other changes.

In the second beta test, users used CIRCLE on a different set of candy. The results of the beta test showed CIRCLE to be technically sound, but it was unclear how to appropriately suggest experiments in order for them to be used. This led to the conclusion that suggesting experiments should be handled as a separate stage. Specimen gathering and elaboration was separated from the classification and identification tasks in the second experiment to great success. Students across campus classified and identified a subset of the trees collected by the students in experiment one while the observations and the results were provided by CIRCLE. Usability scores increased significantly from the first experiment.

Future work on CIRCLE will focus on the development of curricula and instructional materials for using CIRCLE in formal and informal settings. Potential classroom settings include geology, weed identification, biology, and taxonomy courses. Informal release to the Internet would allow users to identify and classify any interesting object. Further developments will also focus on the elaboration steps, making the interface more intuitive and usable for acceptable data entry into the CIRCLE ecosystem, building on experience developed by our group (Hokanson & Slator 2013). In addition, more game templates will be developed to attract a wider audience. Finally, researchers in STEM education could use the data collected by CIRCLE. Everything from chat messages to time on task can be analyzed to formulate more theories of how students learn, bringing the whole experience full CIRCLE.

PAPER: BUILDING A MOBILE COLLABORATIVE LEARNING ENVIRONMENT FOR THE IDENTIFICATION AND CLASSIFICATION OF REAL WORLD OBJECTS[1]

A paper submitted and accepted at the First International Conference on e-Learning e-Education and Online Training. Authored by Otto Borchert and Brian M. Slator

Abstract

This paper describes the theoretical framework and in-progress implementation of CIRCLE (Collaborative Identification, Retrieval, and Classification Learning Environment). CIRCLE uses recent research findings in collaboration, constructivism, mobile development, and retrieval learning to develop a multi-user tool for the identification and classification of real world objects. CIRCLE supports group efforts at taxonomy building by providing a framework for data gathering in the field and scientific hypothesizing and debate in a virtual laboratory. Future plans include a pilot usability study and classroom experiments to determine the effectiveness of the approach towards learning the identification of rocks and minerals (in a geology lab), weeds (in a weed identification course), and animals (in an ecology class).

Introduction

Identification and classification games are not new. Classification games abound on the Internet and in mobile app stores (App Shopper 2013; AmericanTorque.com 2013; Crimson Trails 2013; Kids Know It Network 2013; Kinder Web Games 2013). Rather than socially-mediated, actively constructing knowledge, as in Vygotsky's social constructivism (Vygotsky 1978), these identification games provide a method for retrieval learning. This theory suggests

[1] The material in this chapter was co-authored by Otto Borchert and Brian M. Slator. Otto Borchert had primary responsibility for the development of the CIRCLE algorithm, literature gathering, and authorship of this paper. Brian Slator served as a proofreader and beta tester for the software.

that the act of retrieving knowledge in multiple different ways results in better learning outcomes (Karpicke 2012).

Pervasive and location-based mobile gaming serves to harness the locative functionalities of mobile devices to supplement the real world with virtual information. Pervasive games “pervade” the user’s life, in contrast to games that occur at one particular time and place. Montola defines a game as an activity that involves a certain set of individuals that occurs at a particular place and at a particular time (Montola 2005). He further states that a pervasive game expands on this definition along social, spatial, or temporal dimensions. Social expansion means that people not playing the game can still participate within the game structure. Spatial expansion means that a pervasive game can be played anywhere, while temporal expansion means that a pervasive game can be playing at any time.

Even before computers, students were memorizing “things” using flashcards. A number of studies have been performed to determine effective ways to improve learning while using flashcards (Nist & Joseph 2008; Kornell 2009). In addition to physical artifacts for learning identification tasks, students handle actual physical specimens in a lab or real world setting. This project seeks to enhance even these highly effective methods of learning by putting the gathering and organization of content in users’ hands.

Location-based games are a subset of pervasive games, where a fixed location is used within the mechanics of the game itself. Location-based games use the environment to augment the players’ experience. For example, MicroBlog (Gaonkar, Li, Choudhury, Cox & Schmidt 2008) asks players to take pictures of their environment to share them with other users, using the environment as part of the game. In these games, a parallel world is created in virtual space. Individuals interact with this virtual world by going to real world locations. Other location based games that use user-generated content include Gopher (Casey, Kirman & Rowland 2007) and Indigator (Lee, Goh, Chua & Ang 2010).

Within this discipline, distinctions are made between designed activity and user-generated activity in mobile settings (Kukulska-Hulme, Traxler & Pettit 2007). Designed activities are developed by programmers, designers, and experts with a specific pedagogical plan in place. User-generated activity is content and structure spontaneously created by users to meet their own learning needs.

A number of designed activities have been created for a variety of different subjects and situations including games for geometry learning in an outdoor setting (Wijers, Jonker & Drijvers 2010), as museum guides (Grinter et al. 2002), in college orientation sessions (Schwabe & Goth 2005), in biology topics including genetics, protein synthesis, evolution, and food webs (Perry & Rosenheck 2012), and in weather forecasting (Klopfer, Sheldon, Perry & Chen 2012). While designed activities are important to provide guided opportunities for learning, the Internet allows individuals to guide and create their own learning affordances. This self-directed learning is intrinsic to this research project.

The combination of social constructivist principles that guide collaborative, user generated, multiplayer mobile games and retrieval learning that inform identification games provides an opportunity to create a new application for concept learning. We call this application CIRCLE, the Collaborative Identification, Retrieval, and Classification Learning Environment.

Implementation

CIRCLE is composed of five different activities: content acquisition, trait elaboration, hypothesis formation, tree construction, and game play. Students will be tracked to observe their use of the system and identify the amount of time they spend during each activity. Within the application, students can take on various collaborative roles, each correlated to one of the five activities.

Content Acquisition

First, students go out into the field using a mobile device to collect identifiable objects from the real world. The student will collect photographs, video, and audio where appropriate for the object under study. These multimedia artifacts are uploaded to a central server, where other users will be able to interact with them in real-time. Students performing this role will be identified as gatherers. An example task for a gatherer would be to take a photograph of the interesting trees or rocks in their neighborhood.

Trait Elaboration

Second, students look at acquired content from their group and offer suggestions on traits to observe or experiments to conduct to further refine a potential identification. The originating student then performs the requested experiments and observations to elaborate on the content either by performing them immediately, if feedback is prompt, or return to the object under study. As experiments provide attributes for features (traits) these will be stored for use by future students. For example, the first person to suggest a ‘hardness’ experiment using a ‘glass plate’ to classify a mineral will have those items stored in the system. Or a player might just observe from the photographic evidence the tree is covered with sharp needles, not leaves, and store that information in the system. Later players will be able to choose these observational or experimental results from the inventory. Students in this stage will be identified as elaborators. Continuing the rock example, an elaborator would suggest an acid test be performed on the neighborhood rock specimen.

Hypothesis Formation

Third, students offer hypotheses about the general classification or precise identity of the unidentified object. Eventually, an expert (typically the teacher) will verify the hypothesis. In absence of an expert, hypotheses could be voted upon, where more votes could be measured as a confidence in an identification. Students in this role would be called identifiers, and would offer the category "Conifer" in the tree example, or suggest "Limestone" in the rock example.

Taxonomy Construction

Fourth, students build versions of identification trees together. Traits and experiments will be shown graphically as potential branches, while the content collected by students will be the leaves. Students will move the nodes and leaves around in real time, seeing how other students are arranging the tree and offering suggestions for node placement. Players in this role will be called constructors. After enough identifications have been made in the hypothesis formation stage, constructors can build dichotomous keys of the trees or rocks.

Game Play

Finally, games are automatically created by the system based on gathered content. Students are given multimedia, traits, observations, and experiments and are asked to identify the object in game form. Students in this role will be called players. These students will play identification games similar to flashcards to help them retrieve knowledge they have gained from working in the other four stages.

Observing students in each of these stages may result in the identification of roles that are motivating or act to increase student learning. Ideally, students of all roles will:

a) Learn how to identify objects in the real world within their discipline of interest

b) Determine the experiments necessary to accurately identify these objects

c) Classify these objects for faster identifications

d) Use retrieval learning to strengthen their knowledge

e) Gain collaboration skills

Ultimately, CIRCLE is an innovative approach to identification and classification that combines successful attributes of other environments into one. CIRCLE will utilize effective learning science principles, harness the advantages and inherent motivation of user generated content, including images, videos, sound, and text, allow for synchronous and asynchronous collaborative interactions, create a system of juried peer review of results and hypotheses, and finally create computationally generated “flashcard” games to strengthen student learning.

Rationale for Implementation

According to constructivist theorists, students combine their prior understanding and new information to actively construct new knowledge (Richardson 2003). This learning is created within a social context. A person’s culture, environment, and social context combine to affect the construction of knowledge. This idea lies at the heart of social constructivism (Palincsar 1998). Individuals create new knowledge mediated by interactions with other human beings and with the environment. These interactions can be structured so that learning is indifferent, compromised, or supported by the efforts of other individuals. Social interdependence theory describes these interactions as being individualistic, competitive, and cooperative, respectively (Deutsch 1949; Deutsch 1962; Johnson 1970; Johnson & Johnson 1989).

Individualistic behavior is characterized by people working towards their own personal goals. The success or failures of others do not matter to the successful completion of their goals. While this approach towards learning can be effective, it has a number of negative side effects. People working individualistically tend to have lower psychological health, including lower self-esteem and higher anxiety (Johnson, Johnson & Smith 2007). They also do not gain the benefits of cooperative learning listed below.

Competition is characterized by the requirement that other people must fail in order for one to achieve personal goals. Competition increases self-acceptance based on meeting external standards and expectations (Norem-Hebeisen & Johnson 1981), reduces effort in lower achieving students (to reduce negative self-worth), decreases effort in high achieving students when they realize they'll always "win" (Wang & Yang 2003), marginalizes weaker, lower achieving members, and moves student focus from the process of the task to the end result (Shindler 2010). Because of these drawbacks, CIRCLE focuses on cooperative learning.

Cooperative learning provides many benefits to students including: an increase in academic skills (Sharan & Sharan 1978; Johnson & Johnson 1991; Kagan 1994; Slavin 1995), misconception identification and repair (Brown, Collins & Duguid 1989; Janis 1972; Koschmann, Kelson, Feltovich & Barrows 1996), collaborative work skills (Brown, Collins & Duguid 1989), the insight that the sum of individual knowledge is greater than the parts (Feltovich et al. 1996), the development of social skills (Goldman 1996), and the strengthening of inter-group relations (Slavin 1995).

These many benefits do not come simply by placing individuals into a group and giving them a task. Certain conditions must be met to create the most effective group dynamic. The ideal cooperative group fosters 5 primary conditions: positive interdependence, individual accountability, promotive interaction, social skills, and group processing (Johnson & Johnson 1989; Johnson, Johnson & Holubec 1998).

Positive interdependence is achieved when group members depend on each other for skills, talents, and knowledge to advance shared group goals. All group members believe that all other members are needed to succeed in the task at hand. Positive interdependence in CIRCLE will be enforced using game rules. For example, a certain number of people in the group will need to offer evidence on the identity of an object.

Individual accountability means that all members of a group must participate in the shared tasks and all actions should be visible to the group. No one should do all the work, nor should anyone do no work at all. This will be enforced in CIRCLE by means of a group status window, where the number of interactions each user has completed and when they completed it can be viewed by all members of the group.

Promotive interaction means that students actively communicate with one another in the group to achieve group goals. CIRCLE will include communication mediums so that students can discuss hypotheses, potential observations, and dichotomous key construction both synchronously and asynchronously.

Social skills are also required for positive group interactions. While not envisioned for the first iteration of CIRCLE, social skills could be prompted from students. For example, a student that is not participating could be encouraged to voice their opinion. Students could also be given a lesson on appropriate social skills or proper etiquette when discussing topics in an online setting. Soller (2001) instructs users to use sentence "openers" rather than allowing students to type in a free form text box. Her system uses these sentence openers to guide users to more effective conversation skills.

Finally, group processing occurs when groups reflect on their progress towards group goals. Message boards with appropriate prompts provide an opportunity for students to state how they feel the group is progressing towards the creation of their classification structures, or make suggestions on how to improve the group process.

Once students have successfully co-created knowledge in their cooperative groups, they need a way to reinforce this understanding. This reinforcement occurs when a student's knowledge is assessed. This assessment can come at the end of a lesson in the form of summative assessments, or during the lesson in the form of formative assessments. Ongoing, formative assessment is the key to higher learning gains (Wiliam 2007). Studies in retrieval learning have shown that students learn best when they actively attempt to "retrieve" the knowledge from memory during self-assessment, as opposed to re-reading study materials (Karpicke 2012).

Flashcards can be used effectively to retrieve knowledge from memory (Nist & Joseph 2008; Kornell 2009). Ideally, we hope to show that CIRCLE will provide a more interesting and potentially motivating method in the form of identification games. The final "game play" portion of the activity can be varied substantially. Students could be given an image or other multimedia content, a list of characteristics, or an example of where the object could be found. The number of images, and thus, the number of different available contexts, would only be limited by the number of images taken by the users. This combination of well-motivated learners, cooperative and collaborative co-construction of knowledge, and focused retrieval should provide an opportunity to help students learn identification and classification tasks in a more entertaining and effective way.

Discussion / Future Work

Implementation of CIRCLE is ongoing as a part of the primary author's Ph.D. dissertation. Upon completion of the described implementation, a series of experiments are planned to determine the usability and educational effectiveness of the proposed work.

A pilot study is planned to be held during May 2014. Approximately 20 STEM Education faculty and graduate students from North Dakota State University (NDSU) will be involved. Subjects will be divided into two groups: a software group and a manual group. Both groups will be given a bag of candy to identify and classify into a dichotomous key. The software group will use CIRCLE as described, making note of any difficulties they encounter. The manual group will use paper and pencil to perform the same task. The groups will be video recorded to determine similarities and differences in the identification and classification of objects in both a virtual and non-virtual setting. A secondary objective of the research is to determine any software errors and usability issues that exist in the initial prototype of CIRCLE. The System Usability Scale (Brooke 1996) will be administered to the software group upon completion of their tasks. This will be a useful first step, as these users come from fields as diverse as biology, chemistry, and mathematics, and will provide excellent feedback.

After the completion of the pilot study, further experiments are planned for actual classroom implementation. Three separate courses at NDSU have been identified for the use of CIRCLE. These courses include introductory geology lab, weed identification, and ecology. In these classrooms, subjects in different course sections will either use the original method of teaching identification or CIRCLE. This will provide feedback on whether the software is educationally effective or not. Since CIRCLE is an online tool, data will also be collected on what particular tasks students do, when they do them, how long it takes them, and how they interact with other students. This collection of data can also be analyzed to characterize patterns of student use and how it compares to expert use of the system.

CIRCLE is available online at circle.cs.ndsu.nodak.edu

LITERATURE REVIEW

The Classification, Identification, and Retrieval-based Collaborative Learning Environment (CIRCLE) is an application that uses mobile technology to gather user-generated content into an application for classifying and identifying real world objects. These content and classification structures are then used to create identification and classification games that students play to solidify their learning thorough practiced retrieval. Mobile learning (specifically, user-generated content in the context of mobile learning), structured collaboration, and serious gaming provides a unique combination of tools to create this experience for students to learn about identification and classification tasks. There is reason to believe that CIRCLE is a motivating, educational, and usable collaborative tool, because of the educational theory it is grounded upon. Creating a summative game that supplements the application may increase educational gains, usability, and other metrics. People who play a CIRCLE-based game will have a significantly different experience from those who simply use CIRCLE as a formative collaborative tool.

This research is grounded in the learning sciences. According to the International Society of the Learning Sciences, learning science investigations attempt to determine how people learn, together and alone, including those using information and communication technologies (ICT) and computer supported collaborative learning (CSCL) (Pea 2009). This cross disciplinary field draws its roots from cognitive science which includes fields as diverse as neuroscience, linguistics, anthropology, philosophy, and artificial intelligence (Miller 2003). One of the first links between artificial intelligence and the learning sciences included games, including Arthur Samuel's checkers playing game, which learned how to play checkers over time (Sutton & Barto

1998). In this research instead of teaching computers, computer games will be used to teach students.

Serious Games

Serious games are "any form of interactive computer-based game software for one or multiple players to be used on any platform and that has been developed with the intention to be more than entertainment" (Ritterfeld, Cody, & Vorderer 2009). The World Wide Web Instructional Committee is an ad-hoc group of faculty, staff, and students dedicated to the development of Immersive Virtual Environments for Education. WWWIC and its commercial successor, WoWiWe Instruction Co LLC, have developed several serious games in the form of these IVEs for science education. IVEs have been implemented for a range of scientific disciplines including anthropology, biology, economics, geology, environmental science, and computer science (Hokanson et al. 2008; White, McClean, & Slator 1999; Borchert et al. 2013; Slator & Farooque 1998; Saini-Eidukat, Schwert, & Slator 2001; Slator et al. 2011; Hill & Slator 2009; Slator et al. 2006). Each of these topics has spawned many research projects with positive results.

Although one may have a serious game with engaging play mechanics and tools for creating compelling content, this content must still be firmly integrated into the game to maximize the educational effectiveness of the environment. A number of scholars suggest that educational video games provide advantages for student learning (Papert 1998; Jenkins 2002; Gee 2003). Domain specific applications towards integrating educational content into serious games abound including applications to music (Denis & Jouvelot 2005) and geology (Grambo 1995). Fisch suggests three issues that must be overcome to integrate educational content into video games (2005).

First, educational topics and concepts must be adapted to the most appropriate medium. A mobile application for concept learning allows students to learn about identification and classification tasks in a variety of different contexts. For example, if a student sees a plant or animal in real life that they identified in the mobile game, it is relatively simple to load the game and verify their identification. This variety allows the student to build new connections in their memories to the content under study (Redish 2004).

Second, educational content must be the central focus of the game. The games developed using CIRCLE will always keep the educational content at the forefront. Content that is not educationally relevant or inaccurate will be labeled as such by experts and the community, keeping the most effective and useful information at the user's fingertips.

Finally, hints and feedback must be developed to scaffold student learning from easy to difficult topics. The decision structures developed for this research will include guards where tutoring and hints can be applied when students are having difficulty with a particular topic. Also, game mechanics can assist students in learning difficult topics by giving them smaller portions or going back to easier levels until the content is fully comprehended.

Identification Games

Identification and classification games are not new. Even before computers, students were memorizing "things" using flashcards. A number of studies have been performed to determine effective ways to improve learning while using flashcards (Nist & Joseph 2008; Kornell 2009). In addition to these alternative representations for learning identification tasks, students handle actual physical specimens in a lab or real world setting. This project seeks to enhance even these highly effective methods of learning by putting the gathering and organization of content in users' hands.

Classification games abound on the Internet and in mobile app stores, and include games to identify whales (New England Aquarium 2013), rock types (Kids Know It Network 2013), birds (Crimson Trails 2013), brain structures (Purpose Games 2013a), fish (App Shopper 2013), human anatomy (Jensen 2013), states (Purpose Games 2013b), letters (Kinder Web Games 2013), engines (AmericanTorque.com 2013), and many more. Rather than socially-mediated, actively constructing knowledge, as in Vygotsky's social constructivism (1978), these identification games provide a method for retrieval learning. This theory suggests that the act of retrieving knowledge in multiple different ways results in better learning outcomes (Karpicke 2012). The combination of social constructivist principles that guide collaborative, user generated, multiplayer mobile games and retrieval learning that inform identification games provides an opportunity to create a new application for concept learning.

Mobile Learning

Mobile learning is a young field. Widespread smartphone adoption has only occurred relatively recently. The iPhone, for example, was first released in 2007, while the Android operating system only debuted in 2008. Mobile technologies did exist before these now ubiquitous technologies, however. In fact, the first international conference on mobile learning was held in 2001 (Wingkvist & Ericsson 2011). These early efforts focused on the development of programs where students were loaned smartphones or used the Short Message Service (SMS) capabilities of phones to build collaborative networks (Kankaanranta, Neittaanmäki, Kuts, Islas-Sedano & Sutinen 2009). Most used local area network wireless technologies, not the cell phone wireless networks that cross the nation today. Mobile learning is not just a young field, but also a promising one. Kukulska-Hulme, Traxler, and Pettit (2007) note that "mobile learning should aim to innovate and to discover what is gained through having portable tools that support observations, interactions, conversations, and reflections”.

Pervasive and location-based gaming serves to harness the locative functionalities of mobile devices to supplement the real world with virtual information. Pervasive games “pervade” the user’s life, in contrast to games that occur at one particular time and place. Montola (2005) defines a game as an activity that involves a certain set of individuals that occurs at a particular place and at a particular time. He further states that a pervasive game expands on this definition along social, spatial, or temporal dimensions. Social expansion means that people not playing the game can still participate within the game structure. Spatial expansion means that a pervasive game can be played anywhere, while temporal expansion means that a pervasive game can be played at any time.

Location-based games are a subset of pervasive games, where a fixed location is used within the mechanics of the game itself. Location-based games use the environment to augment the players’ experience. For example, the MicroBlog game described below asks players to take pictures of their environment to share them with other users, using the environment as part of the game. These games create a parallel world in virtual space. Individuals interact with this virtual world by going to real world locations.

Within this discipline, Kukulska-Hulme, Traxler, and Pettit (2007) make a distinction between designed activity and user-generated activity in mobile settings. Designed activities are developed by programmers, designers, and experts with a specific pedagogical plan in place. User-generated content and structure is spontaneously created by users to meet their own learning needs.

A number of mobile designed activities have been created for a variety of different subjects and situations including games for geometry learning in an outdoor setting (Wijers, Jonker, & Drijvers 2010), as museum guides (Grinter et al. 2002), in college orientation sessions (Schwabe & Göth 2005), in biology topics including genetics, protein synthesis, evolution, and food webs (Perry & Rosenheck 2012), and in weather forecasting (Klopfer et al. 2011). While designed activities are important to provide guided opportunities for learning, the Internet allows individuals to guide and create their own learning affordances. This self-directed learning is intrinsic to this research project.

Determining the motivations of individuals who create user-generated content is paramount to this research as it is the first step of the learning framework. A number of research efforts to discover these motivations have been undertaken with a variety of Web 2.0 technologies. Leung (2009) found four primary needs that generating online content satisfied for creators. They include recognition needs (to establish a personal identity, gain respect and support, build confidence, gain satisfaction, and to show expertise), cognitive needs (increase knowledge, learn about and understand events, and clarify thinking), social needs (express feelings, views, experiences and thoughts, and to communicate with friends and family), and entertainment needs (to pass time, satisfy curiosity, be entertained, and follow current trends). Furthermore, the more these needs are met, the more content they produce. It is imperative that any user-generated content system provides a method for satisfying these needs. A survey by Daugherty, Eastin & Bright (2008) supports these assertions, noting that creators of user-generated content are typically motivated by being part of a community: minimizing self-doubt and maximizing social opportunities. In addition to these requirements, observing similar research projects involving pervasive games that depend on user generated content provides more guidance on the development of this framework. Some examples in this space include the Gopher Game (Casey, Kirman & Rowland 2007), Indagator (Lee et al. 2010), and MicroBlog (Gaonkar et al. 2008).

In the Gopher Game, individuals are able to create “gophers” that have a user-defined task to complete. These virtual gophers can be dropped off at a real world location. When a different user comes to that location, the gopher can be picked up and the task performed. An example task would be to bring the gopher to the beach and take a picture of a beach volleyball court. If a user thinks they have completed a task, the gopher is submitted to a jury, where players of the game decide whether the task was actually completed and points are distributed as determined by the rules of the game. The authors of Gopher Game note that game features should be kept simple for players with less experience and that rather than always giving point rewards, rewards could include access to new content or functionality.

Indagator is a more open ended content sharing system. Rather than being presented with specific tasks, users are presented with a map of markers indicating places where content has been shared. Content includes a title, text, multimedia content, tags, contributor name, location, content rating, and date. Users add these markers by submitting their own information to the centralized server. They can also rate other users’ content on a scale from one to five. Players also create “encounters” when they create new content. These encounters add another layer of gameplay to the system by letting other players find treasure, traps, or play a mini-game when they explore the newly created content. Players earn points in the game by creating content, rating content, and interacting with these encounters. A survey of potential Indagator users suggests that users will play these types of games, specifically for information discovery, entertainment, socialization, and relationship maintenance. Information discovery as a motivator bodes well for CIRCLE, a system that helps students learn about objects in their environment.

Micro-Blog is quite similar to Indagator, but is focused on content sharing without the gaming aspects. Mobile phones and their associated sensors including accelerometers, health sensors, wireless base stations in the area, global positioning via GPS, microphone, and integrated camera are used to gather data about a particular region. Web users direct these micro-bloggers to get more information about an area of interest. Initial studies show that individuals are responsive to the idea, but the user interface needs to be refined, privacy issues are paramount, and that students preferred text in informal situations, but voice in formal situations. These are all useful observations for implementation in this project.

CIRCLE is primarily a designed activity, but harnesses the advantages of user-generated content to spur motivation. The combination of a designed activity with aspects of user-generated content allows students to construct their own knowledge. To truly leverage the advantages of constructivism and the social motivations intrinsic to user-generated content, collaboration will need to be firmly integrated into the system as well.

Gamification

Deterding et al. (2011) defines gamification as “the use of game design elements in non-game contexts”. As compared to serious games, which use game design elements throughout their implementation, gamification takes a previous “non-game” and adds game elements like “points, badges, levels and leaderboards” (Deterding et al. 2011).

There is growing evidence for gamification techniques. The removal of gamification from an enterprise system decreased overall participation (Thom, Millen, & DiMicco 2012). Fitz-Walter, Tjondronegoro, & Wyeth (2011) found that students liked a gamified college orientation application, but noted that designers must balance usability and enjoyment, making sure that achievements are well matched to the goals of the application. Singer and Schneider (2012) attempted to increase the number of commits to a version control system using gamification and found that students had a mix of positive and negative comments in post-treatment interviews, but have not yet compared their results to those of previous "non-gamified" years. Landers and Callan (2011; cited by Singer and Schneider 2012) used gamification to motivate students to take nonmandatory quizzes.

Gamification of CIRCLE will help identify qualities of individuals who play the game compared to those who only use the tool. However, both the tool and the game will need to be solidly grounded in learning theory in order to maximize learning gains. The overarching plan was developed using backward design techniques (Wiggins & McTighe 1998).

Backwards Design

Backwards design offers a compelling way to create new learning modules. Rather than starting with activities, a book, or a specific range of content to cover, designers work "backwards" from by first identifying the learning goals for a course, module, or series of activities. After identifying these goals, assessments are developed to measure student progress towards the completion of these goals. Finally, activities are created that maximize the ability of students to complete the assessments.

This method of pedagogical design has many advantages. Buehl (2000) notes that the design keeps students focused on the point of a lesson, rather than getting mired in factual details, centers on the learning goals and assessment, allowing activities to focus on what students need to know rather than a wide array of potentially non-integrating facts, and focuses on global understandings created by the learning goals, rather than covering each and every topic in a book. Because of these advantages, backwards design was used to guide the development of CIRCLE. Identification and classification are two primary tasks of experts in many scientific disciplines. This makes these tasks a basis for excellent learning goals. Specific learning goals of CIRCLE are: "Students must be able to identify X, Y, and Z.", "Students should be able to indicate important features or characteristics of X, Y, and Z.", and "Students should be able to organize and classify their knowledge about these objects like an expert."

Assessments to determine if they have achieved these goals would include asking what tests or features of a given object they would use to perform a positive identification, asking them to create a dichotomous key involving a number of objects and their associated attributes, and asking the student to perform a series of observations and experiments and present an evidence-supported hypothesis of the identification of an unknown object. CIRCLE provides an ideal activity for these assessments, because, unlike take home assignments or exams, feedback is provided immediately and in context. Gamification adds a concrete implementation of these assessments and immediate feedback for students. Immediate feedback via ongoing assessments has been shown to increase student learning gains (Black & William 1998).

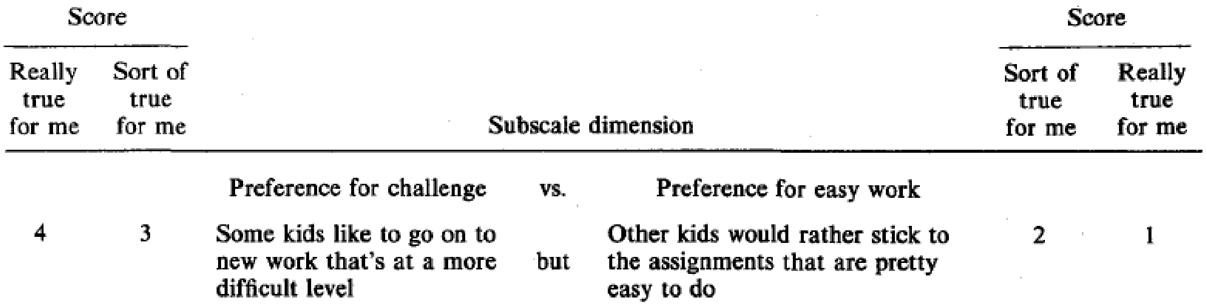
Self-regulated Learning

The field of education seeks to develop self-regulated learners, that is, learners who are able to create meaningful goals for themselves, are motivated to complete those goals, and know what they know (are metacognitive) to be able to determine when they have completed their goals (Zimmerman 1986, 1990). Individuals who perform self-regulation strategies are more likely to be in a higher achievement track, are more likely to persist in a post test, and have higher self-efficacy in academic tasks (Zimmerman 1989).

Games provide a unique opportunity to create self-regulated learners through their ability to motivate and inspire people to keep playing them (Malone 1980). Motivation has been theorized to span a continuum from amotivation (no motivation) to extrinsic motivation and ending at intrinsic motivation (Deci & Ryan 2000). Individuals with no motivation are not self-regulated learners, for they have no desire to learn at all. Individuals with intrinsic motivation towards a task are motivated by internal factors, such as finding the task enjoyable, interesting, or challenging (without being frustrating). This kind of motivation has been positively correlated to higher student grades and standardized test scores (Lepper, Corpus, & Iyengar 2005). Extrinsic motivation is supplied through the use of rewards or performance based incentives. People gain extrinsic motivation when an external factor encourages them to complete the activity. These can include getting a good grade, getting a promotion, receiving points or achievements, or avoiding punishment (Ryan & Deci 2000).

Operant conditioning theorizes that individuals avoid stimuli they find unpleasant and attempt to repeat stimuli they find enjoyable (Skinner 1938). However, operant conditioning is lacking in certain areas. All stimuli are not the same. If people are intrinsically motivated to perform a task (find it enjoyable), but are given performance based incentives, it has been shown that they lose interest, and don't want to do the task as often as people who are intrinsically motivated to perform the task. This is called the undermining effect and has been verified through brain scans (Murayama, Matsumoto, Izuma & Matsumoto 2010).

Motivation can be measured through self-report surveys or well-designed interactions. One self-report survey includes Harter's (1981) intrinsic motivation self-report survey which measures elementary school children's intrinsic motivation. A sample question is shown in Figure 1. This scale has been used in a number of motivational studies (Ginsburg & Bronstein, 1993; Guay, Boggiano, & Vallerand 2001; Newman 1990; Tzuriel 1989; Wong, Wiest, & Cusick 2002). There are two main disadvantages to this scale. First, the scale places intrinsic and extrinsic motivation on separate ends of a spectrum. Some students may actually be motivated by intrinsic and extrinsic means simultaneously. Second, this scale measures motivation specifically within a school setting, not on a particular task.



Score					Score	
Really true for me	Sort of true for me	Subscale dimension			Sort of true for me	Really true for me
		Preference for challenge	vs.	Preference for easy work		
4	3	Some kids like to go on to new work that's at a more difficult level	but	Other kids would rather stick to the assignments that are pretty easy to do	2	1

Figure 1. An example question on Harter's self-report intrinsic motivation scale (1981).

CIRCLE provides an alternative method of measure motivation. If a threshold for successful achievement of an external reward is met (for example, getting an A on the lesson), students can be monitored to determine if they continue to play the game afterwards (called a free choice period). Students that continue to play the game can be said to be intrinsically motivated to play, since the external reward no longer applies to them.

Another instrument for measuring intrinsic motivation is the Intrinsic Motivation Inventory, used in a large number of motivational experiments (Ryan 1982; Ryan, Mims & Koestner 1983; Plant & Ryan 1985; Ryan, Connell, & Plant 1990; Ryan, Koestner & Deci 1991; Deci, Eghrari, Patrick & Leone 1994). The instrument was validated by McAuley, Duncan & Tammen (1989).

Cooperative Learning

According to constructivist theorists, students combine their prior understanding and new information to actively construct new knowledge (Richardson 2003). This learning is created within a social context. A person's culture, environment, and social context combine to affect the construction of knowledge. This idea lies at the heart of social constructivism (Palincsar 1998). Individuals create new knowledge mediated by interactions with other human beings and with the environment. These interactions can be structured so that learning is indifferent, compromised, or supported by the efforts of other individuals. Social interdependence theory describes these interactions as being individualistic, competitive, and cooperative, respectively (Deutsch 1949, 1962; Johnson 1970; Johnson & Johnson 1989).

Individualistic behavior is characterized by people working towards their own personal goals. The success or failures of others do not matter to the successful completion of their goals. While this approach towards learning can be effective, it has a number of negative side effects. People working individualistically tend to have lower psychological health, including lower self-esteem and higher anxiety (Johnson, Johnson & Smith 2007). They also do not gain the benefits of cooperative learning listed below.

Competition is characterized by the requirement that other people must fail in order to achieve personal goals. Competition increases self-acceptance based on meeting external standards and expectations (Norem-Hebeisen & Johnson 1981), reduces effort in lower achieving students (to reduce negative self-worth), decreases effort in high achieving students when they realize they'll always "win" (Wang & Yang 2003), marginalizes weaker, lower achieving members, and moves student focus from the process of the task to the end result (Shindler 2010). Because of these drawbacks, CIRCLE focuses on cooperative learning.

Cooperative learning provides many benefits to students including: an increase in academic skills (Sharan & Sharan 1978; Johnson & Johnson 1991; Kagan 1994; Slavin 1995), misconception identification and repair (Brown, Collins & Duguid 1989; Janis 1972; Koschmann et al. 1996), collaborative work skills (Brown Collins & Duguid 1989), the insight that the sum of individual knowledge is greater than the parts (Feltovich et al. 1996), the development of social skills (Goldman 1996), and the strengthening of inter-group relations (Slavin 1995).

These many benefits do not come by simply placing individuals into a group and giving them a task. Certain conditions must be met to create the most effective group dynamic. The ideal cooperative group fosters five primary conditions: positive interdependence, individual accountability, promotive interaction, social skills, and group processing (Johnson & Johnson 1989; Johnson, Johnson, & Holubec 1998).

Positive interdependence is achieved when group members depend on each other for skills, talents, and knowledge to advance shared group goals. All group members believe that all other members are needed to succeed in the task at hand. Positive interdependence in CIRCLE is enforced using game rules. For example, a certain number of people in the group need to offer opinions on the identity of an object.

Individual accountability means that all members of a group must participate in the shared tasks and all actions should be visible to the group. No one should do all the work, nor should anyone do no work at all. CIRCLE enforces individual accountability by means of a group status window, where the number of interactions each user has completed and when they completed it can be viewed by all members of the group.

Promotive interaction means that students actively communicate with one another in the group to achieve group goals. CIRCLE includes communication mediums so that students can discuss hypotheses, potential observations, and classification structure construction both synchronously and asynchronously.

Social skills are also required for positive group interactions. While not included in this iteration of CIRCLE, social skills could be prompted from students. For example, a student that is not participating could be encouraged to voice their opinion. Students could also be given a lesson on appropriate social skills or proper etiquette when discussing topics in an online setting. Soller (2001) instructs users to use sentence "openers" rather than allowing students to type in a free form text box. Her system uses these sentence openers to guide users to more effective conversation skills.

Finally, group processing occurs when groups reflect on their progress towards group goals. Message boards provide an opportunity for students to state how they feel the group is progressing towards the creation of their classification structures, or make suggestions on how to improve the group process.

Cooperative and collaborative learning tend to be used interchangeably; however, there is a distinct difference between the two. Panitz (1999) describes cooperative learning as learning that occurs when acquiring foundational knowledge. Foundational knowledge is factual information like the content of a document, or the identity of an object. In cooperative learning, the teacher is central to the discussion as the distributor of foundational knowledge.

Collaborative learning is used when acquiring non-foundational knowledge, that knowledge that is derived from the facts of the discipline. In this situation, students are the center of the learning activity, building knowledge from their underlying foundational knowledge, while the instructor is there to guide them. CIRCLE will combine both cooperative and collaborative approaches. During the initial phases where students work together to determine the identity of an object, they use cooperative learning. When they start to construct classification structures, there is no "correct" answer, and thus students collaborate to construct a shared representation of classification structure.

Retrieval Learning

Once students have successfully co-created knowledge in their cooperative groups, they need a way to reinforce this understanding. This reinforcement occurs when a student's knowledge is assessed. This assessment can come at the end of a lesson in the form of summative assessments, or during the lesson in the form of formative assessments. As stated earlier, ongoing, formative assessment is the key to higher learning gains. Studies in retrieval learning have shown that students learn best when they actively attempt to "retrieve" the knowledge from memory during self-assessment, as opposed to re-reading study materials (Karpicke 2012).

Flashcards can be used effectively to retrieve knowledge from memory (Nist & Joseph 2008; Kornell 2009). CIRCLE builds on this effectiveness through summative gamification. The final "game play" portion of the activity can be varied substantially. Students could be given an image or other multimedia content, a list of characteristics, or an example of where the object could be found. The number of images, and thus, the number of different available contexts, would only be limited by the number of images taken by the users. This combination of well-motivated learners, cooperative and collaborative co-construction of knowledge, and focused retrieval provides an opportunity to help students learn identification and classification tasks in a more entertaining and effective way.

Sociability and Usability

Vatrapu, Suthers & Medina (2008) define three scales for measuring the effectiveness of a computer supported collaborative learning (CSCL) environment: learnability, sociability, and usability. Learnability relates to the ability of users to learn the content matter related to the environment. Kreijns, Kirschner, Jochems & Van Buuren (2007) states that sociability “refers to how CSCL environments can differ in their ability to facilitate the emergence of a social space; the human network of social relationships between group members which is embedded in group structures of norms and values, rules and roles, beliefs and ideals." Usability describes how well users can achieve specific goals effectively, efficiently, and satisfactorily using the tool. Learnability is a main driving force of this research, but sociability and usability should not be ignored. An unusable tool will result in people not using it. Well-functioning social interaction is necessary for an effective, functional group. Measuring both usability and sociability will be important.

One potential usability measure is the System Usability Scale (SUS; Brooke 1996). It has been used to determine the usability of a wide range of computer applications (Bangor, Kortum, & Miller 2008) and provides an excellent test of CIRCLE’s usability. A number of sociability measures have been developed that could be used in this research as well. The Social Space Scale measures the positive and negative behaviors that a group performs. Higher score on the "positive group behavior" factor would indicate a healthy social space, while a high score on the "negative group behavior" factor would indicate an unhealthy social space (Kreijns, Kirschner, Jochems & Van Buuren 2004). The Sociability Scale measures the ability of an environment to create a social space by asking users if the environment enables them to build social groups (Kreijns et al. 2007). The Social Presence scale measures the level to which a person feels that someone else in the environment is there or "present" (Kreijns, Kirschner, Jochems & Van Buuren 2011). This research focuses on using the System Usability Scale and the Social Space Scale.

Knowledge Representation

Knowledge representation is a well-established field, especially the relevant domains of semantic networks (Brachman 1983) and graph-based knowledge representation (Chien & Mugnier 2009). But decision trees (Quinlan 1986; Quinlan 1987) and fuzzy sets (Yuan & Shaw 1995) have proven to be more relevant in data mining and applied knowledge systems.

The most seminal work in curriculum representation is the deep history of expert systems (Hayes-Roth, Waterrman & Lenat 1984). A relevant relative is ASK systems (Ferguson, Bareiss, Birnbaum & Osgood 1992), but this focuses more on story-based teaching rather than educational gaming. Knowledge representation has been used (Gang, Wang & Ng 1996) to address the 2-sigma problem (Bloom 1984; Corbett 2001) of scaling tutoring.

One method of describing a set of objects is to list all of the properties of each individual object in a table. Objects are listed in the rows, while attributes of those objects are listed in columns. An example is given in Table 1. Classification tables provide a complete list of all possible combinations of tests, traits, and attributes of a set of objects. The brute force method of identifying an object would be to determine the result of every test given in every column, then finding the row that matched every test completed. This is a "novice" approach towards identification. In the interest of moving students from novices to experts, we propose a system of "expert" knowledge.

Table 1. An example classification table for classifying a set of minerals.

Name	Appearance	Magnetic?	Cleavage
Gypsum	White Vitreous	No	Single inelastic
Muscovite Mica	White Vitreous	No	Single elastic sheets
Graphite	Black Metallic	No	Perfect
Magnetite	Black Metallic	Yes	None

A decision tree is a data structure used to classify an object given the object's attributes. Branches indicate tests that need to be performed in order to traverse the tree, while leaves indicate the final classification. The creation of an optimized decision tree (that is, a tree with the minimum number of branches that still covers all attribute data in the set) will be used in the creation of an "expert" system from classification tables (Figure 2).

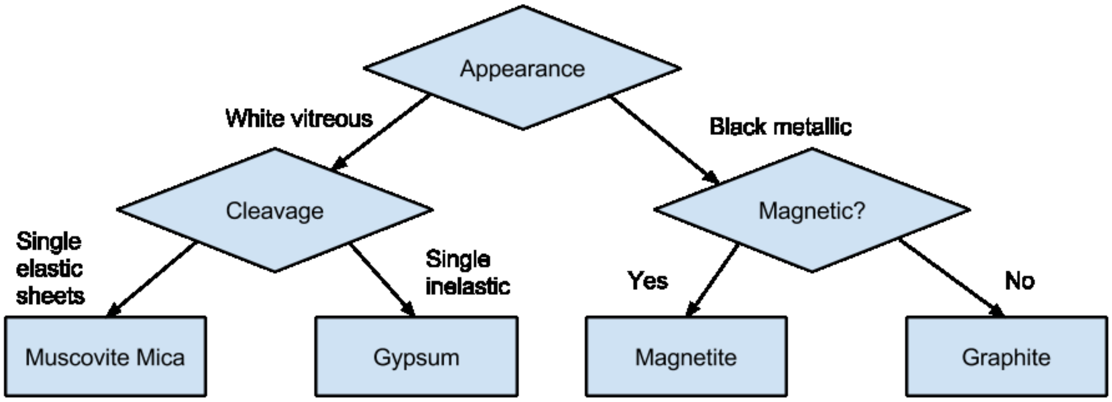


Figure 2. An example decision tree for classifying a set of minerals based on the data in Table 1.

Some decision tree induction algorithms include ID3, and its successor, C4.5 (Quinlan 1993), fuzzy decision trees (Yuan & Shaw 1995), classification and regression trees (CART) like random forests (Breiman 2001), chi-squared automatic interaction detector (CHAID; Kass 1980), and multivariate adaptive regression splines (MARS; Friedman 1991).

IMPLEMENTATION

CIRCLE contains three main components: a client, a server, and a communication protocol. The client describes the player’s view of the game, implemented as a series of javascript-enabled web pages. The server stores game state information for distribution across all clients. The Javascript Object Notation Application Programming Interface (JSON API) communicates a standardized set of messages between clients and servers.

JSON API - Events and Responses

CIRCLE implements a Representational State Transfer (REST) protocol (Fielding 2000) for message passing to facilitate the development of multiple clients and multiple servers. This protocol standardizes communication between a potentially diverse set of clients and servers. It offers the flexibility to create new clients and servers that are interoperable with prior implementations. This document describes the Javascript-based client and JavaMOO-based server as of summer 2015.

The protocol defines two message types: Events and Responses. Events are API requests that are sent by the client to the server. Responses are replies from the server regarding an Event. Figure 3 shows the high level communication that occurs when the client sends a *SaveDiagram* event and the server responds with a *SaveDiagramResponse*. A view of client side event construction is shown in Figure 4. Server side message processing is shown in Figure 5.

CIRCLE formats events and responses using JavaScript Object Notation (JSON). This compact format for describing messages is well suited towards network communication and is widely used throughout web communication (Yap, Huang, Dodson, Lam & McKeown 2010; Zhu, Wang, Chen, Liu & Qin 2010).

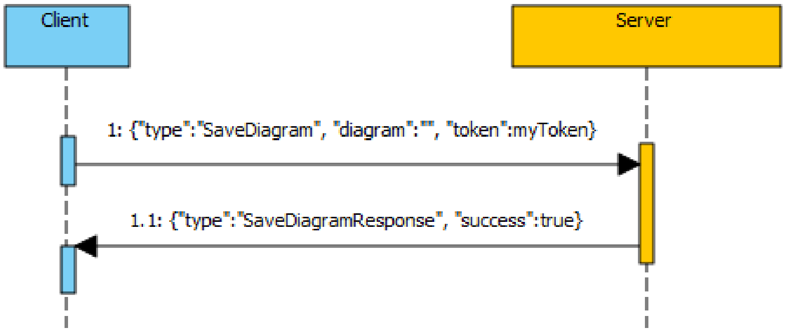

Client
Server
1: {"type":"SaveDiagram", "diagram":"", "token":myToken}
1.1: {"type":"SaveDiagramResponse", "success":true}

Figure 3. High level sequence diagram of JSON protocol.

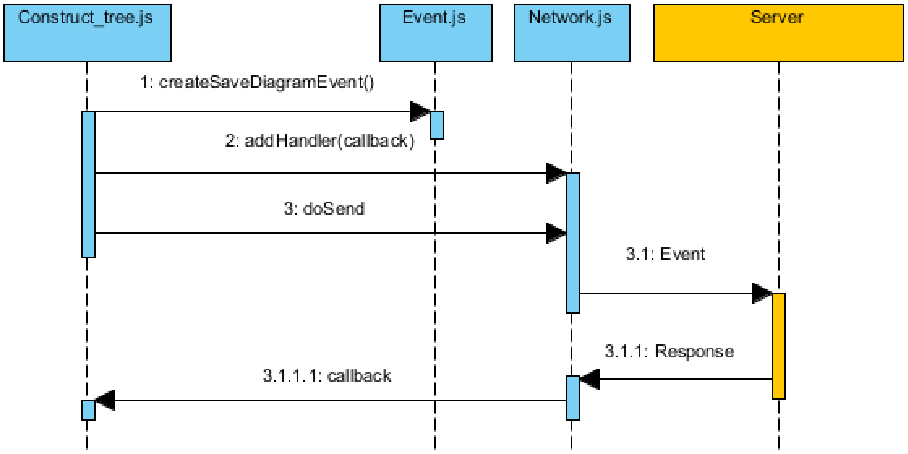


Figure 4. Sequence diagram for client interaction.

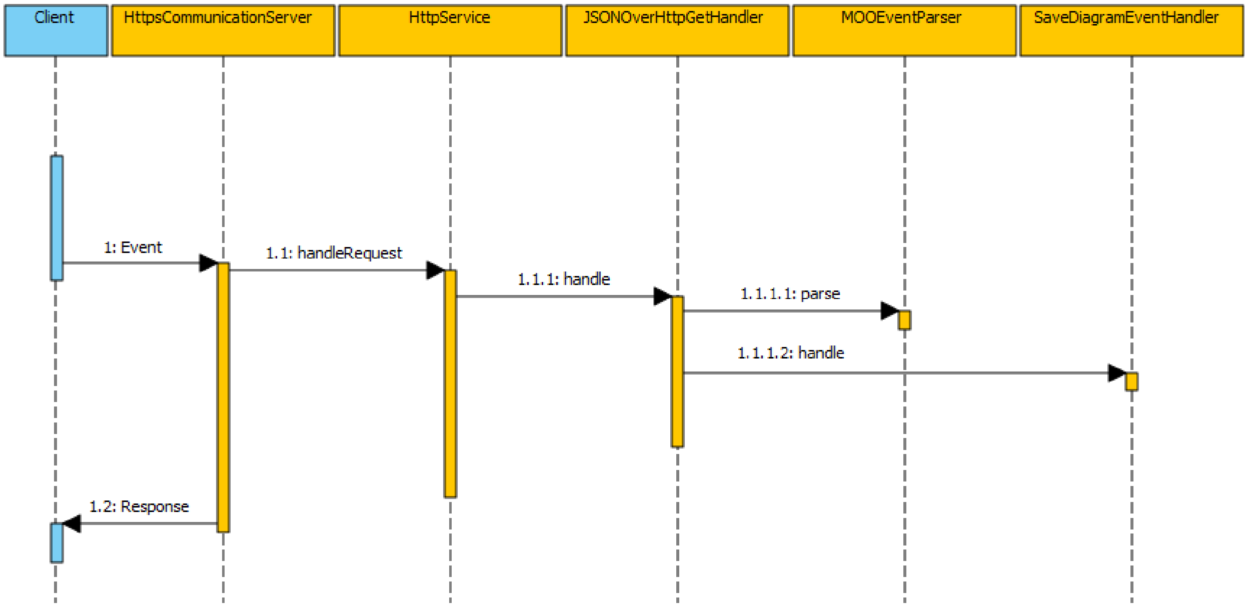


Figure 5. Sequence diagram for server interaction.

JSON implements messages as sets of key/value pairs. Keys are always listed as strings, while values can be Boolean values, numbers, arrays, strings and other key/value pairs. This document denotes events and responses in italics below. The CIRCLE JSON API documentation, including the syntax for all events and responses, is available at: https://circle.cs.ndsu.nodak.edu/wiki/index.php/JSON_API

Client Implementation

The CIRCLE client is written in a combination of HTML5 and Javascript. These languages have exploded in recent years as the field converges on a set of technologies and specifications that are installed on both mobile and desktop platforms. This allows CIRCLE to be deployed to a diverse number of computing devices and maximize its use.

The client used in the first beta test and experiment used a pure HTML5+Javascript client. The only library used was JQuery which provides "HTML document traversal and manipulation, event handling, animation, and Ajax" event calls (www.jquery.com). One of the drawbacks of this approach was that it was difficult to create a modern looking interface with sliders, buttons, and dialogs.

For the second beta test and experiment, JQueryUI, JQuery's Fileupload plugin, and Autogrow were added to the implementation. JQueryUI adds support for dialogs, more graphically pleasing user interface elements, and modern templates for potential graphical adaptation in the future. The JQuery Fileupload plugin provides a method to upload files, specifically the images taken during specimen gathering and elaboration. Autogrow allows fixed sized text areas to grow dynamically based on entered content. This allows users to enter longer descriptions and bulletin board messages than they could in experiment one.

A top level state machine diagram representing client state transitions is shown in Figure 6. This diagram describes the different states that the client will go through as players view different screens. A description of each of the states including the JSON API messages used to communicate between the client and the server follows.

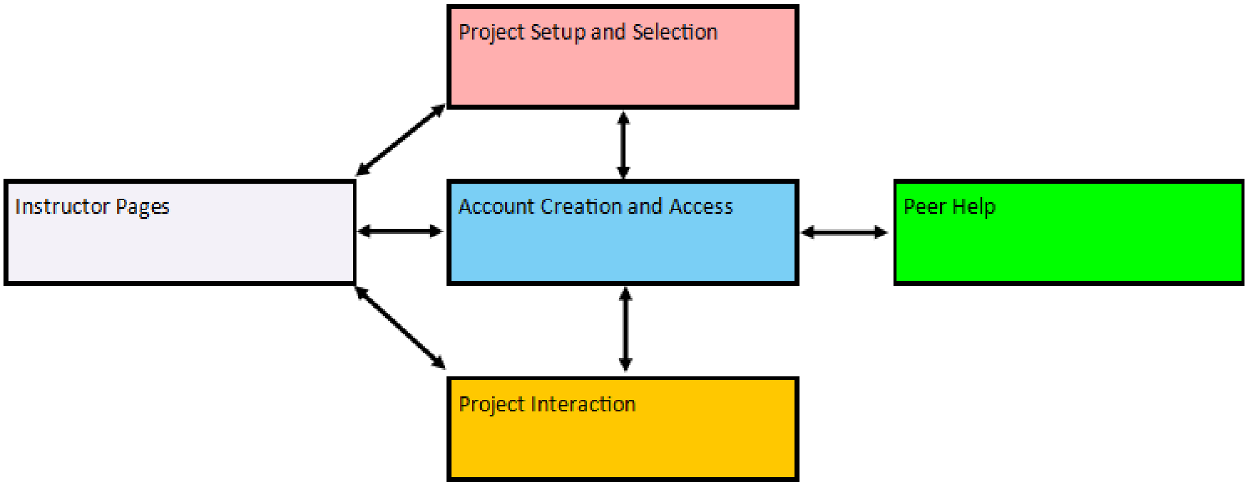


Figure 6. High level state machine diagram for the CIRCLE client. Each component is more fully described below.

Account Creation and Access

The first two tasks in CIRCLE are character creation and account access. These actions are accomplished through three distinct screens (Figure 7). Users can login, create accounts and receive account credentials on these pages.

The three account types are students, faculty, and admins. Admin accounts can access account information and statistics for all users. Faculty create and modify courses for students. Faculty users don't actually play CIRCLE, they organize courses for the actual student players through a series of instructor pages. Student accounts are the "players", that is, they complete the five CIRCLE tasks: gathering, elaborating, classifying, identifying, and playing. CIRCLE automatically creates the admin account when it is installed. The Create Account interface (described below) creates faculty and student accounts.

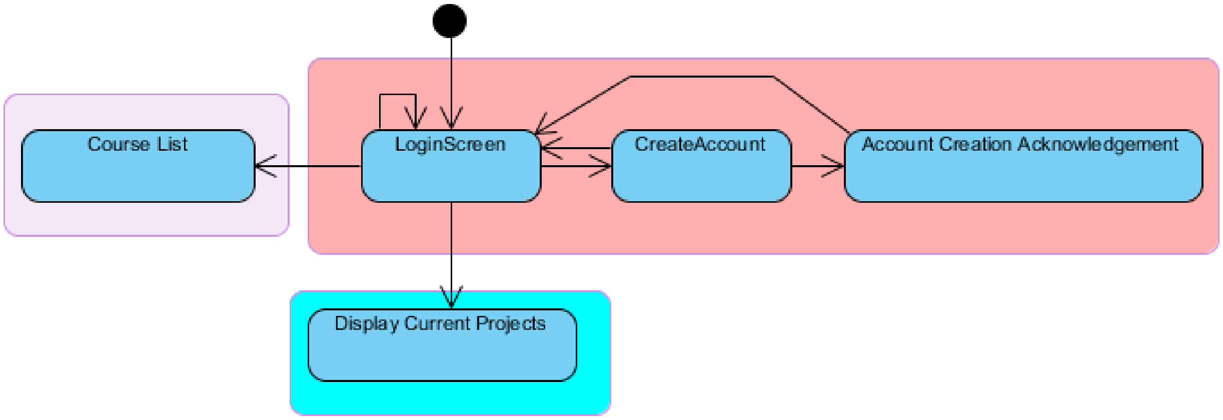
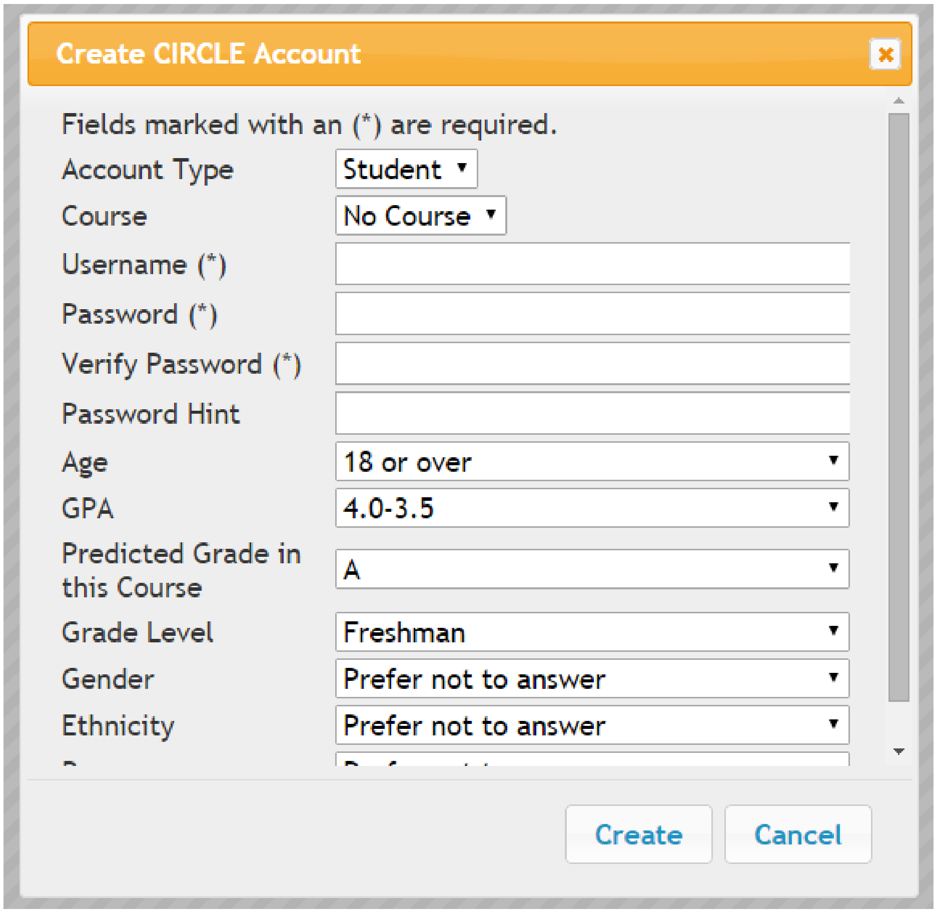


Figure 7. State diagram for account creation and access (pink) with links to the faculty (lavender) and student (blue) starting areas. The black dot indicates the start state.

Create Account

The create account screen begins by sending a *GetAvailableCourses* event to the server to determine what courses are open for registration. A course can be opened or closed for registration on the instructor pages via the course options screen. This list of courses populates the Course drop down box (Figure 8). Individuals are asked to submit user demographics and account credentials. Demographic data includes: age, GPA, grade level, gender, race, and ethnicity. Additional identifying information that is currently hidden, but can be used in future projects includes socioeconomic status, address, city, state, zip, phone, and e-mail.

Players can either click the Create Account or Cancel button after filling out the form. The Create Account button verifies the validity of the entered e-mail address and checks to see if the passwords match. It then sends a *Create* event to the server to create the account. If the username is already taken, the “result” property is set to false, and the user is warned that they need to select a new username. If the “result” property is true, a user is created on the server and the create account acknowledgment page is shown. If the user clicks cancel, they are returned to the login screen.



Create CIRCLE Account

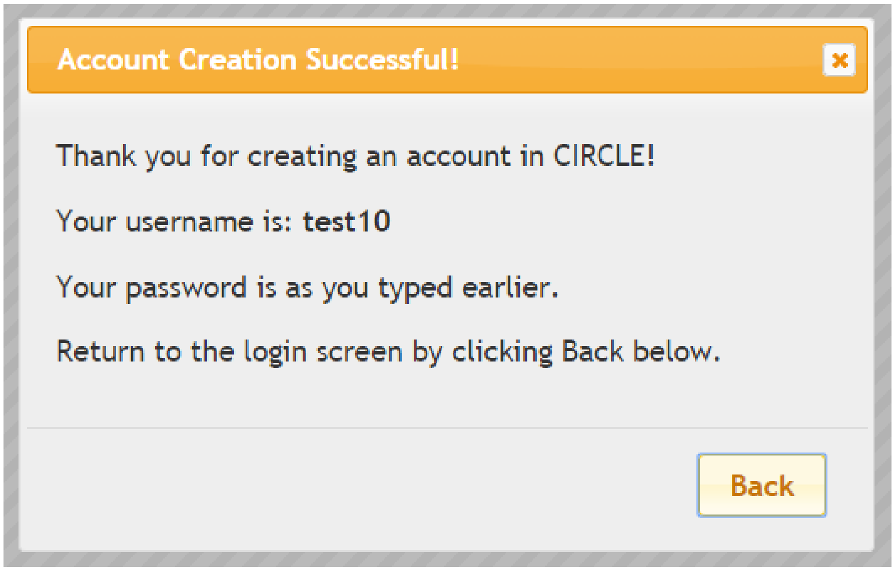
Fields marked with an (*) are required.

Account Type	Student
Course	No Course
Username (*)	
Password (*)	
Verify Password (*)	
Password Hint	
Age	18 or over
GPA	4.0-3.5
Predicted Grade in this Course	A
Grade Level	Freshman
Gender	Prefer not to answer
Ethnicity	Prefer not to answer

Figure 8. The create account interface for students. Account type includes students and faculty. Course is populated by the *GetAvailableCourses* event.

Account Creation Acknowledgement

The account creation acknowledgement screen (Figure 9) indicates that account creation was successful, shows the player's username, and states that the password is the same as was entered on the Create Account screen. The user must click the Back button to return to the login screen to connect.



Account Creation Successful!

Thank you for creating an account in CIRCLE!

Your username is: **test10**

Your password is as you typed earlier.

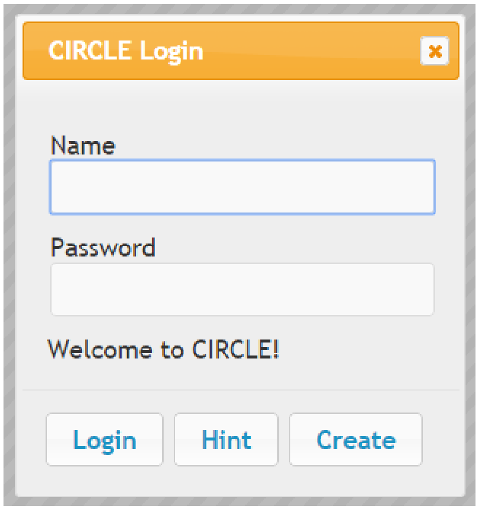
Return to the login screen by clicking Back below.

Figure 9. The create account acknowledgement dialog.

Login

The login interface is the first screen a user sees when starting the program (Figure 10). The interface includes text fields for entering a username and password and three buttons: Login, Hint, and Create. If the player enters a username, they can click the Hint button which sends a *GetPasswordHint* event to the server and returns the hint that was stored during account creation. If Login is clicked, a *Login* event is sent to the server with the entered username and password. If the resulting response returns with result property set to true, the player is logged in, otherwise the user is given an error message that their password was incorrect. The Create button moves them to the account creation screen.

The server sends an authentication token and an integer representing the set of restrictions that client will be using after successful login. The token is used throughout the game to identify a logged in user. Restrictions are a list of actions this particular client is not allowed to perform by virtue of their instructor setting those restrictions (See “restrictions” in the course options view below).



Welcome to CIRCLE!

Figure 10. The login screen in the JQueryUI interface.

Project Setup and Selection

After logging in, the client's primary layout is divided into a series of modular components. At the top of the page is a Navigation pane, which allows the user to switch web pages, moving between activities in the interface. The main view area is displayed immediately below the navigation pane as a series of accordion views from JQueryUI. These accordion views allow the user to expand and contract sections of the interface. In addition, tutor messages appear as JQueryUI modal dialogs, which users must close in order to view the underlying content. Each of these interfaces will be discussed in turn. A diagram of the transitions between each view in the project setup and selection section is shown in Figure 11.

Navigation Pane

All screens after a player has connected include a navigation pane (Figure 12). This pane is displayed at the top of the page and includes a cookie-crumb like navigation menu (Levene 2010), where players can visit any page in the hierarchy that is immediately above the current page. The navigation pane also includes a button to allow the user to disconnect from the game, which, when clicked, sends a *Disconnect* event to the server and returns the player to the login screen. The navigation pane also includes a system wide site map (Peruri, in preparation). On the far right, the navigation pane includes the currently logged in user's name.

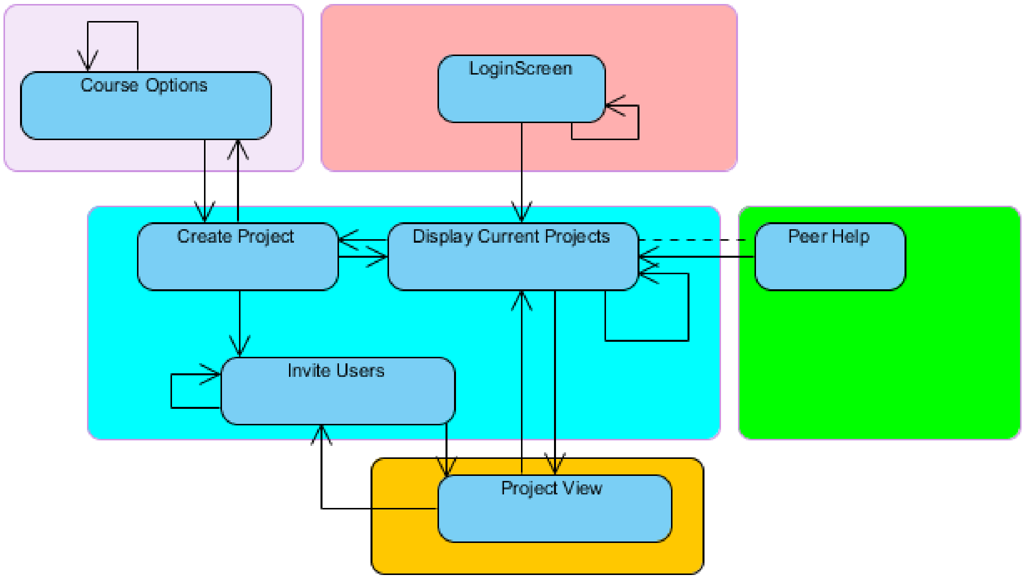


Figure 11. State diagram showing project setup and selection screens (blue) with links to the faculty pages (lavender), creation and access (pink), project (orange), and peer help (green). Dashed lines indicate transitions that require a prerequisite to be fulfilled.

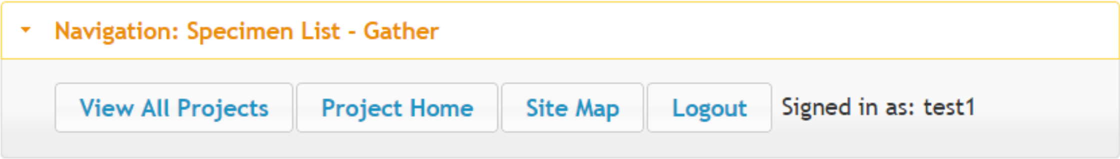


Figure 12. The navigation pane at the top of the gather specimen list page, the "View All Projects" button leads to the display current projects view and the project home button leads to the project view.

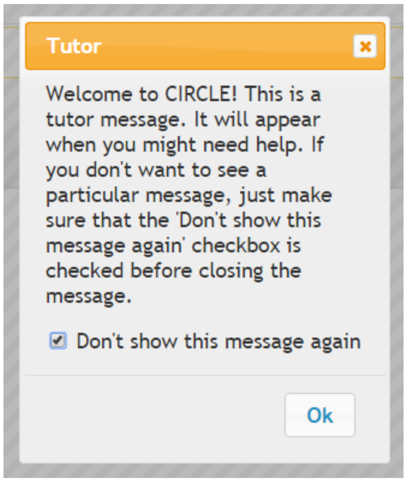
The Navigation Pane contains a number of callback methods that are executed when a player clicks on buttons in the pane. One special button is the "View All Projects" button. When this button is pressed, there are two options. If the player's 'course' cookie is set, that means this player is a faculty user, and they should be returned to the Course View page (see below). Otherwise, a student clicked the button, and they are sent to the Display Current Projects view.

The button also sends a *BeginTask* message with a task of "Leaving", updating the server when the player is no longer working on the project.

Tutor Messages

Tutoring is accomplished through both client and server methods. The client stores a series of messages in its tutoring.js file and displays them when relevant. The tutor message is shown in a dialog box with a checkbox that allows the player to ignore the tutor message in the future (Figure 13). Information included for all tutor messages include a title for the dialog, tutor text, whether the "Don't show this message again" check box should be shown, and a callback that can be executed when the dialog is closed.

All client views with associated tutoring begin by sending a *GetIgnoredTutorMessages* event. This indicates which tutor messages the players has specifically decided to ignore. The client also sets a timer to query for server-initiated tutor messages using the *GetTutorMessages* event. The server sends a title for the dialog box, the text of the tutor message, and whether the tutor message can be ignored to the client, which displays the dialog box using the same methods as the client-initiated messages are delivered.



Welcome to CIRCLE! This is a tutor message. It will appear when you might need help. If you don't want to see a particular message, just make sure that the 'Don't show this message again' checkbox is checked before closing the message.

☑ Don't show this message again

Figure 13. An example tutor dialog. If the player unchecks the "Don't show this message again" box, the message will appear the next time the player activates the tutor.

Necessary tutor messages were identified through task modeling. In Table 2, user goals are identified in the left column, while the tasks and subtasks in order to complete those goals are in the right column. Tutor messages are implemented for each of the tasks and subtasks, moving users through the CIRCLE experience. These tutor messages are displayed in both dialog boxes and as red text in context appropriate locations.

Table 2. CIRCLE task model.

Goal	**Task / Subtasks**
Project Management	• Create a new account • Create a project • Invite new members to project • Accept invitations to project
Classify and Identify Objects	• Gather Specimens • Enter description and location • Uploading multimedia • Suggest Experiments • Enter instructions, description, and type of data needed to perform the suggested experiment • Perform Experiments • Complete suggested experiments • Upload multimedia, if required • Create classification tree • Add start node • Add experiment nodes • Animating specimen nodes • Distinguish between specimens in a specimen container node • Identify specimens • Vote for appropriate specimen, majority vote required to identify the specimen • Play the identification game • Select appropriate experiments • Select correct identification
Communication	• Write bulletin board messages • Read unread bulletin board messages • Write synchronous chat messages

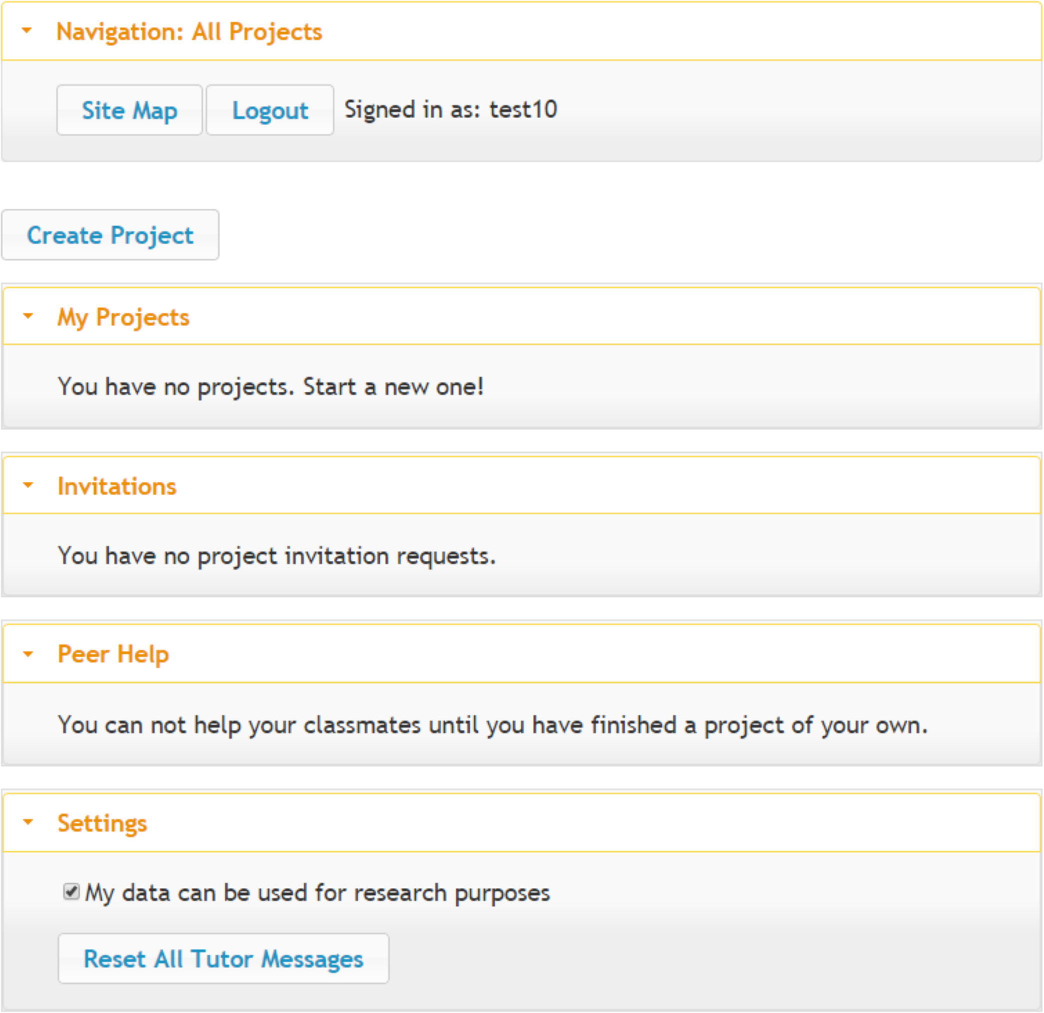
Display Current Projects

The Display Current Projects screen is the main entry point for all student players (Figure 14). The client begins by sending four event sequences to the server. The *GetPlayerOptions* event determines if the player wishes to be a part of a research study. It is used to populate the "My data can be used for research purposes" checkbox. The *GetProjects* event returns a list of the player's current projects. The event sequence *GetPeerHelpHighScore*, *GetSpecimens*, and *GetSpecimen* retrieves the high scores for peer scoring and determines if the player has completed a project, and thus, can use the peer help interface. A project is deemed complete if all of the specimens have been identified, that is, has a name that is not "Unknown". The fourth event sent to the server is *GetInvitations*, which returns the player's list of invitations to other projects.

If all restrictions are disabled, the screen displays a list of current projects, a button to create a new project, a list of invitations so this player can join other projects, an area for the peer help interface, a checkbox for indicating whether the player wants to be included in research study results, and an option to clear all tutoring messages.

If the Create Project or Peer Help buttons are clicked, the user is simply moved to the appropriate screen. If a project button is clicked, the client sets a cookie with the name "project" on the player's web browser. This cookie contains the ID of the project that was selected. This cookie will be used by all subsequent pages to interact with that particular project on the server. The player is then sent to the project view screen.

The *GetInvitations* event is sent to the server on a timer, to dynamically show new invitations as they are created. If a player clicks on the Accept or Decline buttons of one of these invitations, an *InvitationReply* message is sent to the server indicating the user's choice. If they accept, a *GetProject* message is sent to refresh the current list of projects.



Signed in as: test10

You have no projects. Start a new one!

You have no project invitation requests.

You can not help your classmates until you have finished a project of your own.

☑ My data can be used for research purposes

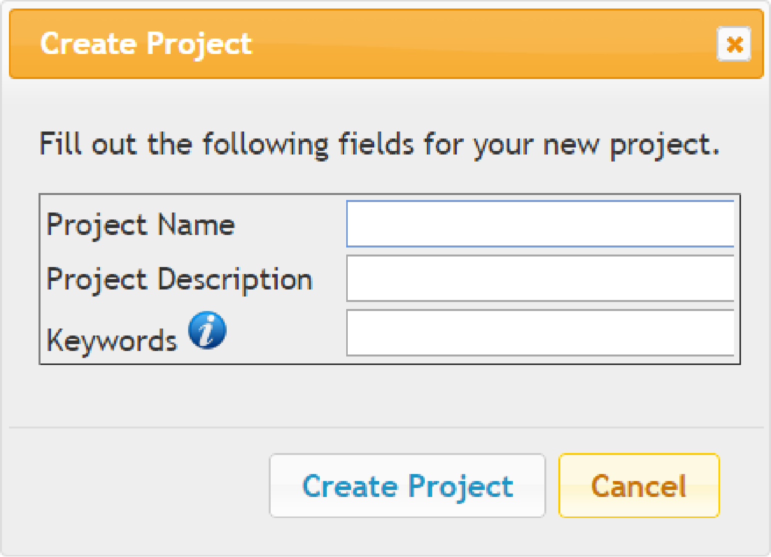
Figure 14. The first screen shown to students after logging in. The navigation pane is included for reference, it will not be included in future screenshots.

The final accordion view shows settings associated with the user's account. If the research study checkbox is clicked, a *SetPlayerOptions* event is sent to the server to store the user's choice. If the Reset All Tutor Messages button is clicked, the *ResetTutorMessage* event is sent to the server. This event uses the keys retrieved via the *GetIgnoredTutorMessages* event to make it so that all tutor messages will be shown to the user again.

Create Project

The user creates a new project using the Create Project screen (Figure 15). Projects contain a set of objects that are similar (for example, trees of North Fargo, rocks of western Idaho, etc.). On this screen, the player selects a name, description, and keywords to search on for their new project.

If the user clicks the Create Project button, checks are performed to make sure they completely filled out the dialog. If these checks are satisfactory, the client sends a *CreateProject* event to the server and moves the user to the Invite Users page where new group members can be invited. If they click the cancel button and no "course" cookie exists, they are moved back to the Project View page. If the course cookie does exist, the faculty player is returned to the course view page.



Create Project

Fill out the following fields for your new project.

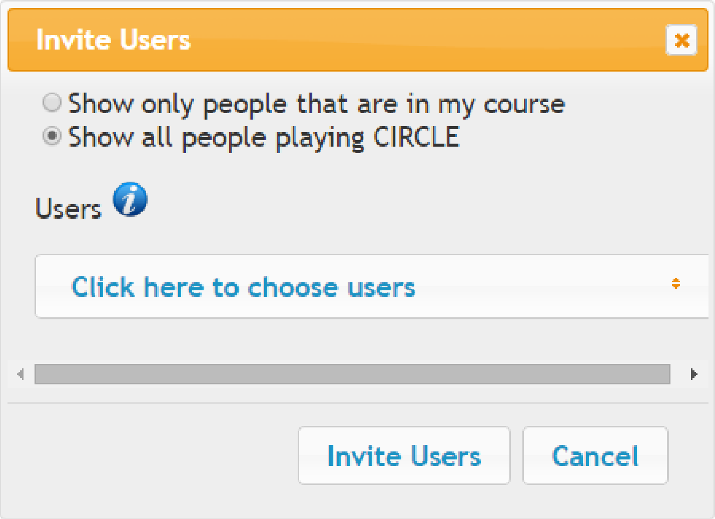
Project Name	
Project Description	
Keywords	

Figure 15. The create project screen. The (i) icons can be clicked for relevant tutor information.

Invite Users

This screen allows players to search for and invite other users to join the project (Figure 16). This is either opened from within the Project View page or displayed immediately after project creation. The interface allows the player to limit the user search based on certain characteristics, such as: within my course or all players in CIRCLE. Limits are in place for searching by state as well, but are removed for this release. Other limits can be implemented by modifying the GetUserListEventHandler class in JavaMOO.

If the user clicks on one of the limitation radio checkboxes, a *GetUserList* event is sent to the server which populates the user list drop down box. If the user clicks the Invite Users box, the client sends an *InviteUsers* event to the server with the user names that were selected.



Invite Users

○ Show only people that are in my course
● Show all people playing CIRCLE

Users

Click here to choose users

Figure 16. Screenshot of the invite users dialog box. Limitations on the search parameters are shown at the top of the dialog. If you click on the "Click here to choose users" button, a drop down appears with all of the users that have enrolled in CIRCLE.

Project Interaction

A project is a way to organize a set of similar objects that a group of individuals would like to classify and identify. The client instantiates this organization through a series of views covering the different tasks CIRCLE students must complete to reach the end of a project (Figure 17): Gathering, Elaborating, Classifying, Identifying, and Playing. Project Interaction views add two new modular components to the bottom of the interface: real-time chat and formative gamification.

Project Cache

One of the client's primary purposes is to retrieve information stored on the server and display it to users. The client accomplishes this through the creation of a series of caches, created in a javascript file called project_cache. These caches store information pertinent to objects in the project: specimens, experiments, multimedia, experiment results, specimen groups, and diagrams. These caches are refreshed on a timer controlled by the view's script.

The project cache is configurable through the use of a JSON specification. Options that can be set include specimens to load, specimen groups to load, and whether to load individual object types.

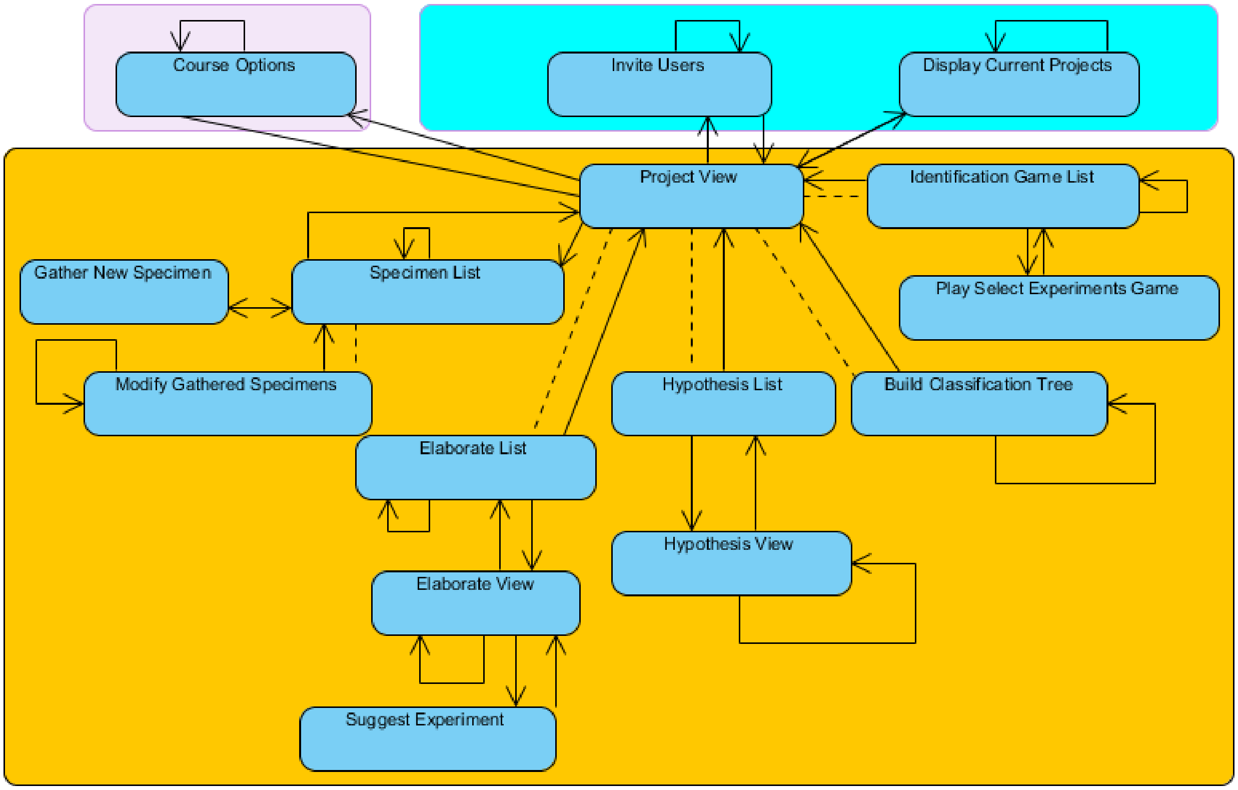


Figure 17. State diagram for project interaction (orange) with links to the faculty pages (lavender) and course selection and creation areas (blue). Dashed lines indicate transitions that require a prerequisite to be fulfilled before the state transition is allowed.

Rather than load all of the specimens or specimen groups in a project, certain pages only need access to a single specimen or specimen group, or to a limited subset of specimens and specimen groups. This can be accomplished by sending an array of specimen or specimen group IDs to the project cache, indicating which need to be loaded.

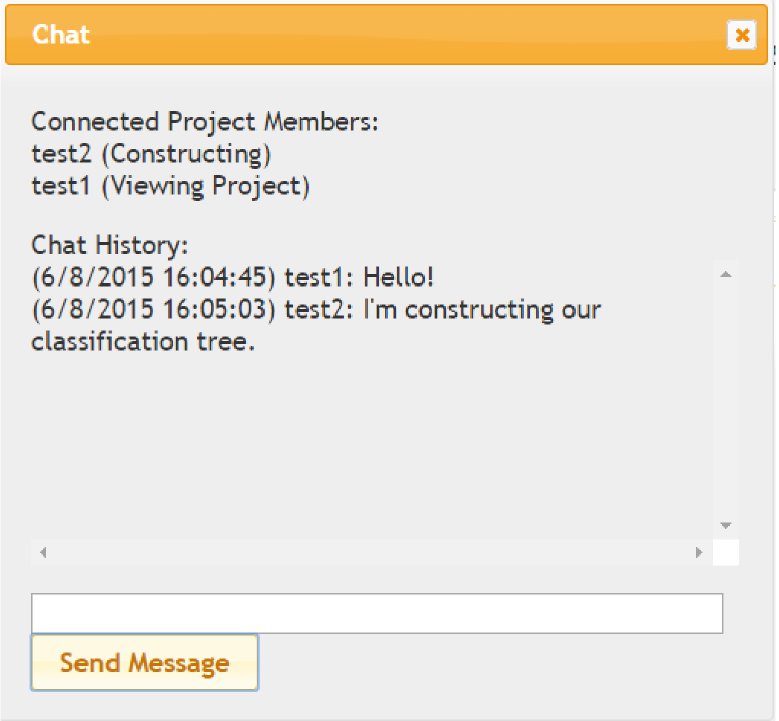
Certain pages also don't need all of the information a project cache could provide. For example, when loading the specimen gathering pages, it is not necessary for the client to load the project's diagram or experiments. These limitations can be invoked by adding fields to a "noload" array in the JSON specification. Valid limitations include "multimedia", "experiments", "diagrams", and "specimen groups".

The project cache starts by determining which events need to be sent based on the "noload" specification. Potential events to be sent include *GetSpecimens* (to load all of the specimens in the project), *GetSpecimen* (loads individual specimens), *GetMultimedia* (loads individual multimedia), *GetSpecimenGroups* (loads all specimen groups in the project), *GetSpecimenGroup* (loads an individual specimen group), *GetExperiment* (retrieves information associated with an individual experiment), and *GetExperimentData* (retrieves information associated with the results of an experiment). While events are being sent and their corresponding responses received, the caches are locked. Any access of the cache will result in an old copy of the cache being read by the client instead. Specimen groups refer to sets of specimens that have the same identity and will be more fully explained in the classification tree section.

Chat

A chat interface is provided for real-time text-based communication between players in a project (Figure 18). This chat interface can be turned on and off by instructors via restrictions applied in the course view page. The chat window itself is divided into two areas. The upper area shows the players that are currently connected to the server and working on this project. The lower area is a history of all of the chat messages that have been sent and received since the last time the player logged in with an enforced buffer limit to avoid memory issues.

All project level pages include a *GetChat* and *GetActivePlayers* event set on a timer that continuously updates the chat interface. *GetChat* retrieves all of the chat messages that have been sent since the last request. *GetActivePlayers* returns a list of all of the players currently working on this project and what task they are doing.



Connected Project Members:
test2 (Constructing)
test1 (Viewing Project)

Chat History:
(6/8/2015 16:04:45) test1: Hello!
(6/8/2015 16:05:03) test2: I'm constructing our classification tree.

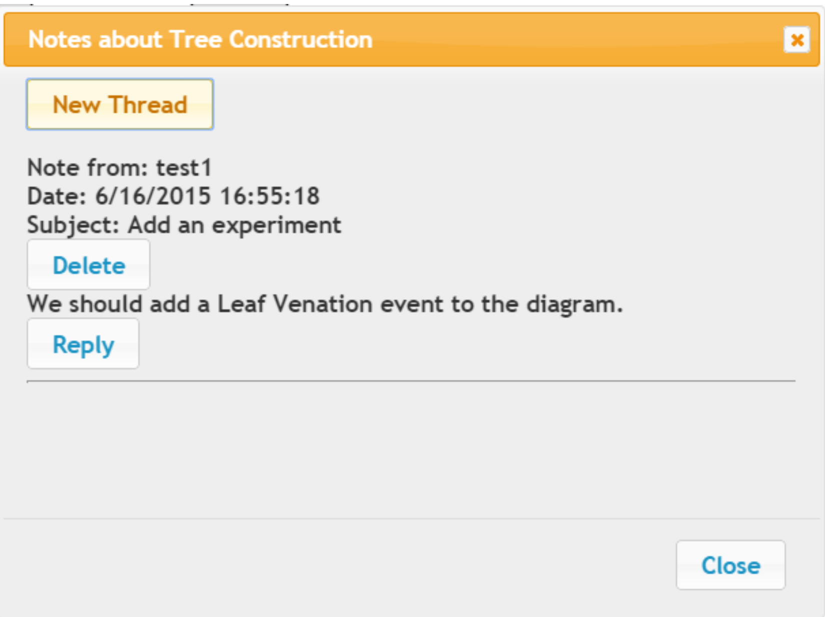
Figure 18. The chat interface. Logged in users and the task they are working on are shown at the top of the dialog. History of chat messages are shown below. Players enter new messages in the text field at the bottom.

Multiple indications for new incoming chat messages exist. When receiving a message, a beep is sounded. Upon receiving more than 10 chat messages without reading them, a tutor is sent to the player to remind them that they have missed chat messages.

Bulletin Boards

Asynchronous communication is implemented via a bulletin board system (Figure 19). The design for this implementation is based on prior work (Borchert 2008). Bulletin boards are available for each suggested experiment and hypothesis, and during tree construction.

Events used during bulletin board interaction include: *GetBulletinBoard*, *GetBoardHeaders*, *GetNoteContents*, *WriteNote*, *DeleteNote*, and *GetUnreadBoardMessageCount*. Each experiment, hypothesis, and classification tree has an associated bulletin board loaded with *GetBulletinBoard*. The *GetUnreadBoardMessageCount* event returns the number of unread messages on the bulletin board for display on the interface. The *GetBoardHeaders* event loads the subject lines of all of the notes on the board. If a user clicks the Read button, the *GetNoteContents* message is sent and returns the contents of the note. Players remove notes by clicking the Delete button, which sends a *DeleteNote* event. The Reply and New Thread buttons open the New Note interface (Figure 20). Upon clicking the Write Note button, a *WriteNote* event is sent with the subject and contents of the note.



Note from: test1
Date: 6/16/2015 16:55:18
Subject: Add an experiment

We should add a Leaf Venation event to the diagram.

Figure 19. The bulletin board interface showing a classification tree board.

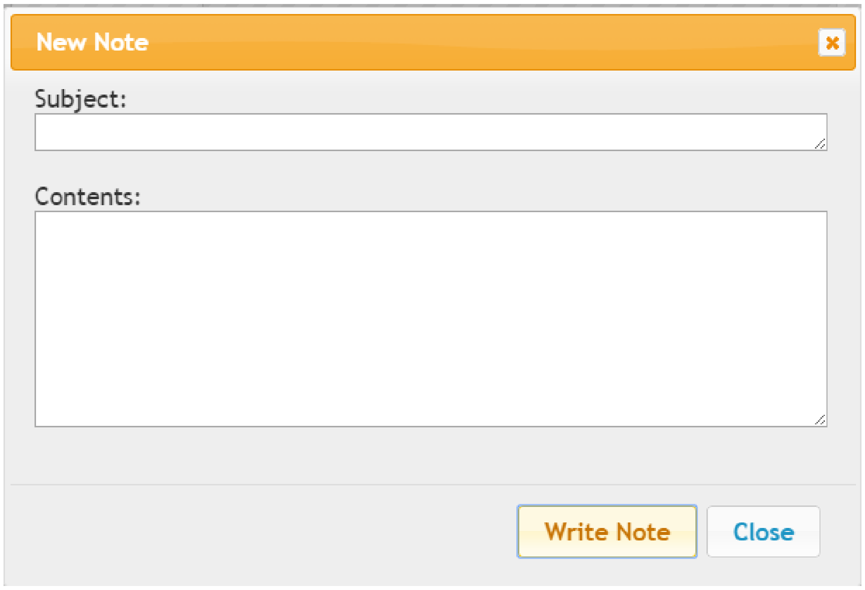
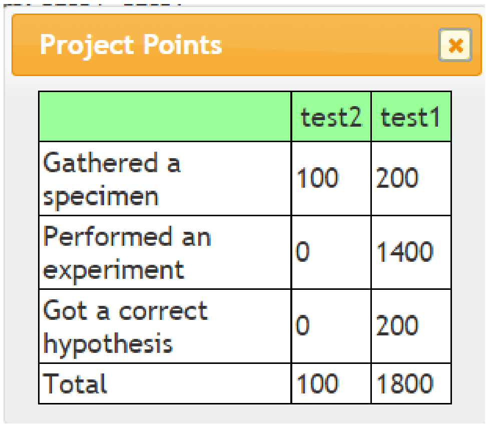


Figure 20. Writing a new note in the bulletin board interface.

Formative Gamification

Points are awarded to players when they accomplish a variety of tasks. These tasks include gathering a specimen, uploading a multimedia object associated with a specimen, suggesting an experiment, performing an experiment, performing a tree modification, suggesting the correct hypothesis for a specimen, moderating an experiment, getting a high score in one of the game templates, and playing one of the game templates. We call this formative gamification (Borchert et al. 2015). A *GetGamificationStats* event is set on a timer to periodically update the gamification window. In addition, a tutor is dispatched any time another player in the group receives formative points. This system can be turned on or off via the restrictions interface.

The gamification interface (Figure 21) consists of a table showing the players in the project and the number of points received for each task. The purpose of this screen is to provide individual accountability as described in the literature review. New tasks can be added in the Achievement class on the server.



Project Points

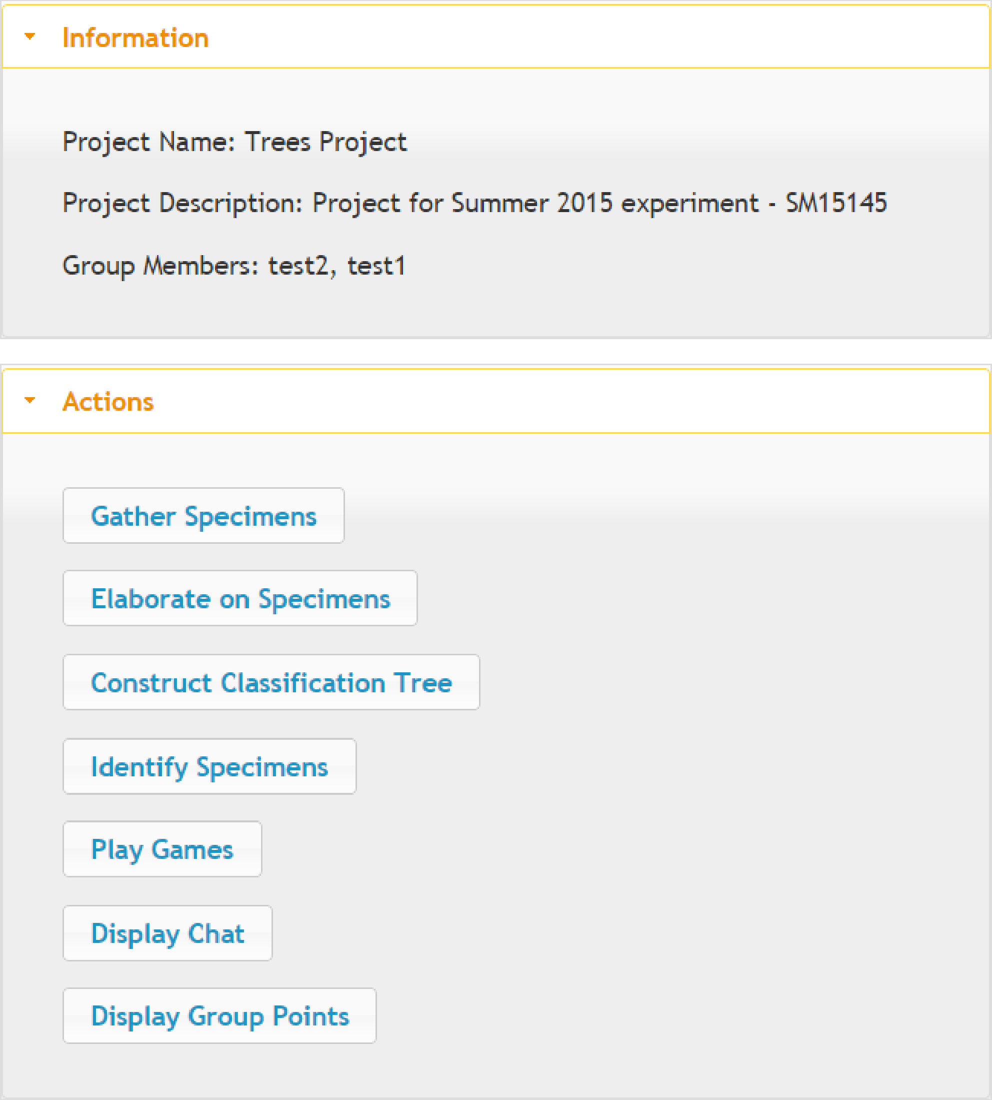
	test2	test1
Gathered a specimen	100	200
Performed an experiment	0	1400
Got a correct hypothesis	0	200
Total	100	1800

Figure 21. The gamification interface. Test2 has gathered one specimen and received 100 points, while test1 has done all the rest of the work.

Project View

This view displays the player's current tasks (Figure 22). These tasks include specimen collection, specimen elaboration, classification tree construction, hypothesis formation, and identification game playing. To support user scaffolding, tasks are slowly introduced to the player as progress is made.

Before loading the project view screen, a series of events are sent to the server. The *BeginTask* event updates the player's current task as "Viewing Project". *GetProject* retrieves the project's name and description. A sequence of events is then sent to determine which portions of the interface are available for players to use. *GetSpecimens* returns a list of all specimen IDs in this project. *GetSpecimen* is sent for every specimen in the project and returns the name of the specimen and a list of experiments and hypotheses performed on that specimen. *GetMinimumSpecimensToGather* returns the number of specimens that must be gathered and experimented on before tree construction will be unlocked. The *IsTreeValid* event determines if identification should be unlocked by checking if the tree is valid. If all specimen names are not "Unknown", game play is unlocked.



Project Name: Trees Project

Project Description: Project for Summer 2015 experiment - SM15145

Group Members: test2, test1

Figure 22. A fully unlocked project view.

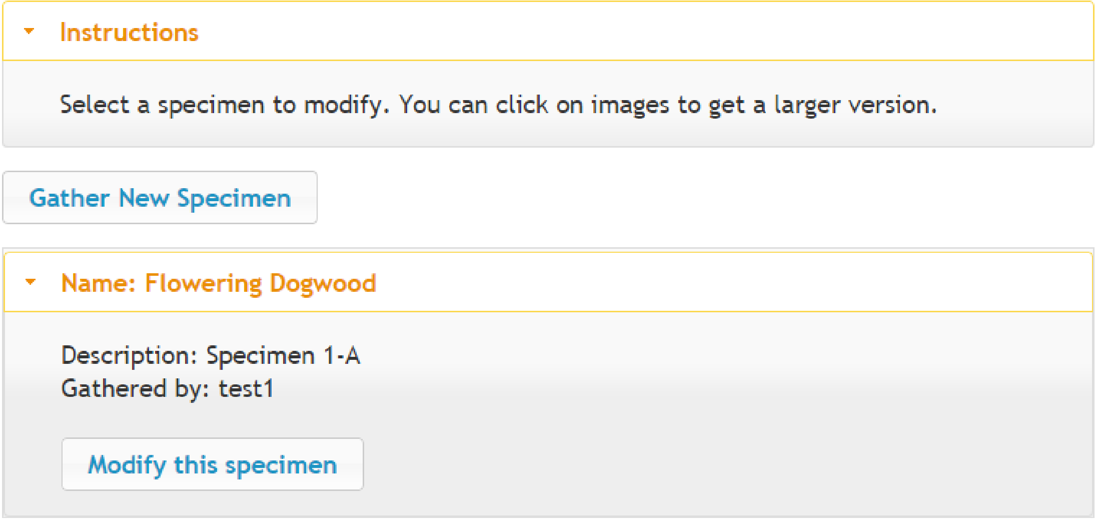
Clicking on a locked button results in no action. Clicking on one of the task buttons moves the player to the corresponding task page. By clicking the Invite Users button, players can go to the invite users page to add new users to the project.

Specimen List

The specimen list page displays the current project’s specimens with associated metadata including name, description, gatherer, and associated multimedia (Figure 23). If the connected player is the gatherer, there are buttons for deleting the specimen and modifying the specimen.

These buttons store the specimen ID in the ‘specimen’ cookie for use by later pages. Additionally, there is a button that moves the player to an interface where they can gather a new specimen.

Upon entering this page, a series of events are sent to the server to populate client interfaces. *BeginTask* with task set to “Gathering” allows other group members to know what this player is doing. The project cache is used to load all specimens and their associated multimedia, but does not load the diagram, experiments, or specimen groups.



Instructions

Select a specimen to modify. You can click on images to get a larger version.

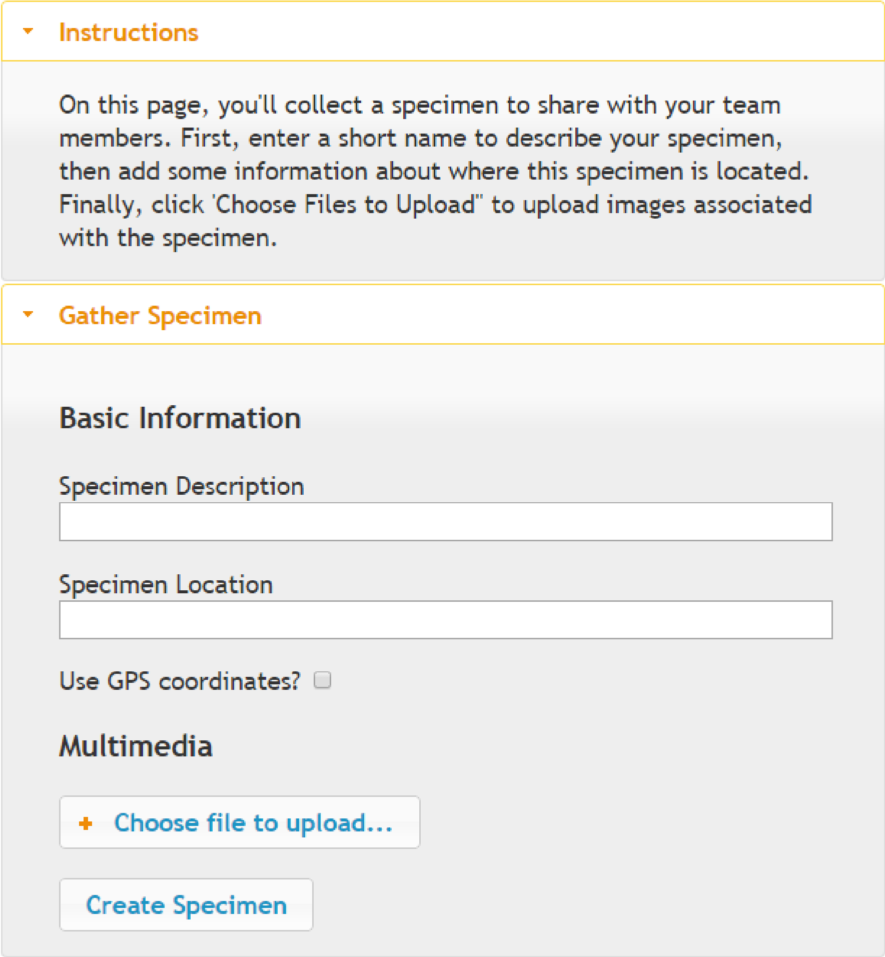
Name: Flowering Dogwood

Description: Specimen 1-A
Gathered by: test1

Figure 23. Specimen list view with one identified specimen.

Gathering and Modifying Specimens

The “gather a new specimen” and “modify an existing specimen” pages are similar, the primary difference being the events sent during each task. In both views, the player is asked to enter a description, location (via the device supplied GPS coordinates, or a user-editable text field), and relevant multimedia associated with the specimen (Figure 24). The *AddMultimedia* event is used to add multimedia to the game in both pages.



On this page, you'll collect a specimen to share with your team members. First, enter a short name to describe your specimen, then add some information about where this specimen is located. Finally, click 'Choose Files to Upload" to upload images associated with the specimen.

Figure 24. A screenshot of the gather new specimen interface.

In the modify specimen page, the client sends *GetSpecimen* and *GetMultimedia* events to retrieve the current description, location, and multimedia artifacts associated with the specimen in the ‘specimen’ cookie. In addition, the *ModifySpecimen* event is used to change the existing specimen. *DeleteMultimedia* can be used to delete multimedia associated with the specimen. On the gather new specimen page, the *CreateSpecimen* event is used to create and initialize new specimen on the server.

Players are allowed to add as much multimedia content to an object as they wish. Audio and video are given a default open source icon. If no multimedia is uploaded, none will be displayed by the client.

Multimedia content is uploaded to https://circle.cs.ndsu.nodak.edu/game/uploads/ via a separate perl script integrated with the JQuery Fileupload plugin discussed earlier. This script avoids name collisions by renaming multimedia that has the same name to include a "(#)" where # is an increasing integer that stops name collisions.

Elaborate List

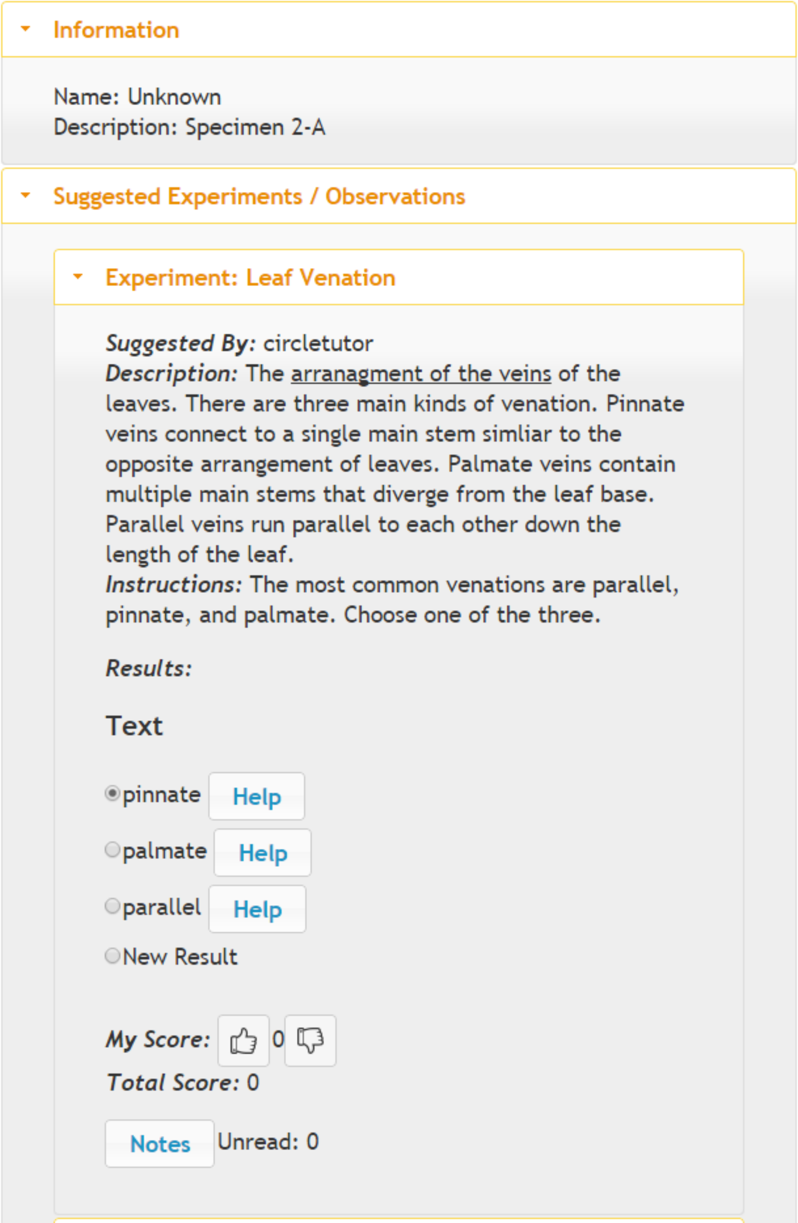
The elaborate list page displays the current project's specimens with each name, description, and associated multimedia displayed next to a button that opens the elaborate view page. It also indicates which specimens still have experiments yet to perform via red text displayed next to the specimen. The list is visually similar to the specimen list view.

When entering this page, the client sends a *BeginTask* event to start the 'Elaborating' task. The project cache loads the specimens, multimedia, and experiments, without including the diagram or specimen groups.

On this page, players are allowed to delete specimens if they were the one that gathered the specimen. They are also allowed to select a specimen to work with. If they click on one of these specimen buttons, the 'specimen' cookie is set to the ID of the specimen and they are sent to the Elaborate View page.

Elaborate View

In the elaborate view (Figure 25), the player is able to see the list of current experiments and observations performed on this specimen. Any player that has not gathered this specimen is allowed to suggest new experiments (which opens the suggest experiment interface). Only the player that has gathered the specimen is allowed to perform or edit experimental results. This separation creates positive interdependence, because one person is required to suggest experiments and observations, while the other is required to perform those experiments or observations. Additions included for the second experiment allowed the system to suggest experiments to perform.



Information

Name: Unknown
Description: Specimen 2-A

Suggested Experiments / Observations

Experiment: Leaf Venation

Suggested By: circletutor
Description: The arranagment of the veins of the leaves. There are three main kinds of venation. Pinnate veins connect to a single main stem simliar to the opposite arrangement of leaves. Palmate veins contain multiple main stems that diverge from the leaf base. Parallel veins run parallel to each other down the length of the leaf.
Instructions: The most common venations are parallel, pinnate, and palmate. Choose one of the three.

Results:

Text

● pinnate
○ palmate
○ parallel
○ New Result

My Score: 0
Total Score: 0

Unread: 0

Figure 25. A portion of the elaborate view interface as seen by the gatherer. A pinnate leaf venation has been selected.

Players are also able to moderate experiments, that is, they are able to suggest which experiments they think will be the most and least useful when identifying the specimen. This is accomplished by sending a *ModerateExperiment* event to the server when clicking on the thumbs up or thumbs down button in the interface.

Upon entering this view, the client loads the ‘specimen’ cookie’s information via the project cache. In addition, the *GetAllExperimentData* event retrieves all experimental results for all experiments in the project to populate the results of each experiment in the list.

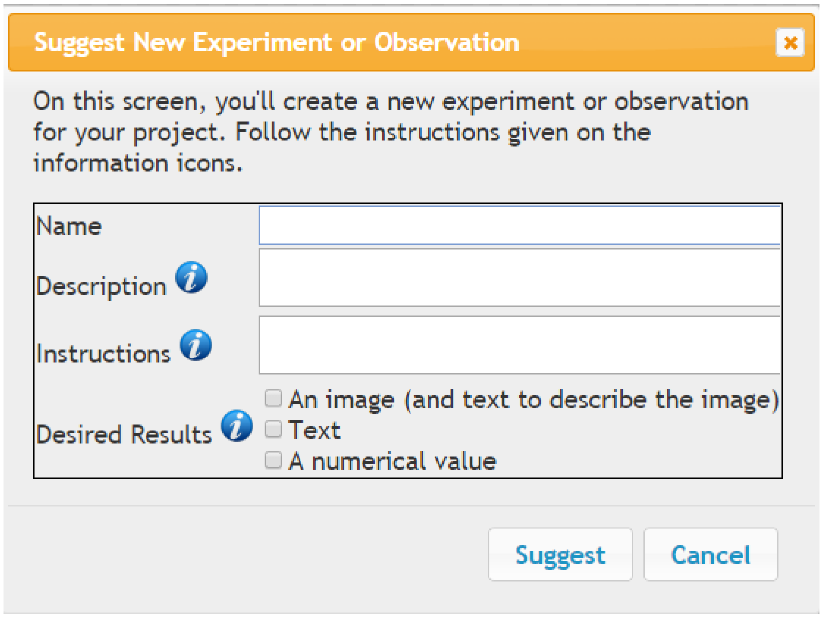
There are two options at this point. If the player is not the one who gathered the specimen, they suggest an experiment via an interface button that leads to the Suggest Experiment view.

If they have gathered the specimen, and experiments have been suggested, either by other players or by the game itself, they can perform the experiments on this page by clicking on radio buttons associated with individual results. The client sends a *PerformExperiment* event when a user clicks on one of these radio buttons. The gatherer can also click a New Results radio button, allowing them to add new results to the experiments. This sends an *AddExperimentData* event to store the new experimental result and a *PerformExperiment* event to associate the new result with the specimen.

Suggest Experiment

On the suggest experiment page, players either suggest an existing experiment that was previously defined for another specimen within this project or create a brand new experiment to perform. New experiments are defined by a name, description, and what types will be used as data for the experiment (Figure 26). Players can choose to include text, numbers, audio, video, images, or any combination of those types as data to represent the results of the experiment.

The view begins by sending a *GetSpecimen* event to get information about the specimen and a *GetAllExperiments* event to get all of the experiments performed on all specimens within the project to populate a drop down menu to suggest previously entered experiments.



Suggest New Experiment or Observation

On this screen, you'll create a new experiment or observation for your project. Follow the instructions given on the information icons.

Name

Description

Instructions

Desired Results
□ An image (and text to describe the image)
□ Text
□ A numerical value

Figure 26. The suggest new experiment dialog. The (i) icons can be clicked for relevant tutor information. Students must enter a name, description, instructions, and result types.

If the experiment is brand new, an *AddExperiment* event is sent to store the experiment in the project. Once the experiment has been added or if the experiment already exists in the project, a *SuggestExperiment* event is sent to link this experiment with the current specimen.

Construct Trees

The diagram was originally implemented completely server-side with D3.js as the client-side diagramming API. In the D3 implementation, users clicked individual nodes in a classification tree, then selected context-dependent options to fully describe the tree (Figure 27). A tree started as a single empty node. When clicking on an empty node, players could add an experiment or an identified specimen to replace that empty node. When clicking on an experiment, players could add any of the results of that experiment. This result would be added to the tree as a child of the experiment and an empty node would be placed as a child of the result. In addition, players could delete the experiment node, or move the node to a different spot on the tree or to one of the “scratch” trees (an area for testing alternative tree arrangements). If the player clicked on a specimen, they could delete or move the specimen. In addition, if the specimen was at the root of the tree, they could add an experiment/pair as a parent of the specimen.

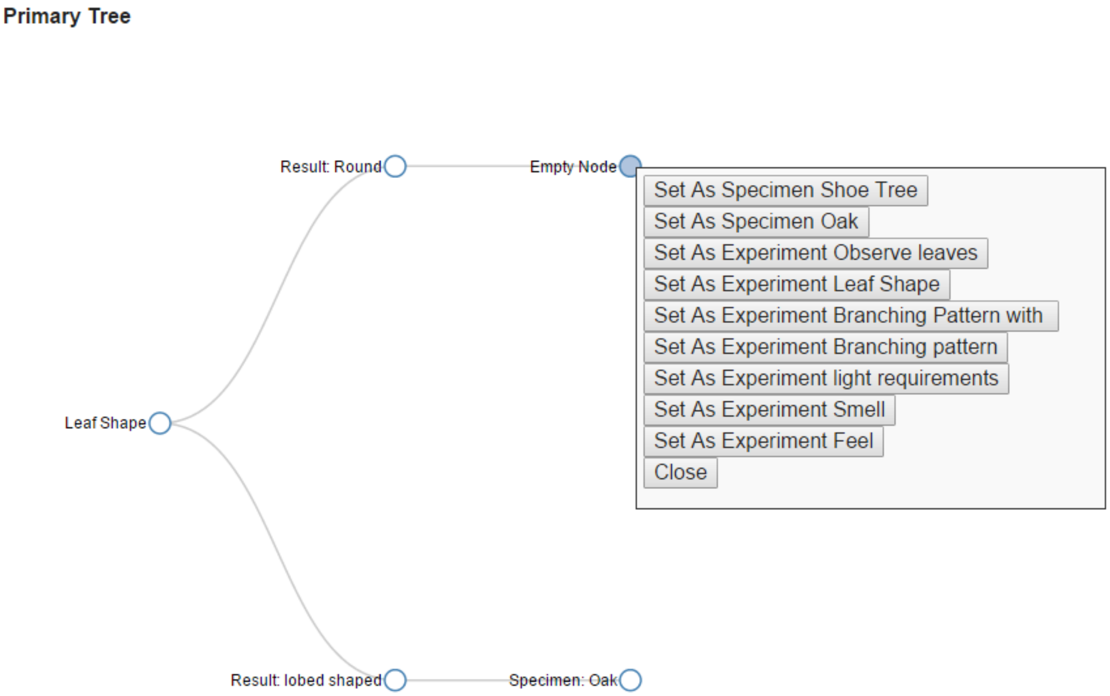

Primary Tree
Result: Round
Empty Node
Leaf Shape
Result: lobed shaped
Specimen: Oak
Set As Specimen Shoe Tree
Set As Specimen Oak
Set As Experiment Observe leaves
Set As Experiment Leaf Shape
Set As Experiment Branching Pattern with
Set As Experiment Branching pattern
Set As Experiment light requirements
Set As Experiment Smell
Set As Experiment Feel
Close

Figure 27. A classification tree using the D3 interface. The player has clicked the empty node and is given a list of all experiments and specimens in the project to choose from. The root of the tree contains an experiment with player-added results.

It was determined during the first beta test and experiment that changes were needed to improve the interface. The node clicking interaction wasn’t intuitive, so GoJS was identified as a solution that supported more diagram-like functionality, including drag and drop support, easy node creation and linking, and a basic tree layout. In addition, animation was implemented using

TWEEN.js. This allows students to simulate identifying a specimen using their tree. Specimens follow a path from the start node to each of the experiments, following the result that corresponds to the specimen until a “leaf” node is reached, indicating the identity of the specimen. This method also allows multiple specimens to have the same identification from the same set of experimental results, which was not directly supported in the D3 version.

The GoJS diagram interface (Figure 28) is split into three main areas: a palette, a diagram, and an information panel. The palette shows the nodes that are currently available for the player to add to the diagram. There are two different palette states. If the start node has not been added, it is the only item in the palette. This introduces the user to dragging and dropping items into the diagram. If the start node has been added, the palette includes all specimens and experiments in the project.

Each node type has a different action when dropped into the diagram area. When the start node is added, a link with an attached empty grey node is added to the diagram. Grey nodes, called containers, represent the leaf nodes of the tree. When an experiment is dropped into the tree, all results for that experiment are automatically added as links out of the experiment node. A text label on the link displays the name of the corresponding result, and the client attaches a container to the link. If the experiment is dropped into a pre-existing container, it replaces the container and is integrated into the tree hierarchy. If it is dropped elsewhere in the diagram, an error message is displayed to the user, indicating that they should attach the experiment to the tree structure. This disconnected experiment can be used for scratch space and eventually reattached to the diagram when the player is ready. Specimens automatically animate towards the Start node when they are dropped in the diagram area. Upon reaching the start node, the specimen follows the link to the first experiment if one exists. The client then checks to see which of the experimental results matches the one performed on the specimen, and animation resumes along the link that matches that result. This continues until the specimen reaches a container, at which point it is added to the container. If at any point, an experiment has not been performed on the specimen, an error message is displayed to the user, a new Error child link and container are created, and the specimen animates to the error node.

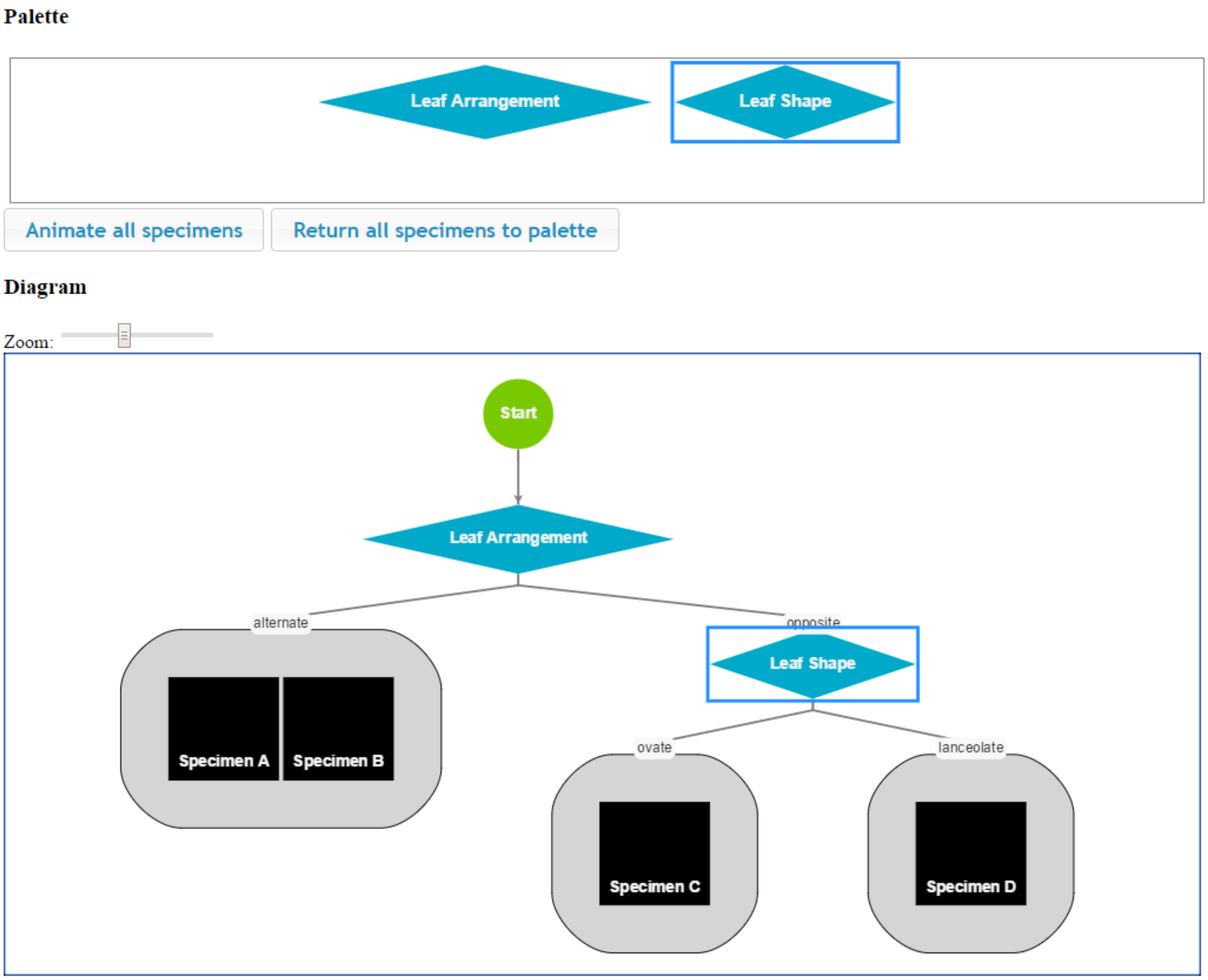


Figure 28. A fully specified tree in the GoJS platform. The palette is shown above with two experiments, leaf shape and leaf arrangement. The diagram below shows the results after clicking the “Animate all specimens” button. Specimens A and B have an alternate leaf arrangement, Specimen C has an opposite leaf arrangement and an ovate leaf shape, Specimen D has an opposite leaf arrangement and a lanceolate leaf shape. The leaf shape experiment is selected.

The Animate All Specimens button performs this drag and drop algorithm on all specimens at once, while the Return all Specimens to the Palette button does the reverse, placing all specimens in the diagram back into the palette.

The client displays data associated with a clicked node in the information panel. For specimens, this includes the name and description of the specimen and the experimental results for that specimen. For experiments, this includes the experiment's instructions and description, and a list of all results within the experiment and the number of specimens with each result. If a container containing specimens is clicked, it will display all of the experiments performed on all specimens in the container with the associated results and the number of specimens in each result. This allows easy identification of which experiments might be used to distinguish between specimens in that particular container node. The container information also includes an option to return all specimens in the container to the palette. This option makes it easier to replace the container node with an experiment node.

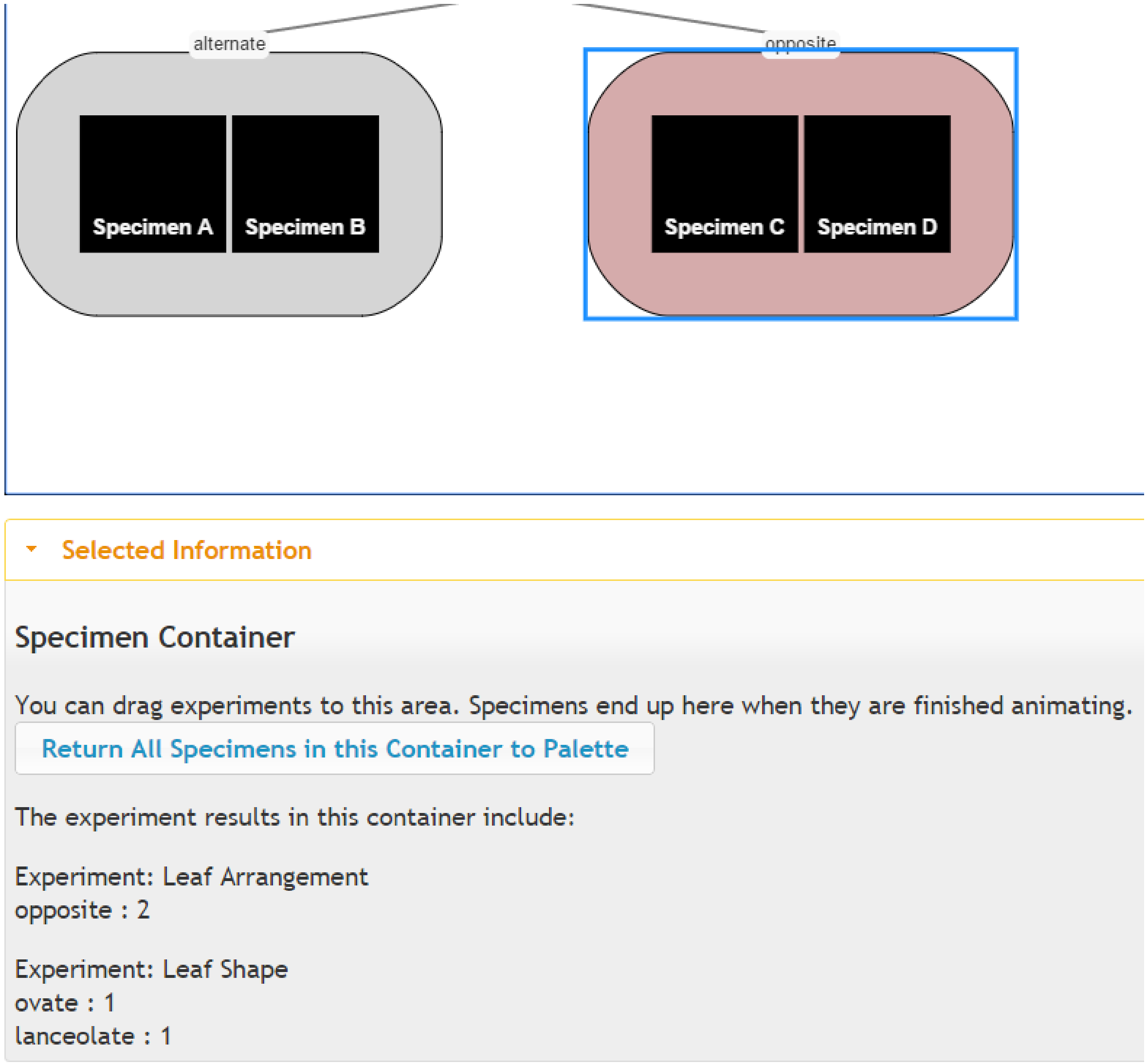
One limitation of GoJS is that the diagram is implemented client-side. This necessitated the creation of a locking mechanism, so that the diagram would remain in a stable, consistent state. Only one player is allowed to edit the diagram at a time, and must request a token from the game in order to perform edits. Players currently editing the diagram may release the lock to the game or to other players. Events used to support this interface include: *ReleaseDiagramLock*, *RequestDiagramLock*, *TransferLock*, and *GetDiagramLockState*. The diagram can only be saved, using the *SaveDiagram* event, by the person holding the locking token. In addition, only the token holder is allowed to click the Reset the Diagram button, which empties the diagram and returns the palette to containing only the Start node.

Specimen locations in the diagram are removed before storage on the server. This allows all users, not just the one with the token, to animate specimens along the tree and to return the specimens to the palette.

In order to continue to the next stage of CIRCLE, the tree must be deemed valid by the software. This ensures that game templates can be created using the user-generated tree. In the D3 version, the tree was valid if the following was true: 1) No empty nodes may be present; 2) All specimens in the project must be represented by at least one node; and 3) The path to each specimen must represent correct experiments and result. This validity check was performed server-side by completing the following steps: 1) Check the tree for empty nodes using a recursive algorithm; 2) Get all specimens in the tree using a recursive algorithm, matching them with the project's specimens; 3) Use GetParent on the specimens until parent is null to get the path for each specimen, then check each result on the path to ensure validity.

The GoJS version added the requirement that specimens might or might not be different in an individual specimen group. Nodes with specimens that have different experimental results are highlighted, indicating that it may be necessary to add an experimental node to differentiate the two specimens (Figure 29).

Tree validation in GoJS is somewhat simplified: 1) The server keeps track of the set of all (experiment+result) paths through the current diagram, beginning with the start node; 2) After creating this list of paths, specimens are assigned to the path that corresponds to its list of experimental results; 3) If an experiment has not been performed on a specimen, yet it is needed in order to progress on a path, it is marked as an error. The player must perform that experiment before the tree will be valid; 4) If there are no experiments on the diagram, the tree is marked as invalid; 5) On the client, after animation is complete, specimens are checked to determine if they have the same experimental results. If not, the container's color is changed (Figure 29). This does not mean that the tree is invalid. The player should check to make sure that this experimental result is not important towards distinguishing the specimens.



Specimen Container

You can drag experiments to this area. Specimens end up here when they are finished animating.

The experiment results in this container include:

Experiment: Leaf Arrangement
opposite : 2

Experiment: Leaf Shape
ovate : 1
lanceolate : 1

Figure 29. A partially constructed tree in the GoJS platform. The pink container indicates that the two specimens have different experimental results. In the information panel below, it can be seen that one specimen has an ovate leaf shape, while the other has a lanceolate leaf shape.

The tree construction view begins by getting all of the project data, specimens, experiments, experimental data, diagram and hypotheses via the project cache. It also sends a BeginTask event with a "Constructing" task to indicate the task switch to other project members.

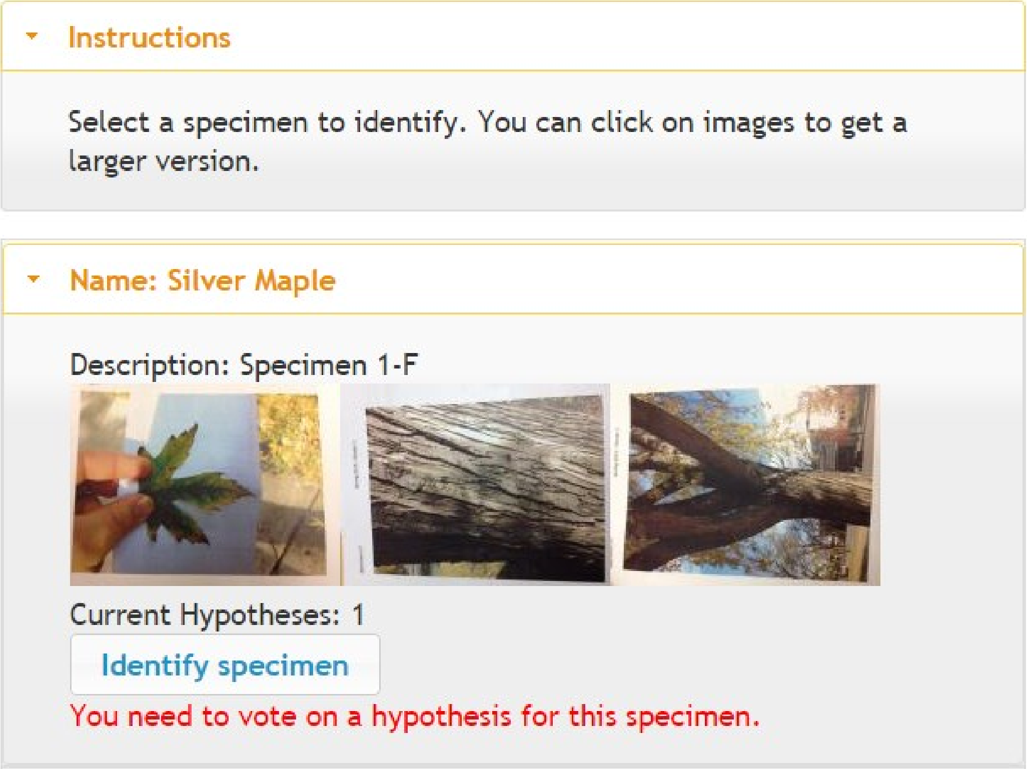
While the classification tree interface was being developed, a number of libraries were identified and reviewed. D3 (http://d3js.org/) was selected because of the free cost, good online documentation, swift tree implementation, and robust platform for visualization. While GoJS is quite expensive, they do offer a free license for educational use and have a vast assortment of examples and a well-documented API. Other reviewed packages included diagramo (http://diagramo.com/), jointjs (http://www.jointjs.com/), and jsdiagram (http://mindfusion.eu/features-jsdiagram.html), which were all either too expensive or lacked good examples or online support.

Hypothesis List

The classification tree task can result in multiple specimens having the same set of experimental results. These specimens with common experimental results are placed into a construct called a specimen group. The hypothesis list screen displays all the specimen groups that are available for identification (Figure 30). Unidentified specimens are listed first. The specimen's name, description, and multimedia are displayed next to a button that takes the player to the hypothesis view page when clicked. Unidentified specimens are called "Unknown", otherwise, the specimen name is displayed.

Prior to loading the view, the client sends a BeginTask event with an "Identifying" task, then uses the project cache to load information related to the specimen groups.

For each specimen group, a button labeled "Identify Specimens" can be clicked. This button sets the 'specimen group' cookie to the reference of the specimen group and the user continues to the Hypothesis View page.



Instructions

Select a specimen to identify. You can click on images to get a larger version.

Name: Silver Maple

Description: Specimen 1-F

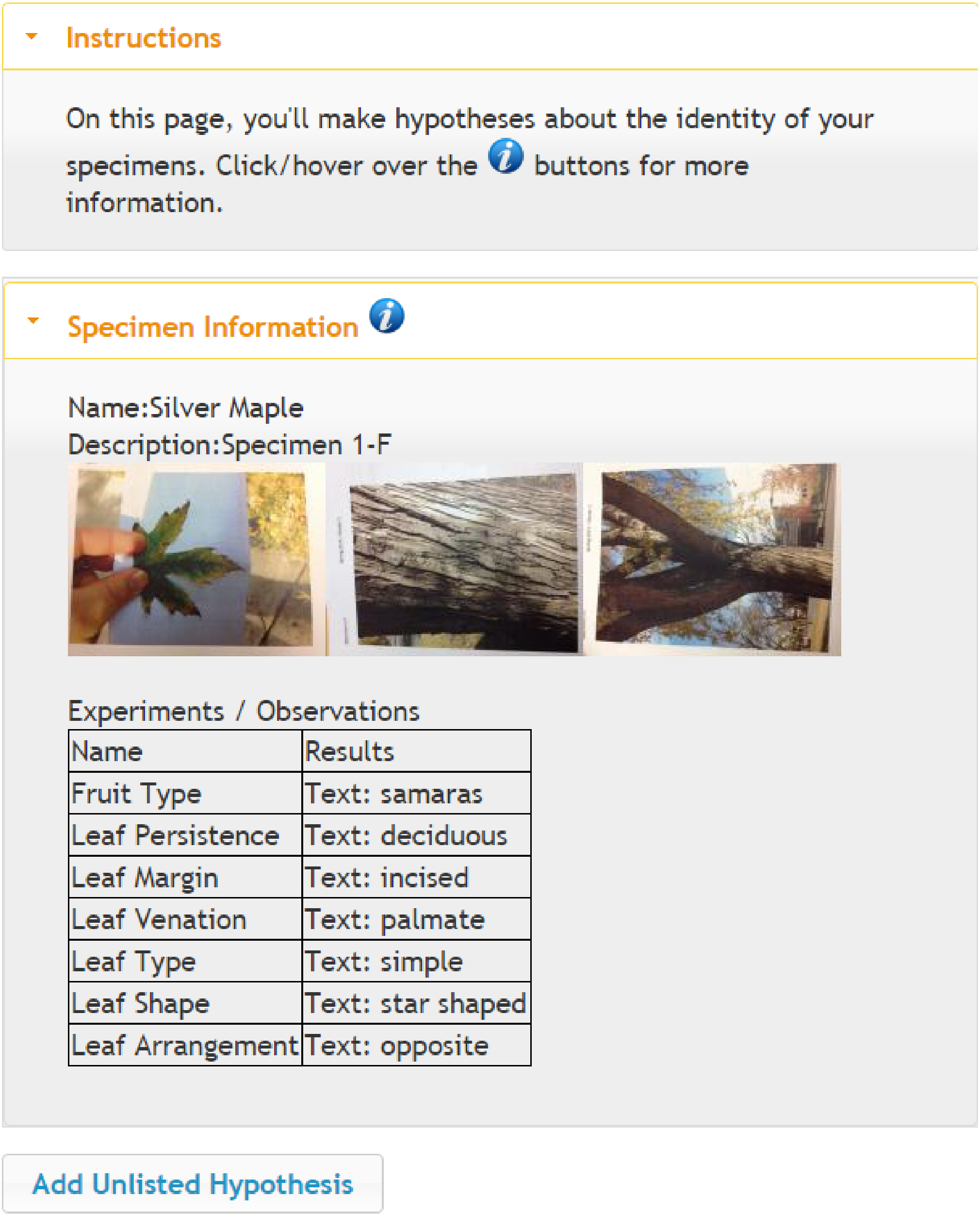
Current Hypotheses: 1

You need to vote on a hypothesis for this specimen.

Figure 30. A screenshot of the hypothesis list page with red text indicating to the user what they need to do in order to progress.

Hypothesis View

Players identify specimen groups on the hypothesis view page (Figure 31). Players either create a new hypothesis for the identity of the specimen or use an existing project hypothesis. By using an existing hypothesis, players indicate that two specimen groups have different experiments and results but have the same identity. When adding a hypothesis, the student must give a reason for their suggestion. Students also moderate hypotheses, casting a vote for whether they agree that the identification hypothesis is correct. Once more than half the group agrees on a hypothesis, that hypothesis becomes the specimen's name. Students see the number of votes provided by others via Peer Help (described below), although these votes do not count towards the right to name the object. Players that create a hypothesis can delete that hypothesis, but only if the hypothesis has not become the object's identity. The project cache loads the specimen group's information.



Instructions

On this page, you'll make hypotheses about the identity of your specimens. Click/hover over the ⓘ buttons for more information.

Specimen Information ⓘ

Name:Silver Maple
Description:Specimen 1-F

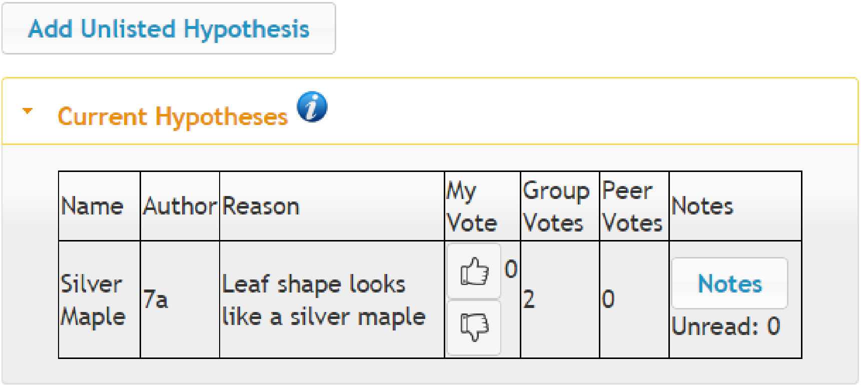
Experiments / Observations

Name	Results
Fruit Type	Text: samaras
Leaf Persistence	Text: deciduous
Leaf Margin	Text: incised
Leaf Venation	Text: palmate
Leaf Type	Text: simple
Leaf Shape	Text: star shaped
Leaf Arrangement	Text: opposite

Figure 31. The top half of the hypothesis view page showing information about this specimen group that would be useful to know before suggesting hypotheses.

Players click the "Add Unlisted Hypothesis" button (Figure 32), which opens a dialog box (Figure 33) that allows them to suggest an existing hypothesis (as retrieved by the

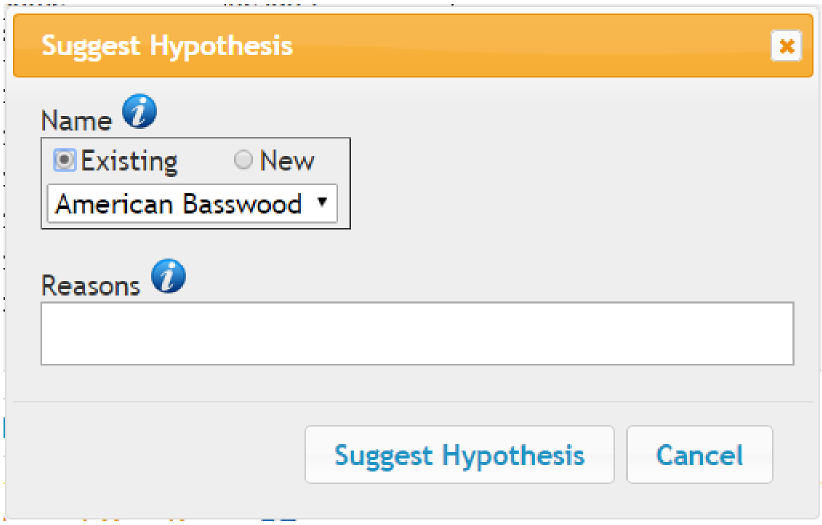
GetAllHypotheses event) or a new hypothesis. In both cases, they are required to enter an explanation for why they chose that hypothesis. An *AddHypothesis* event is sent if the hypothesis has not been suggested in the project. The *SuggestHypothesis* event associates the hypothesis and player reasoning with this specimen group. In addition, players can click thumbs up and thumbs down to vote for the hypotheses they think are correct or incorrect. This action sends a *ModerateHypothesis* event. Users are only given a single vote for each hypothesis, they can either vote positively or negatively, giving it a +1 or a -1 vote. After all specimen groups have been positively identified, the game play activity is unlocked.



Current Hypotheses

Name	Author	Reason	My Vote	Group Votes	Peer Votes	Notes
Silver Maple	7a	Leaf shape looks like a silver maple	👍 0 👎	2	0	Notes Unread: 0

Figure 32. The bottom half of the hypothesis view page showing the current hypotheses for this specimen. The voting interface is shown as thumbs up and thumbs down buttons.



Suggest Hypothesis

Name

● Existing ○ New

American Basswood ▾

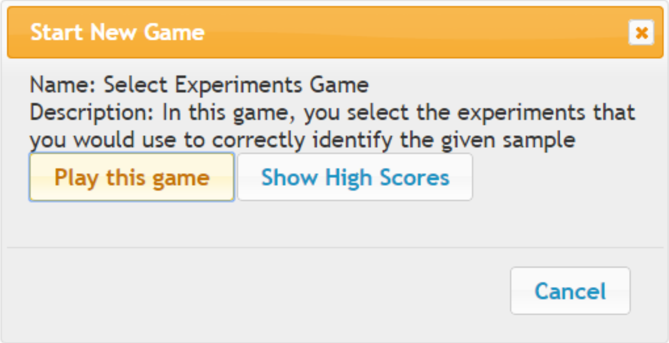
Reasons

Figure 33. A dialog box showing the interface for suggesting an existing hypothesis.

Game List

The game list screen enumerates the retrieval-based learning games that can be played by the player (Figure 34). Retrieval learning is enforced by not allowing players to start a new game until a certain time period has elapsed since the previous game finished. This amount of time is called game spacing and is determined by the course instructor. It is set at one day by default. Future experiments could be developed to determine the most effective length of time between play sessions.

Upon entering this view, the client sends a *GetAvailableGames* event which gets the games that are implemented on the server. A *GetActiveGames* event is also sent to check if there are any in-progress games and to allow the player to continue those games from prior save points.



Name: Select Experiments Game
Description: In this game, you select the experiments that you would use to correctly identify the given sample

Figure 34. The new game interface. The play this game button is enabled because the game spacing timer has elapsed.

A player starts a new game by clicking the Play this game button, which sends a *StartGame* event. The client then loads the specified game interface. The client also allows players to resume games they have previously started. The *GetGameLevel* and *GetGameState* events load the saved game state. Players can also stop previously started games on this page by clicking Quit this Game. This sends a *StopGame* event to the server, cleaning up game resources.

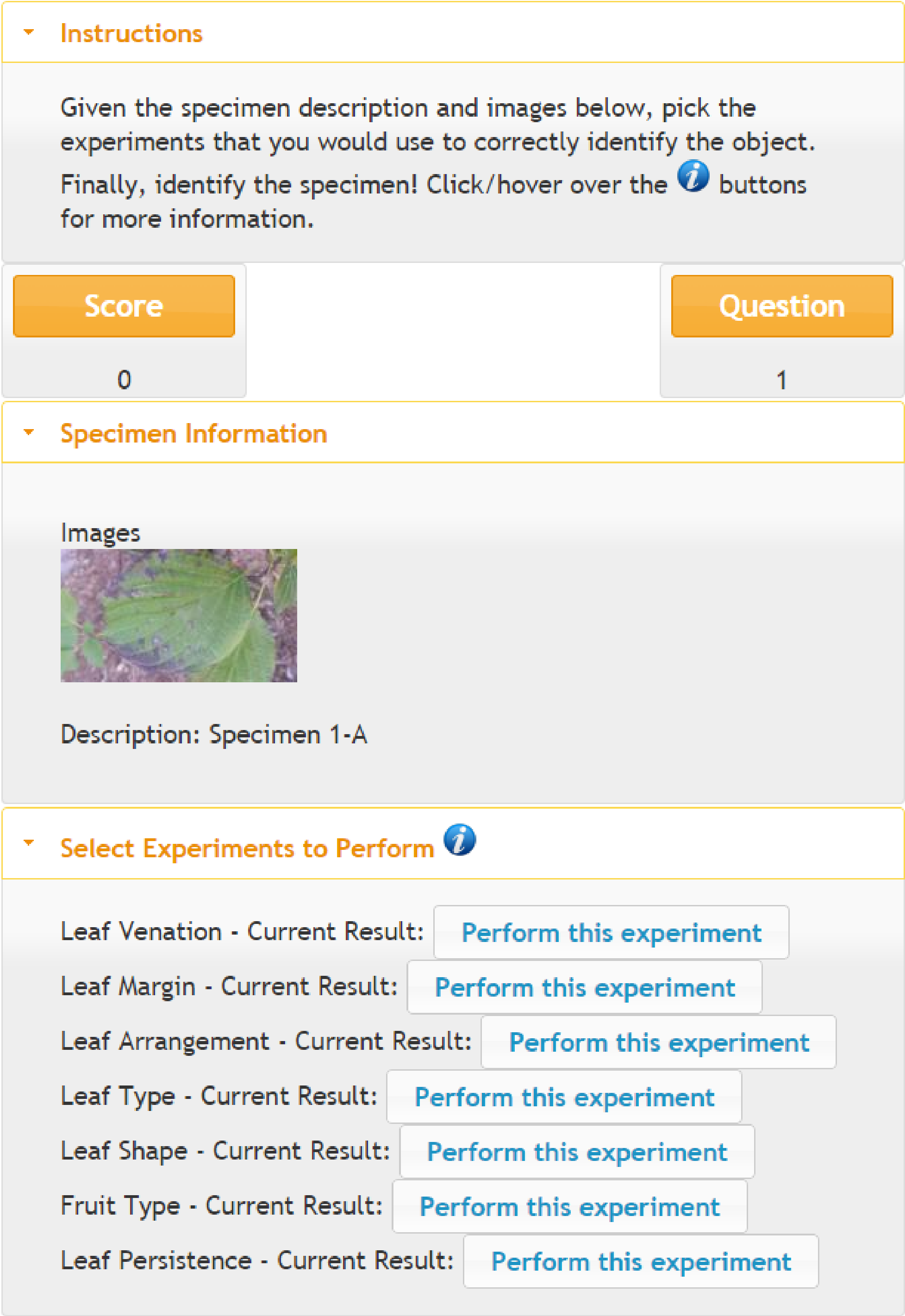
Players can also view the high scores for each individual game. This information is retrieved using a *GetGameHighScores* event. Only the top ten scores are currently stored by the server.

Play Select Experiments Game

Users play the Select Experiments game on the page shown in Figure 35. In this game, the player starts with a description and all multimedia associated with a specimen. They are asked to choose a series of experiments or observations to perform on that specimen. After selecting a set of experiments and observations, they identify the specimen. Students get points for identifying the specimen correctly and for performing the tests that were required to identify the specimen as specified by their own classification tree. They lose points for using too many experiments or incorrectly identifying the specimen. The specimen is on a particular path in the group's classification tree. The correct tests are the experiments on that path.

This view begins by sending a *GetGameLevel* event which returns information specific to the individual level. This includes the multimedia and specimen descriptions, experiments to perform, and all identities in the project. The *GetGameState* event returns a reference to the game, the game's type, the player's current score and whether the game is in progress or not.

Players can either click a Perform Experiment or a Choose: Specimen button. The Perform Experiment button reveals the result of performing that experiment by sending the server an *ActivateGameClue* event. The Choose Specimen button (Figure 36) sends a *SubmitGameAnswer* event and displays the correct answer, the chosen experiments on the classification tree path, and the total points received. It adds a Next Question button to the interface if there is another question or a Start a New Game button if there are no further questions. Clicking Next Question redraws the interface with the new level. Clicking Start a New Game returns the player to the game list.



Given the specimen description and images below, pick the experiments that you would use to correctly identify the object. Finally, identify the specimen! Click/hover over the ⓘ buttons for more information.

Score: 0

Question: 1

Description: Specimen 1-A

Leaf Venation - Current Result: Perform this experiment

Leaf Margin - Current Result: Perform this experiment

Leaf Arrangement - Current Result: Perform this experiment

Leaf Type - Current Result: Perform this experiment

Leaf Shape - Current Result: Perform this experiment

Fruit Type - Current Result: Perform this experiment

Leaf Persistence - Current Result: Perform this experiment

Figure 35. The top half of the select experiments game. The current game state, description and specimen multimedia is shown. Users select experiments corresponding to branches in their diagram.

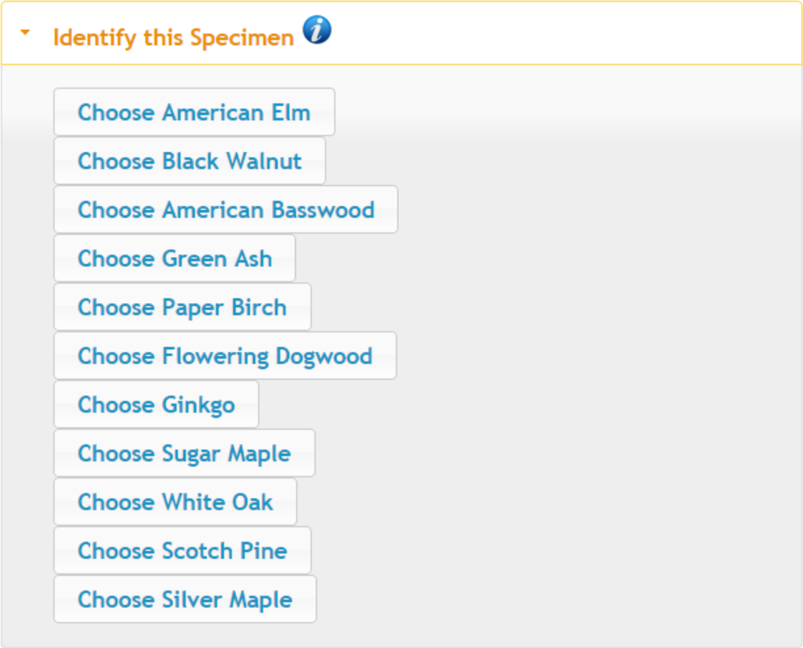


Figure 36. The bottom half of the select experiments game. After selecting the appropriate experiments, the player is asked to identify the specimen based on the results of the experiment.

Peer Help

In an effort to decrease the instructor effort, a peer help interface was created (Figure 37). In this area, students work with individuals enrolled in their CIRCLE course, but not in the same project. Peers can both suggest elaborations for other students to perform, and suggest and moderate hypotheses for specimens other groups have classified.

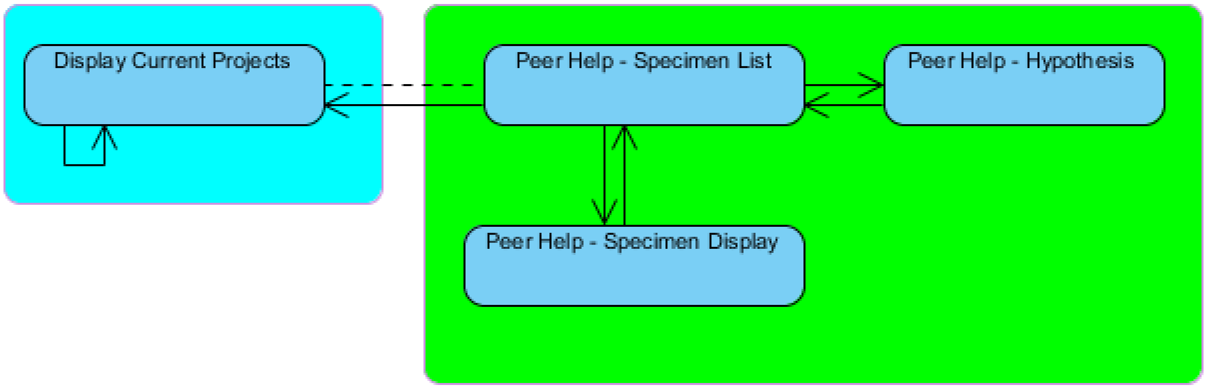
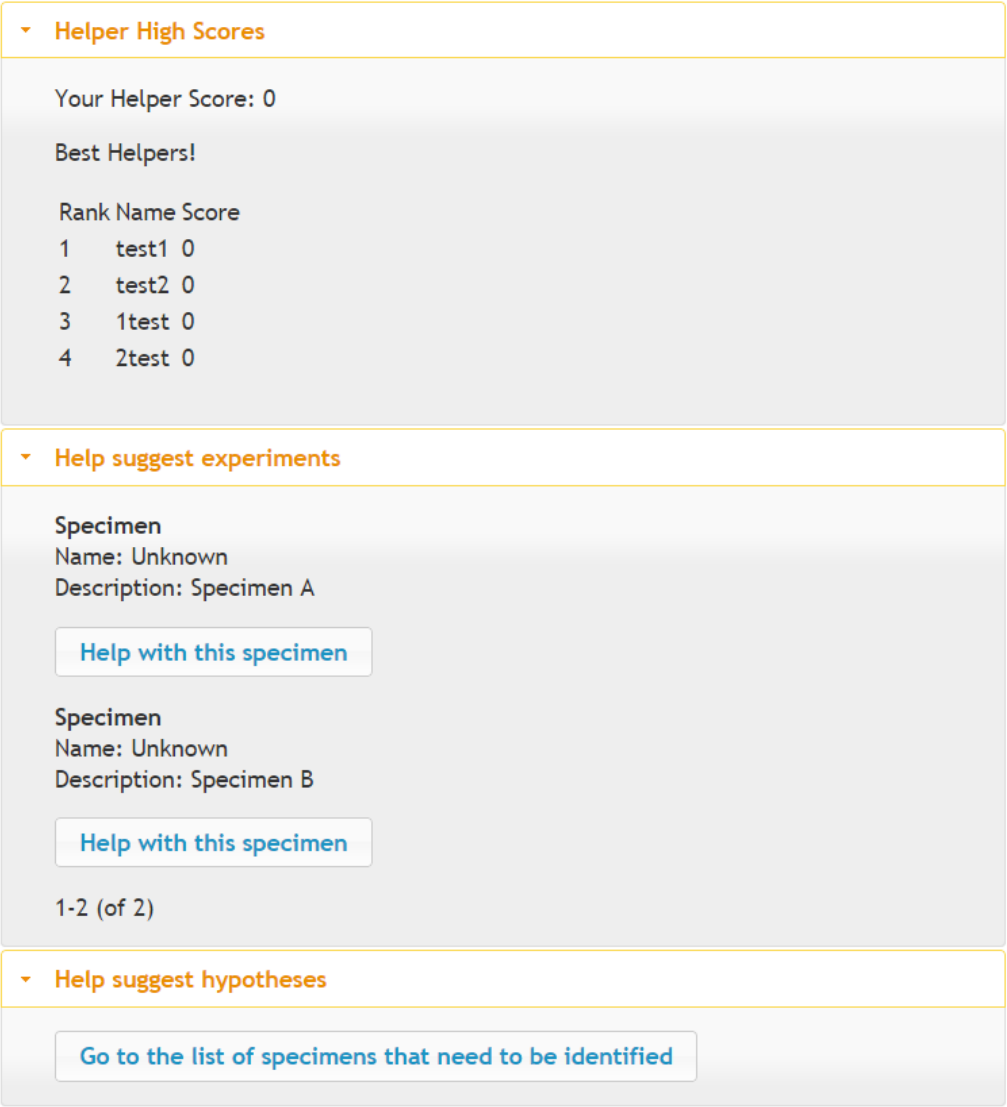


Figure 37. State diagram for peer help (green) with link to project setup and selection. Dashed lines indicate transitions that require a prerequisite to be fulfilled before the state transfer can occur.

Peer Help Specimen List

The peer help specimen list page (Figure 38) shows all specimens within this player's course. The primary purpose of this page is to show all of the specimens this player's classmates have gathered. They are allowed to click on the Help with this specimen button to go to the Peer Help Specimen Display view. The 'specimen' cookie is used to store a reference to the specimen of interest.



Helper High Scores

Your Helper Score: 0

Best Helpers!

Rank	Name	Score
1	test1	0
2	test2	0
3	1test	0
4	2test	0

Help suggest experiments

Specimen
Name: Unknown
Description: Specimen A

Help with this specimen

Specimen
Name: Unknown
Description: Specimen B

Help with this specimen

1-2 (of 2)

Help suggest hypotheses

Go to the list of specimens that need to be identified

Figure 38. Peer help specimen display. The peer help high scores are displayed at the top with the two available specimens to help with immediately below.

Players get peer help points for moderating experiments, suggesting hypotheses, and moderating hypotheses for other classmates' projects. A *GetPeerHelpHighScores* event is sent when the page loads to populate a table showing the players that have the highest score.

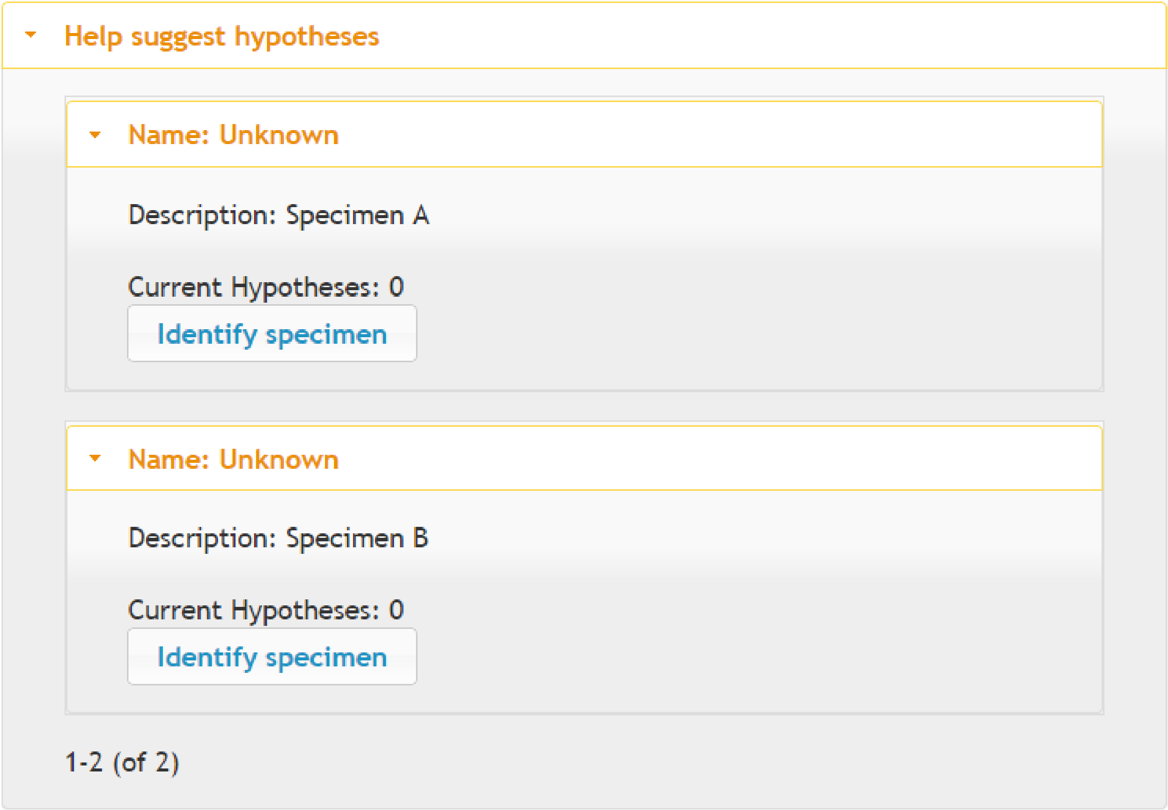
When the page loads, the project cache is used to load a subset of the specimens in the project. This is done for scalability reasons. There can be many specimens gathered within a particular course. The page currently loads five specimens at a time, with a Next button available. Whenever this button is pressed, another 5 specimens are loaded by the project cache. At the bottom of the page is a button to open the Peer Help Hypothesis List view.

Peer Help - Specimen Display

On the specimen display page, players help other groups with specimen elaboration. It is nearly identical to the Elaborate View, except that it is impossible for a peer to be the gatherer. Therefore, interface elements are limited to suggesting new experiments, moderating existing experiments, and writing bulletin board messages relating to the experiments. Clicking the Suggest Experiment button causes the client to open the Peer Help - Suggest Experiment page which is exactly the same as the Suggest Experiment view above, except that the navigation path is different. As in the Elaborate View, the project cache is used to load the specimen indicated by the 'specimen' cookie.

Peer Help - Hypothesis List

On the peer help hypothesis list page (Figure 39), players see all of the specimen groups in other projects. These specimen groups are limited by the server to be ones which are not error paths. As with the Peer Help Specimen Display, this list is limited to displaying five specimen groups at a time. Each specimen group has an Identify this specimen button which leads to a view that is identical to the Hypothesis View described earlier.



Help suggest hypotheses

Name: Unknown

Description: Specimen A

Current Hypotheses: 0

Name: Unknown

Description: Specimen B

Current Hypotheses: 0

1-2 (of 2)

Figure 39. The peer help hypothesis list page showing the two specimen groups that this user can help with.

Instructor Pages

Instructor pages are for faculty users (Figure 40). They allow faculty to create new courses, apply restrictions to those courses, and view statistics, including the history of player interactions with their classification trees.

Course List

This page displays all instructor-created courses and a button to create new courses (

Figure 41). The course list is loaded using the *GetCourses* event. Buttons associated with courses lead to the Course Options view page after storing the course reference in the ‘course’ cookie. The Create Course button opens the Create Course view.

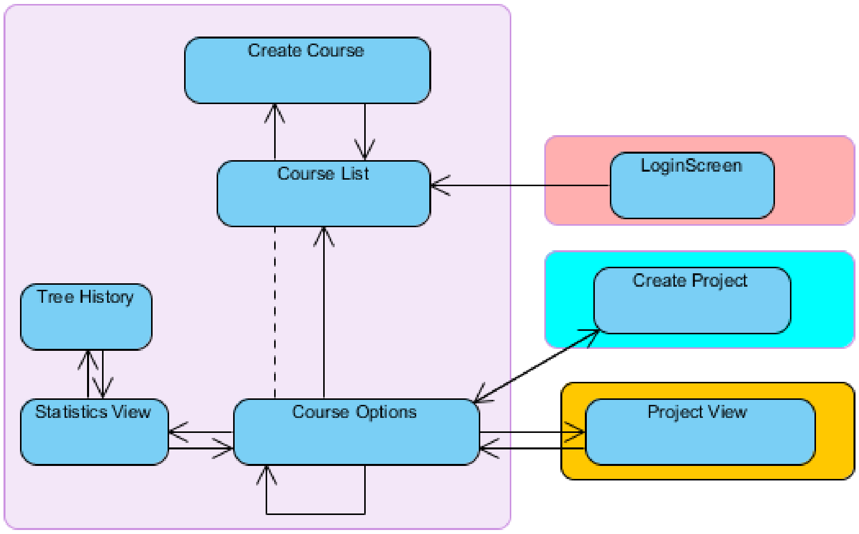
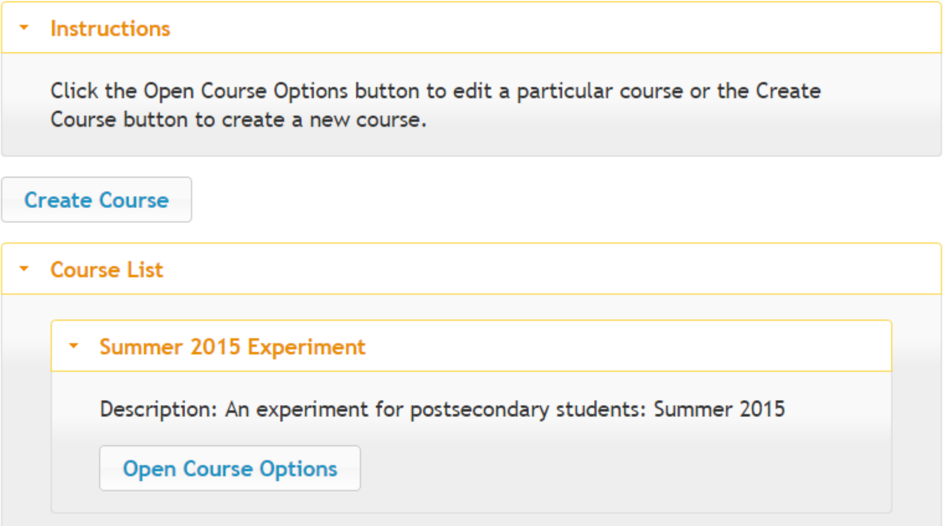


Figure 40. State diagram for instructor pages with links to the instructor log in (upper right), to create new projects (middle right) and to view their students' projects (lower right). Dashed lines indicate transitions with a prerequisite before the state transition can occur.

Course Options View

This page displays details of the course indicated by the 'course' cookie. These details include the course name and description. It also allows the instructor to restrict student behavior within their courses (Figure 42).



Instructions

Click the Open Course Options button to edit a particular course or the Create Course button to create a new course.

Create Course

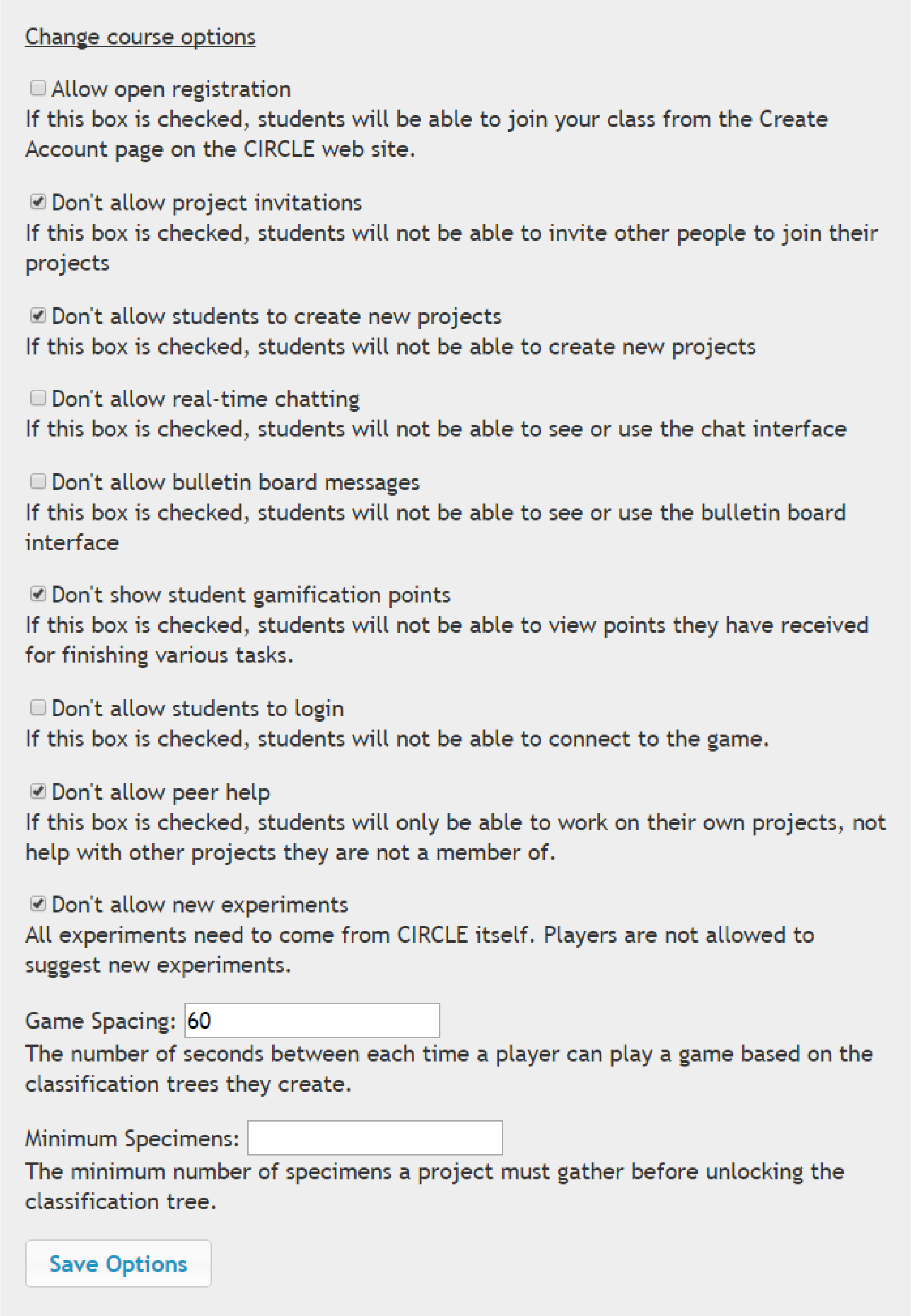
Course List

Summer 2015 Experiment

Description: An experiment for postsecondary students: Summer 2015

Open Course Options

Figure 41. The course list page: the first screen a faculty member sees when logging in.



Change course options

☐ Allow open registration
If this box is checked, students will be able to join your class from the Create Account page on the CIRCLE web site.

☑ Don't allow project invitations
If this box is checked, students will not be able to invite other people to join their projects

☑ Don't allow students to create new projects
If this box is checked, students will not be able to create new projects

☐ Don't allow real-time chatting
If this box is checked, students will not be able to see or use the chat interface

☐ Don't allow bulletin board messages
If this box is checked, students will not be able to see or use the bulletin board interface

☑ Don't show student gamification points
If this box is checked, students will not be able to view points they have received for finishing various tasks.

☐ Don't allow students to login
If this box is checked, students will not be able to connect to the game.

☑ Don't allow peer help
If this box is checked, students will only be able to work on their own projects, not help with other projects they are not a member of.

☑ Don't allow new experiments
All experiments need to come from CIRCLE itself. Players are not allowed to suggest new experiments.

Game Spacing: 60
The number of seconds between each time a player can play a game based on the classification trees they create.

Minimum Specimens:
The minimum number of specimens a project must gather before unlocking the classification tree.

Figure 42. The course options view showing modifiable options available for instructors.

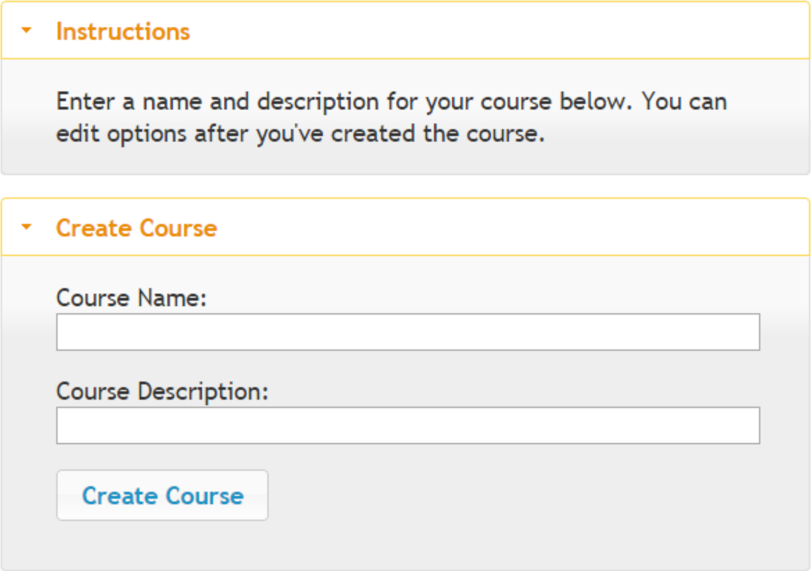
Restrictions allow instructors to decide what features of CIRCLE they would like to use in their course. For example, some high school teachers have expressed the desire to limit inter-student communication. The restrictions implemented in CIRCLE include the ability for students to: 1) register in the course, 2) invite other students to their projects, 3) create new projects, 4) chat (synchronously) with other users, 5) post bulletin board (asynchronous) messages to other users, 6) see gamification stats, 7) help individuals not in their project with CIRCLE (peer help), and 8) suggest new experiments for individuals in their project to perform. Faculty members are also allowed to set retrieval-based game spacing and the number of specimens that must be gathered before the classification tree is unlocked.

Options are set via the Save Options button, which sends a *ModifyCourseAttributes* event. Instructors can also create projects within this course and add students. When instructors invite students to projects, they are automatically added to the project, rather than going through the invitation process. Finally, instructors are allowed to access any project in their course and interact with students as if they were a part of the project, using the student interface. An extra "course" cookie is applied to the instructor before entering the project view screen so that the state change to return to the project list returns to the course options view.

Upon opening the course options view, the client sends a *GetCourseProjects* event to get information about the different projects in this course and a *GetCourses* event to get the initial restrictions and information on this course.

Create Course

On the create course page, the instructor enters a name and description (Figure 43). The course is formed after the Create Course button is clicked via the *CreateCourse* event. If the instructor hits cancel, the instructor is returned to the Course List view. New courses are automatically added to the available courses for student registration. Faculty must uncheck Allow Open Registrations on the course view page to stop random students from registering.



Enter a name and description for your course below. You can edit options after you've created the course.

Figure 43. The course creation view.

Statistics View

One of the primary contributions of this work is the ability for STEM researchers to analyze data to determine which behaviors lead to more expert-like classification trees and accurate identifications. While a number of other systems provide click level data (Asbell-Clarke, Rowe, & Sylvan 2013; Romero, Ventura, & Garcia 2008), CIRCLE includes specific information about the classification tree and the steps taken to create the classification tree. Data that is logged throughout CIRCLE is shown in Table 3.

The statistics view begins by sending a *GetCourseProjects* event to return all projects associated with the course in the 'course' cookie. For each project a *GetProjectStatistics* event is sent, which returns a JSON formatted string of the statistics listed in Table 3. These data points are organized into categories. Each category is associated with a button (Figure 44) that, when pressed, creates a chart (Figure 45) of the value for each of the projects in the course. This chart is created using the D3 framework discussed in the classification tree section.

Table 3. List of data points collected by CIRCLE.

Project Level Statistics
• Average number of experiments per specimen • Number of unidentified specimens • Number of identified specimens • Average number of multimedia per specimen • Number of chat messages • Number of bulletin board messages • Number of bulletin board messages read • Current tutor message for tree validation • Total degree of current tree • Average degree of experiment nodes in the classification tree • Number of Diagram Saves
Player Level Statistics
• Time on Task for each role (Viewing Project, Gathering, Elaborating, Classifying, Identifying, and Playing) • Number of times the user clicked the site map • Average time spent with an open tutor message (a corollary to time spent reading the message) • Total number of tutor messages closed

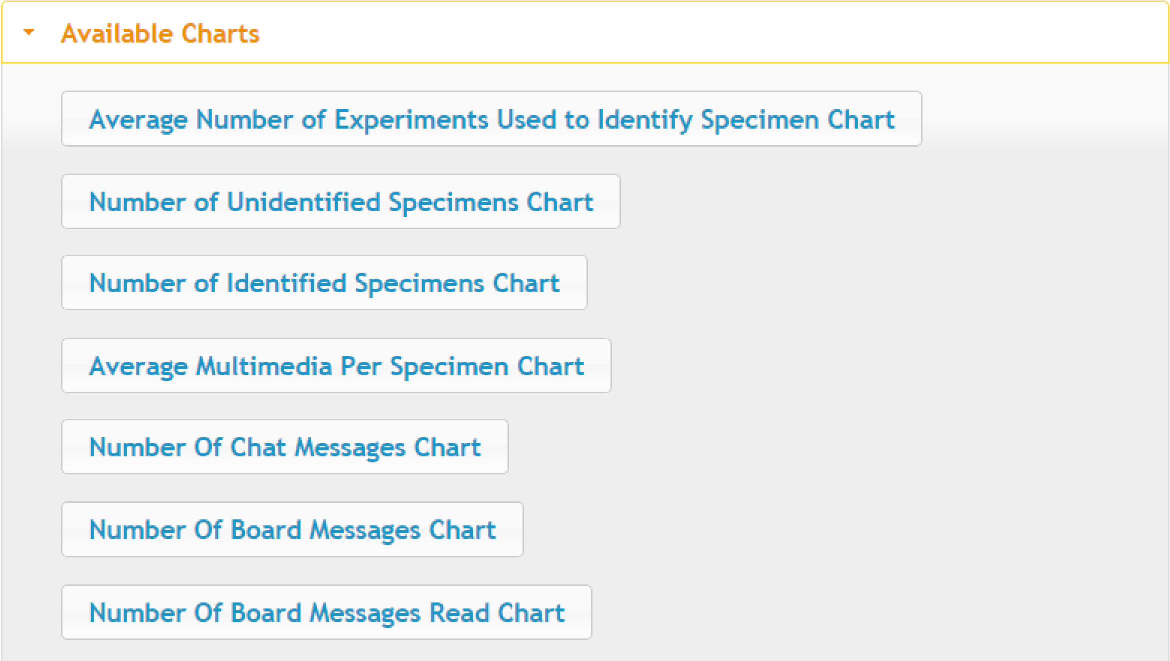

Available Charts
Average Number of Experiments Used to Identify Specimen Chart
Number of Unidentified Specimens Chart
Number of Identified Specimens Chart
Average Multimedia Per Specimen Chart
Number Of Chat Messages Chart
Number Of Board Messages Chart
Number Of Board Messages Read Chart

Figure 44. Top of statistics view page showing a subset of the charts that can be opened by the instructor.

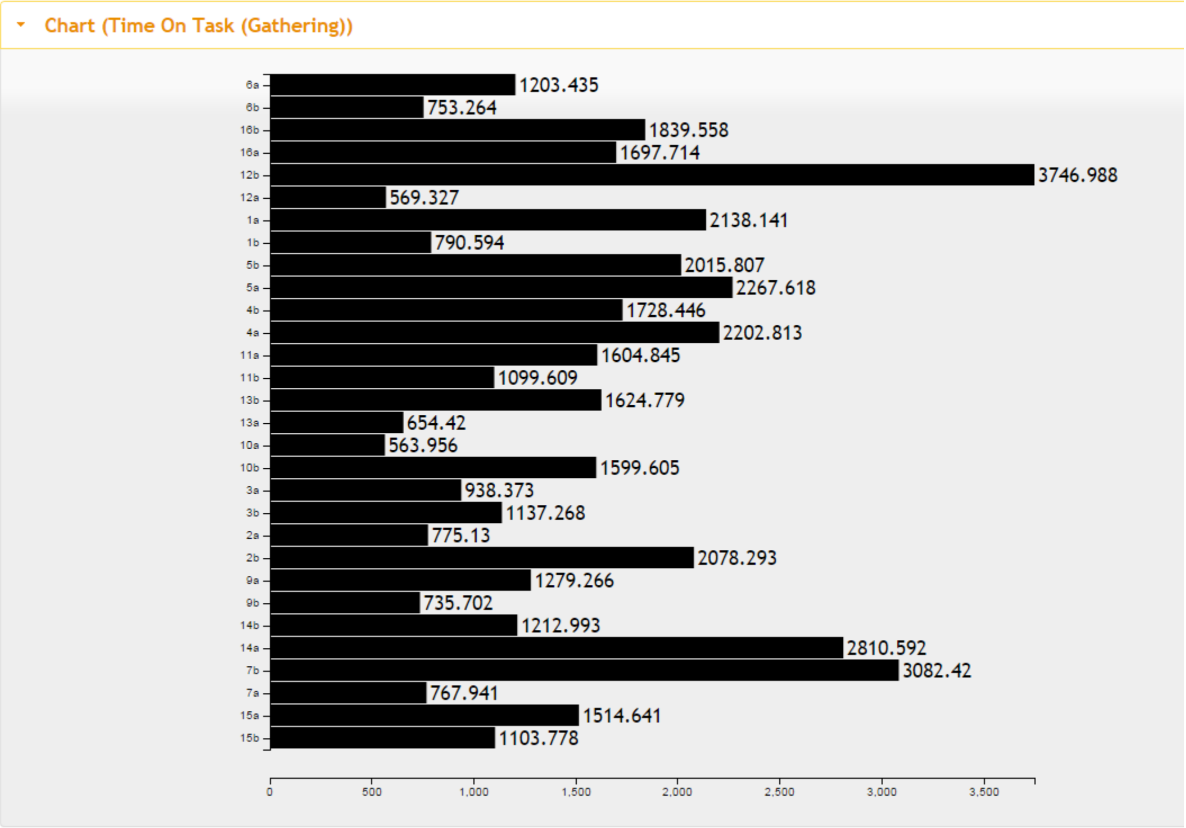
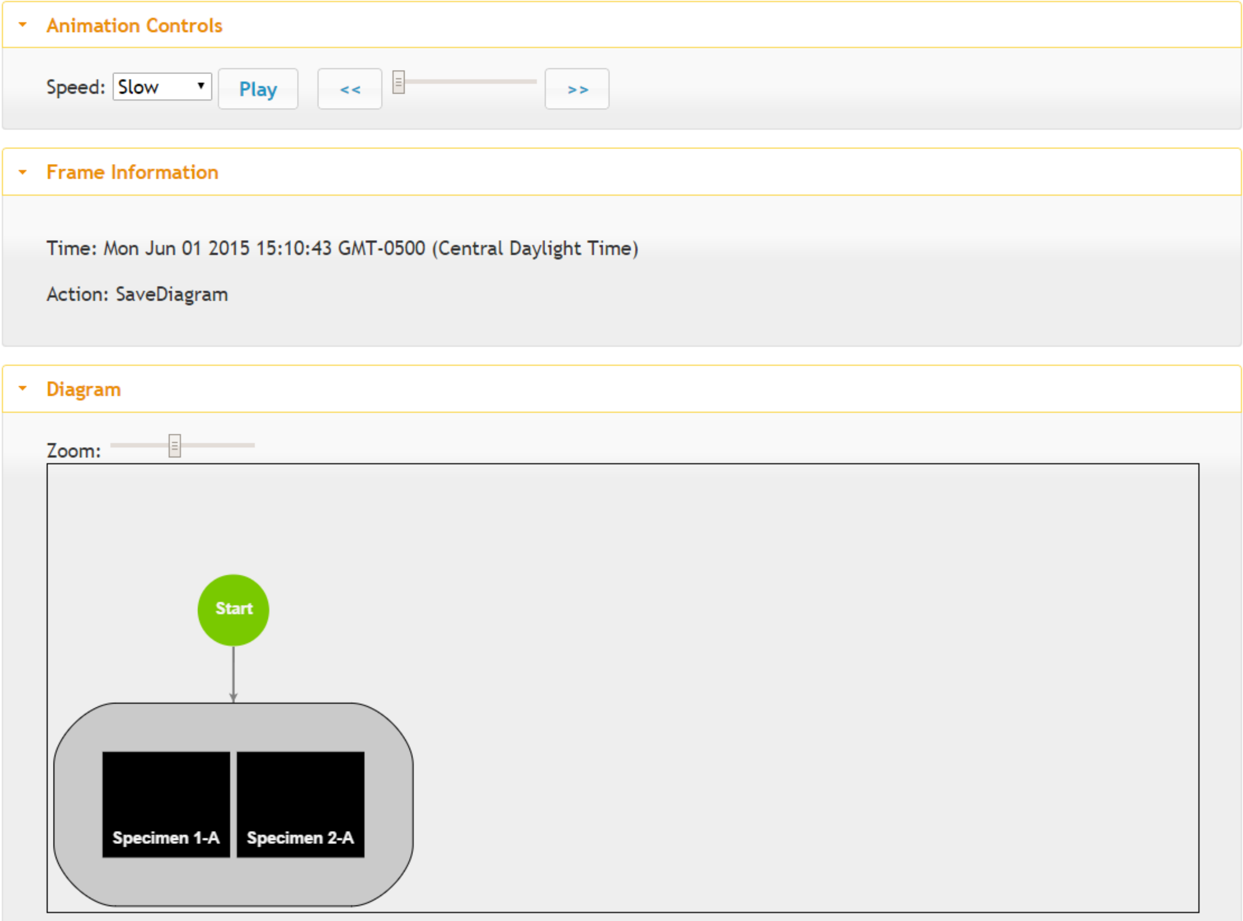


Figure 45. Chart of the time on task for the gathering task for all projects in a class.

Tree History

The system also includes a tree history interface. Each time *SaveDiagram* is called, a fully specified diagram is attached to a log entry. This diagram includes the experiments, links, and what would have appeared if the player had clicked the “Animate All Specimens” button and allowed the animation to finish. Instructors and researchers can press the play button and watch as the student builds their classification tree (Figure 46).

A ‘project’ cookie is set before entering the tree history page. The page sends a *GetTreeHistory* event for that project, which returns a series of JSON messages corresponding to each *SaveDiagram* event the players sent during the project. These diagrams are placed in a queue and viewed using the animation controls. Animation controls include a play/pause button, and buttons to advance a single frame backwards or forward. A drop down box changes the tree history animation speed.



Speed: Slow

Time: Mon Jun 01 2015 15:10:43 GMT-0500 (Central Daylight Time)

Action: SaveDiagram

Figure 46. The tree history view. This is the first saved frame. The user has added the start button, but no experiments are included yet.

Server Implementation

CIRCLE uses JavaMOO as its primary server technology. JavaMOO (Kariluoma et al. 2013; Vender et al. 2010) was developed by North Dakota State University and WoWiWe Instruction Co LLC for use in the creation of multiplayer educational games. It is founded on REST (Representational State Transfer) principles (Fielding 2000) which allows multiple clients to be created as long as a suitable client-to-server messaging protocol is developed.

JavaMOO was originally developed to connect to standalone clients. For example, the WoWiWe Virtual Cell client was written in Unity and connects to a socket and sends events in JSON format. This work extends JavaMOO with the ability to receive AJAX (Asynchronous Javascript and XML) style events from web browser based clients.

Game State

The server stores game state information as domain objects. All domain objects created for CIRCLE are children of MOObject, the base object in JavaMOO. These domain objects store information about players, projects, specimens, experiments, experimental results, classification trees, hypotheses, games, and other related components of CIRCLE. A full list of domain objects is shown in Table 4.

Players

The CirclePlayer class stores information specific to an individual player. This information includes the player's projects, current project, current task (Gathering, Elaborating, Classifying, Identifying, or Playing), current invitations to other projects, ignored tutor messages, server-side tutor messages, enrolled course, and restrictions.

As implied by the name, the Demographics class stores demographic information about the player. Information stored includes: age, gender, education level, ethnicity, race, income, address, city, state, postal code, grade point average, predicted course grade, current grade level, and whether or not they are a part of a research study.

The CircleAccount class provides a container for all information about the player's account. This includes the player's object, their demographic information, and the player's log information.

Table 4. Domain objects in CIRCLE.

Players • CirclePlayer • CircleAccount • Demographics	**Classification Tree** • ClientDiagram
Logging • CircleLog • BasicCircleLogElement • ClientTrackingLogElement • PeerHelpLogElement • ServerTutorLogElement • ProjectBasedLogElement • AchievementLogElement • DiagramLogElement	**Specimen Groups and Hypotheses** • SpecimenGroup • Hypothesis • HypothesisData
Projects • Project	**Game Templates** • Scoreboard • ScoreEntry • Game • SelectExperimentsGame • GameInfo
Specimens • Specimen • MultimediaContent	**Asynchronous Communication** • Board • Note
Experiments / Results • Experiment • ExperimentData • AudioExperimentData • ImageExperimentData • VideoExperimentData • TextExperimentData • NumberExperimentData • SuggestedExperimentData	**Course Information** • CircleCourse • FacultyAccount • ClientActionRestrictions

Logging

One of the most important features of CIRCLE is logging. A full class diagram is shown in Figure 47. Whenever an important action is performed by the player, it is stored on the server as a specific LogElement using the CircleLog class. The parent class of all CIRCLE LogElements is the BasicCircleLogElement.

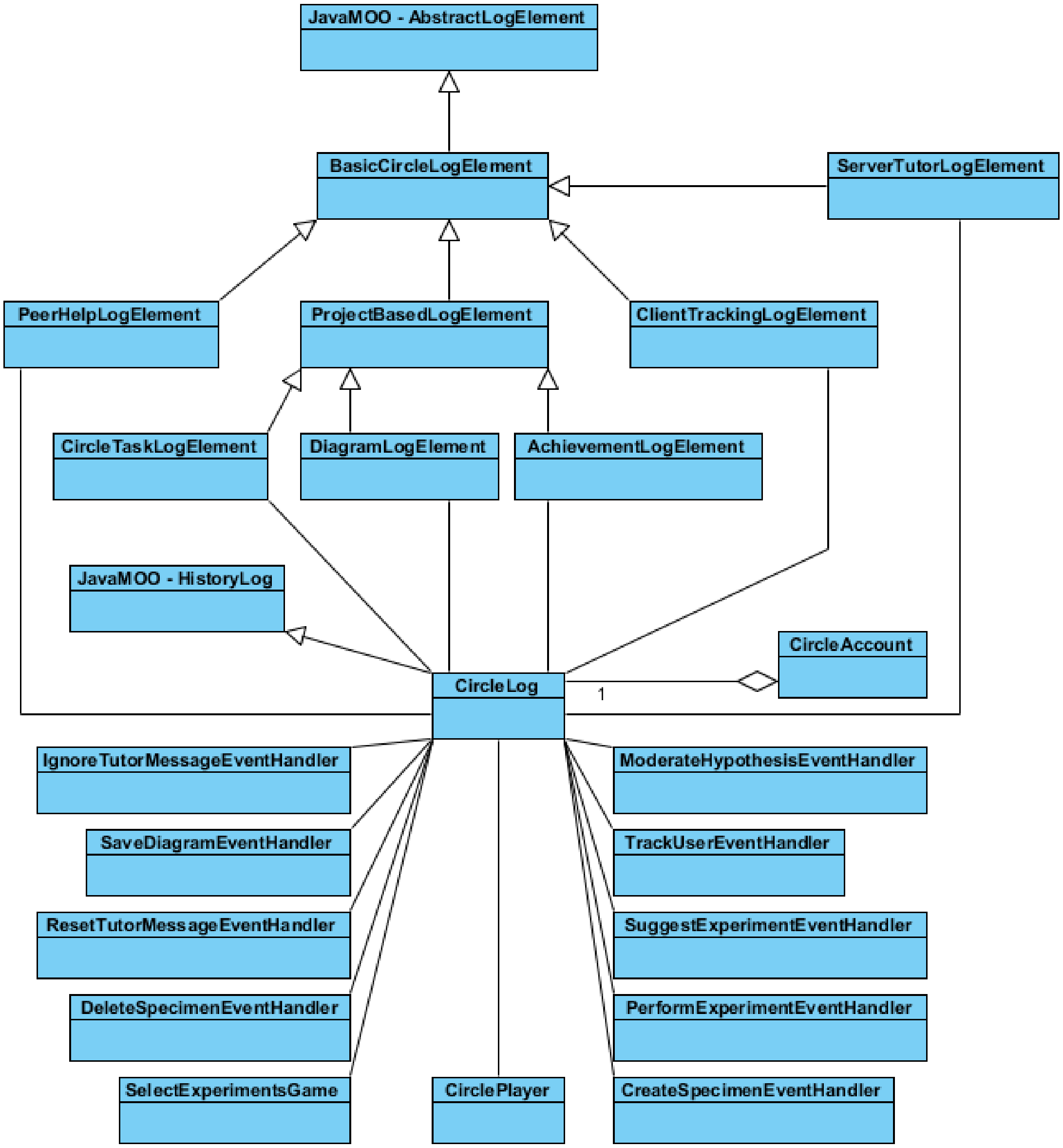


Figure 47. Class diagram for logging interactions.

Children of BasicCircleLogElement include ClientTrackingLogElement, PeerHelpLogElement, ServerTutorLogElement, and ProjectBasedLogElement. The server creates a ClientTrackingLogElement when the player performs a *TrackUser* action on the client. This includes button presses, link clicks, opening and closing tutor dialogs, and entering and leaving web pages. The server creates a PeerHelpLogElement when a player moderates a hypothesis or suggests an experiment to perform in someone else's project. The server stores a ServerTutorLogElement when a player ignores or resets a tutor message.

The ProjectBasedLogElement is the parent of all logs that are related to a particular project, including AchievementLogElement and DiagramLogElement. The AchievementLogElement stores the formative gamification data discussed earlier. The DiagramLogElement stores a copy of the project's classification tree for later replay.

Project

A CIRCLE project (Figure 48) organizes a group of individuals towards the classification and identification of a particular type of real world object. Project objects store information about an individual project. Players can be involved in multiple projects, each corresponding to a different set of objects. The typical classroom use of CIRCLE involves students joining a single project that corresponds to an assignment. The Project class stores information linking all of the data collected by a project together.

Group membership is handled through the project class. It keeps track of the all current players, all logged in and participating players, and invited players.

The project also stores all data about the group's progress. This includes the list of all specimens in the project, all hypotheses suggested on any specimen group, all experiments suggested on any specimen in the project, the current classification tree, and the current summative gamification games being played by project members. It also keeps track of which summative gamification games are available for group members to play, the bulletin board associated with the classification tree, the keywords associated with the project, and a link to the project's course.

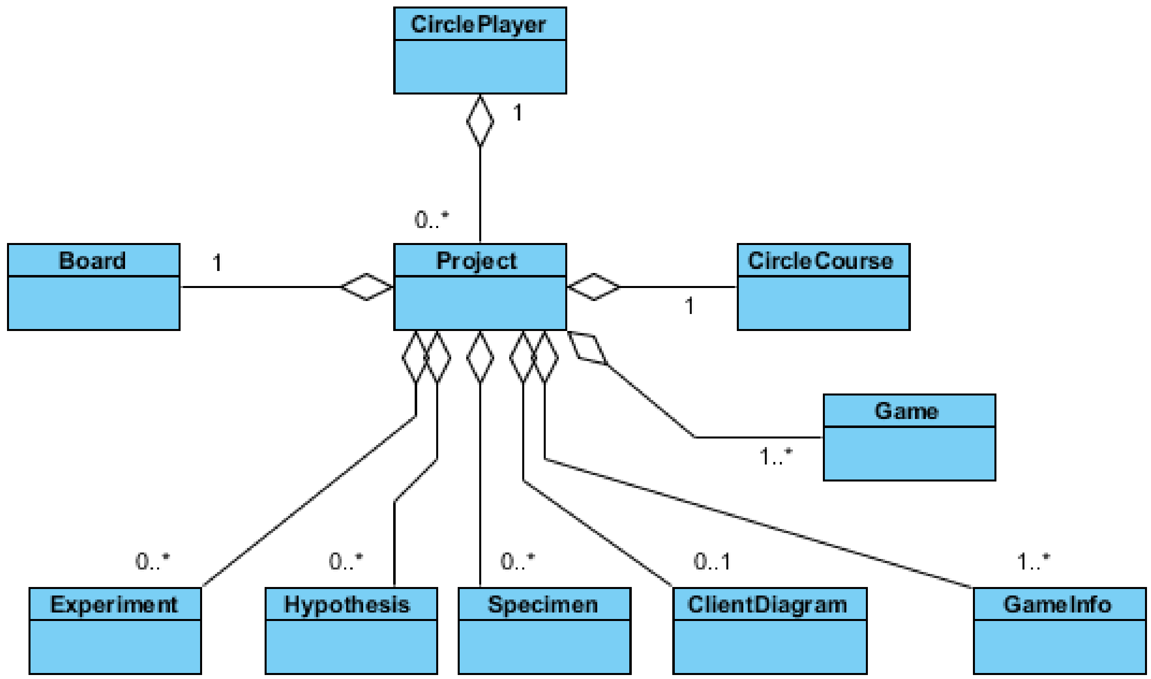


Figure 48. Class diagram for projects.

Specimen

A CIRCLE specimen is an individual object that the group would like to classify and identify. The class stores information associated with the specimen including who gathered it, when it was gathered, the gathered location, included multimedia, and the experiments that have been suggested for the specimen via the SuggestedExperimentData class.

The MultimediaContent class stores information about multimedia gathered by the group. This includes who created the content, the content's type (which currently can include image, video, or audio), and the current Uniform Resource Identifier of the content.

Experiments

In order to distinguish between specimens, players need to suggest and perform experiments. There are two main classes used for experiments, the Experiment class and the

ExperimentData class (Figure 49). These two classes are linked to specimens using the SuggestedExperimentData class.

The Experiment class defines the project-level object for experiments. This class stores common information across an experiment, regardless of its specimen. This includes instructions for performing the experiment, the description, the content types required for the results in order to satisfy the experiment, a list of the results that have been collected for this experiment, and a bulletin board to discuss this experiment across the project.

The ExperimentData class stores the results of performing an experiment. The base class contains abstract methods for getting the result, getting the type of the result, and parsing incoming data for storage in the object. The subclasses: TextExperimentData, NumberExperimentData, VideoExperimentData, AudioExperimentData, and ImageExperimentData provide specific implementations for storing the results of an experiment.

The SuggestedExperimentData class is the specimen-level object for experiments. While a project can have many different experiments, only a certain subset of the experiments needs to be suggested for an individual specimen. This class handles that distinction. It stores a link to the project level experiment, the specific results of performing this experiment on the associated specimen (as a list of ExperimentData objects), the moderation scores of the experiment as it relates to this specimen (that is, how important the group feels this experiment will be towards classification), and who suggested the experiment for this specimen.

The distinction between a project-level and specimen-level experiment ensures that an Experiment can be used in the classification tree with all of the ExperimentData classes as results. SuggestedExperimentData stores information specific to an individual specimen and its results.

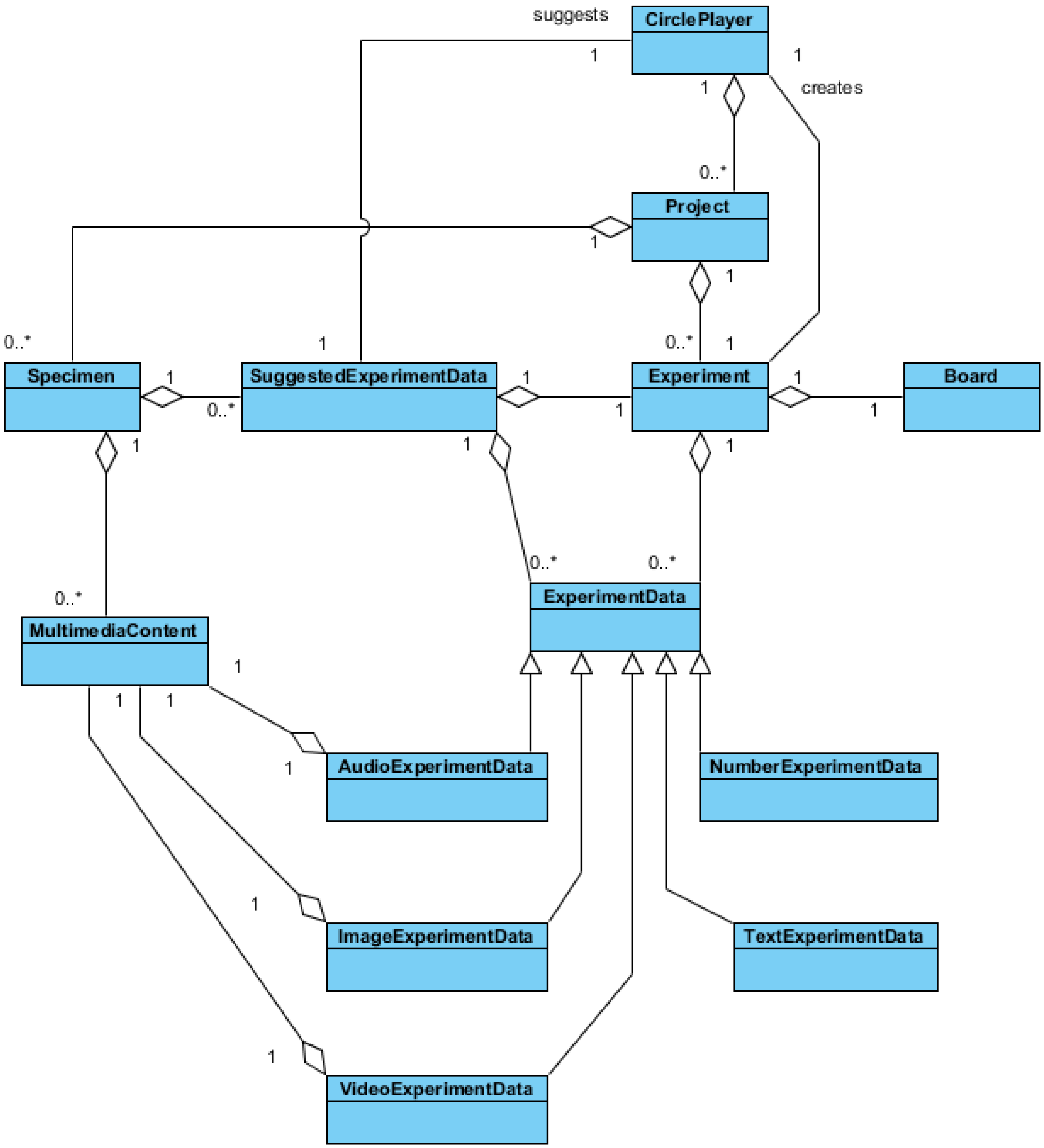


Figure 49. Class diagram for experiments and experimental data.

Classification Tree

The ClientDiagram class contains a copy of the classification structure stored as a JSON object (in Java, this is a HashMap<String, Object>). The class also maintains the token indicating which player is allowed to edit the diagram. It also contains a series of helper methods used by specimen groups and summative gamification templates to access the information in the diagram. Finally, it maintains a list of all paths within the client-side diagram. These paths correspond to specimen groups that need to be identified.

Specimen Groups and Hypotheses

A specimen group is a set of specimens that shares the same experimental results as described by an individual path through the classification tree (Figure 50). The SpecimenGroup class contains all of the hypotheses suggested for the group, a list of the experiments and results indicated by the classification tree to identify the group, a list of the specimens within the group, an integer storing the GoJS key to the specimen container within the diagram that matches this specimen group, and a boolean indicating if this particular group is an error group, that is, if it is a group immediately after the start node or if the experiment immediately prior to the group has not been performed on the specimens in the group.

The Hypothesis class is the project-level description of a hypothesis. It contains information about hypotheses that are common across specimens. This information includes the name and description of the hypothesis. The distinction between project-level and specimen level hypotheses is needed so that multiple specimen groups can have the same hypothesis. Rather than having a player type out a new specimen name every time, the client can return a list of all hypotheses that have been offered so far.

The HypothesisData class defines the information specific to an individual specimen group. This includes the reason this hypothesis was suggested for this particular specimen group, a bulletin board for discussing the hypothesis as it relates to the specimen, and the current votes the group has given towards a positive identification of the specimen group.

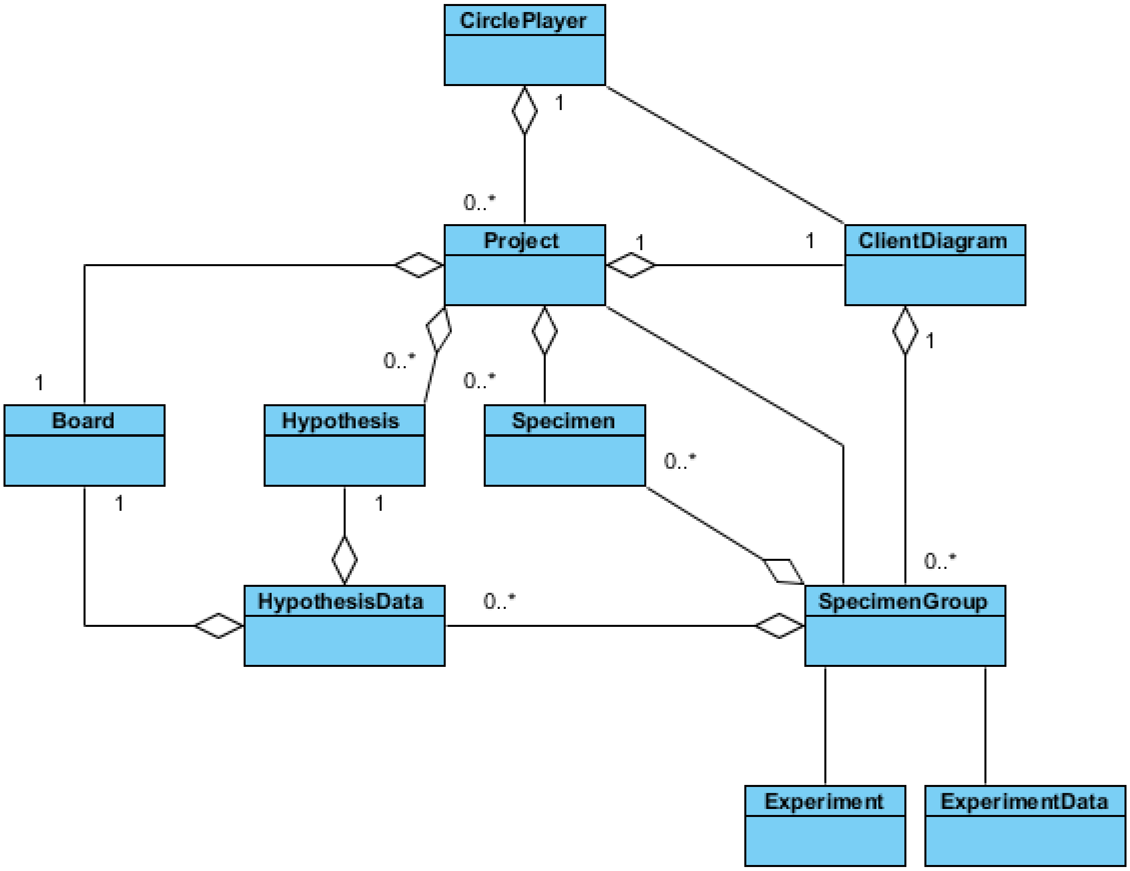


Figure 50. Class diagram for specimens, diagrams, specimen groups, and hypotheses.

Game Templates

The server implements summative gamification through the use of game templates (Figure 51). These templates provide a framework for creating a variety of different games.

The GameInfo class stores project-level information about game templates. This includes the high score list, and when the games were last played.

Game is the base class for creating new summative gamification game templates. There are abstract methods for starting a game, getting the current level and a description of the level, getting the number of points in the current game, getting a hint or clue for continuing in the game, getting the type of game, and processing an answer sent by the client. Currently, the only implementing class of Game is the SelectExperimentsGame. It uses the client diagram and its helper functions to create an individualized game for the players based on their gathered specimens and classification structure.

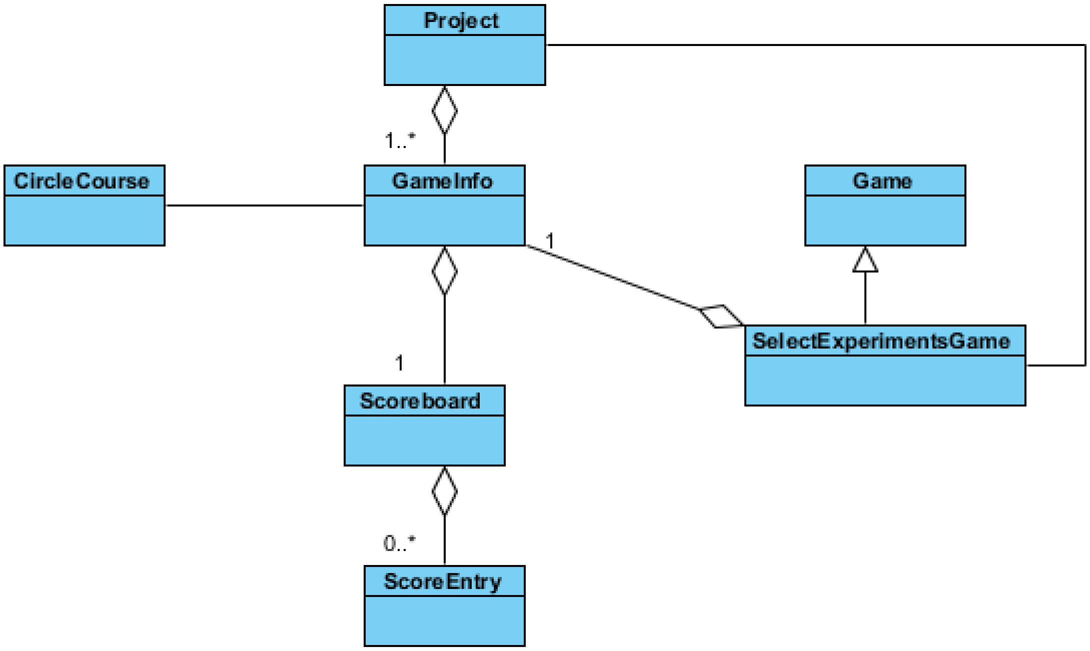


Figure 51. Class diagram for game templates.

To create new game templates, a programmer must extend the Game class and implement the Start, GetClue, GetLevel, GetLevelNumber, GetPoints, ProcessAnswer, and GetGameType methods. The sample implementation in SelectExperimentsGame can provide a useful starting point.

Synchronous and Asynchronous Communication

The server defines asynchronous communication via two classes: Board and Note (Figure 52). These classes are derived from prior research (Borchert 2008). Board contains a list of Notes and helper methods for retrieving information about stored messages. Note maintains information specific to an individual note, such as the note's parent within the conversation thread, subject, content, author, and who has read the note.

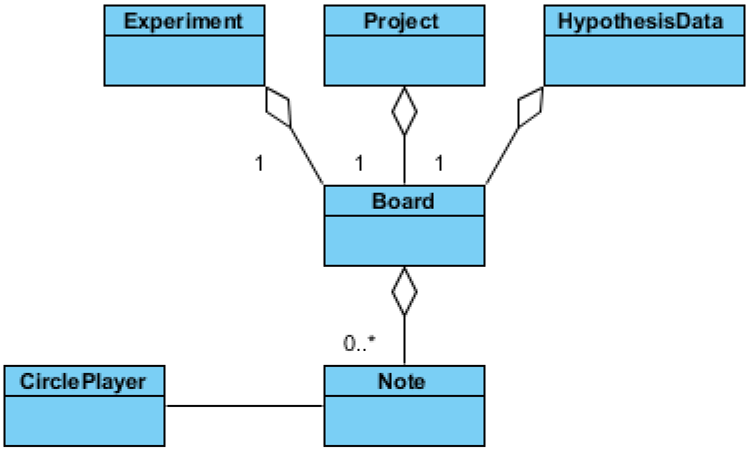


Figure 52. Class diagram for asynchronous communication.

Synchronous communication is handled by the default JavaMOO installation. ChatLogElements are stored on the server via SendChat events and responses. These elements are retrieved using GetChat.

Courses

When creating their account, players register for an individual course. If the player does not choose a particular instructor-created course, they are assigned a default course. This course is created when CIRCLE is installed for the first time. Courses are represented via the CircleCourse class (Figure 53). The class contains the specifications for individual projects within the course. These include the number of specimens required to be gathered and experimented on before classification is unlocked, the set of restrictions that clients must obey (via the Restrictions class), the minimum amount of time between formative gamification sessions, whether this course is open to registration, and a link to the FacultyAccount that represents the instructor of the course.

The FacultyAccount class is a child of CircleAccount that adds the list of courses for which that faculty is an instructor. This class provides higher privileges to a number of different events, such as ChangePassword (faculty members can change their own password, and those of their students), InviteUsers (faculty don’t invite users, they directly add users to projects in their course), and GetProjects (rather than getting their own project, it returns a list of all projects in their course). Faculty also receive multiple votes when moderating hypotheses, since it is assumed that the faculty member is correct when they are voting on a hypothesis.

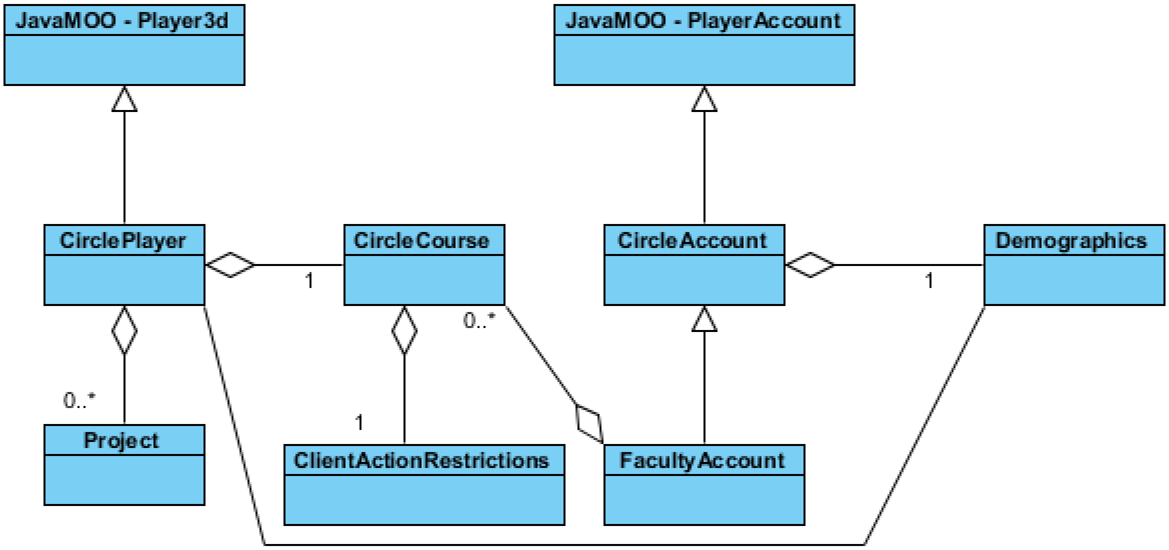


Figure 53. Class diagram for players, accounts, and courses.

Individual instructors are allowed to limit the actions of their students through the use of ClientActionRestrictions. These restrictions include the ability to stop players from inviting other players, creating new projects, chatting synchronously, creating bulletin board messages, logging in, offering peer help, and suggesting experiments. If the suggesting experiment option is selected, the instructor must provide a list of experiments that will be suggested for all new specimens. This is currently done through the *BuildExperimentGroups* event.

Event and Response Handling

There are three main types of classes associated with event handling in CIRCLE (Figure 54). Each event has its own event class, a child of MOOEvent, which stores the data retrieved from the incoming client message. Each of these events has an event handler, a child of MOOEventHandler, which performs the actual action required by the event. These classes contain a handle method that must return an object of type MOOEventResult. This result describes the response that is returned to the client.

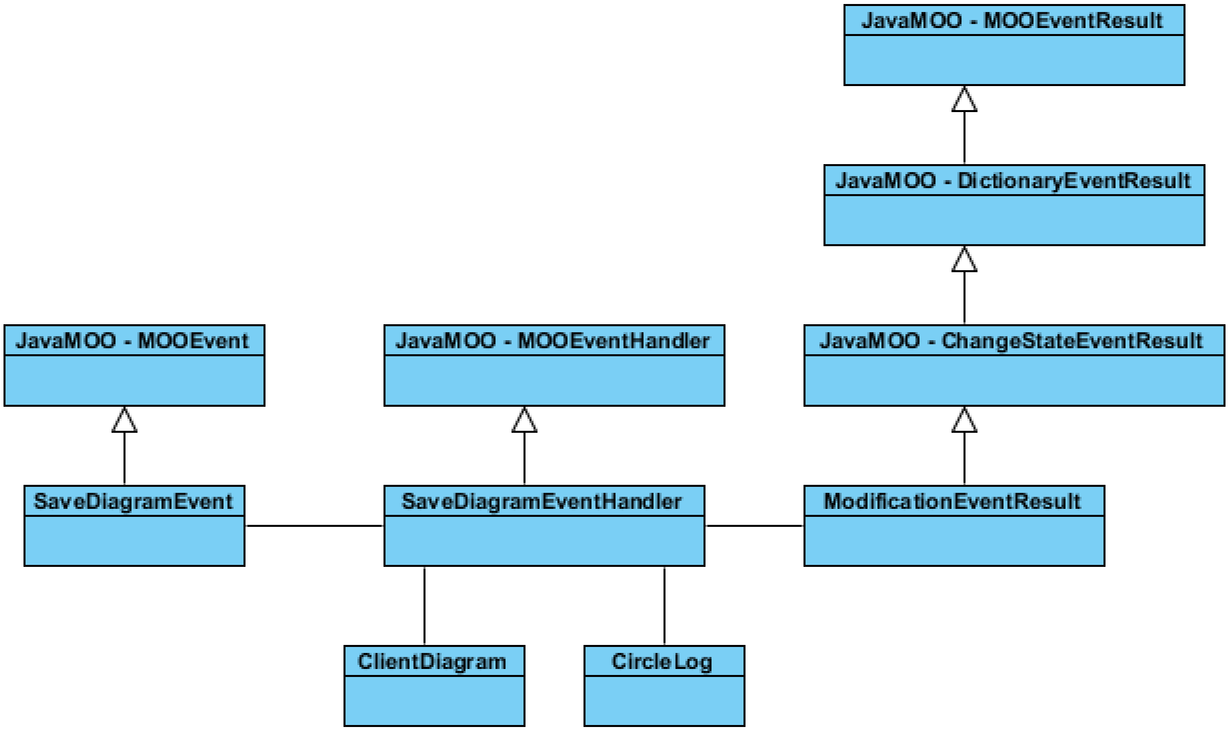


Figure 54. Class diagram for the SaveDiagram event, result, and event handler (the others are similar).

New events are registered in CIRCLE by adding the appropriate XML lines to the javamoo-events.xml file. There is one line for associating a particular event (ex: GetProject) with a particular class (circle.event.GetProject) and an entry for associating an event (circle.event.GetProject) with a particular event handler (circle.event.handler.GetProjectEventHandler). The Spring Framework (http://projects.spring.io/spring-framework/) is used to parse these XML files and create associations between the events and the event handlers.

PAPER: HERE A GAME, THERE A GAME: CLASSIFICATION, GAMIFICATION, AND RETRIEVAL LEARNING[2]

A paper submitted and accepted at the First International Conference on Higher Education Advances. Authored by Otto Borchert, Guy Hokanson, Alekya Peruri, and Brian M. Slator.

Abstract

A multi-user classification system is used in a classroom setting. Groups of students take samples, then collaborate on conducting experiments and building classification trees. Students are motivated by a point scoring system as they build the trees: formative gamification. Once the trees are completed they are exchanged between groups. The trees are constructed with structural commonality enabling them to act as content for template-based identification games: summative gamification.

Introduction

CIRCLE (Collaborative Identification, Retrieval, and Classification Learning Environment) uses mobile technology to support teams of students who gather content for classroom projects that identify and classify real world objects such as plants and animals or rocks and minerals. The classification trees produced are used to populate identification and classification game templates that students can then play to rehearse through practiced retrieval. Retrieval learning postulates the act of retrieving knowledge in multiple different ways leads to better learning outcomes (Karpicke 2012).

[2] The material in this chapter was co-authored by Otto Borchert, Guy Hokanson, Alekya Peruri, and Brian M. Slator. Otto Borchert developed the methods and conclusions, identified and applied software fixes, and authored the paper. Guy Hokanson was a beta tester and proofreader. Alekya Peruri linked SUS questions with specific suggestions and fixes. Brian Slator was a proofreader and invented the terms summative and formative gamification.

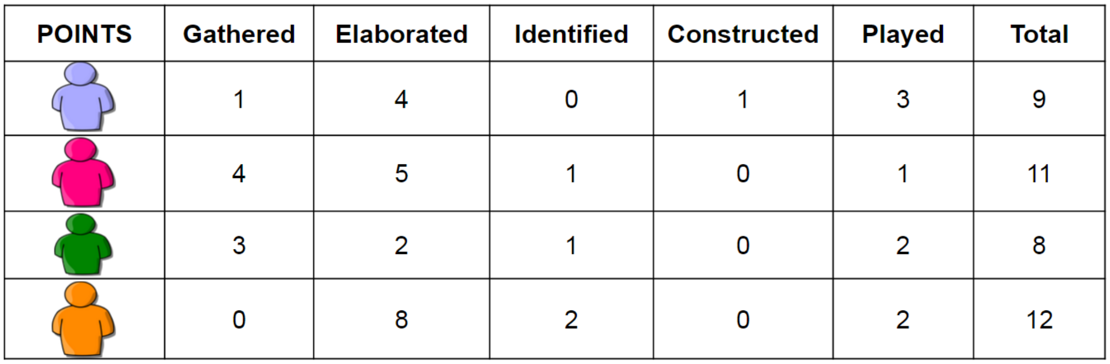
In decades past, identification and classification learning took place through the use of flash cards. These paper-based practices evolved into identification and classification games during the Internet age (Kids Know It Network 2013; Crimson Trails 2013; Jensen 2013; Kinder Web Games 2013). There are two primary drawbacks to these methods. While they have been studied extensively (Nist & Joseph 2008; Kornell 2009), they are not informed by modern educational practice. It has been shown that students are not simply empty vessels, meant to be filled, but actively construct their knowledge (Richardson 2003; Bransford, Brown & Cocking 2000). In addition, these games are fairly static and designed completely by their developers.

Game templates offer an avenue for improvement on these static games. An example of a game with a template is Geoguessr (Coldwell 2013). In this game, students see a Google Street View image and get points for identifying where on the world they are located. Less technological game template solutions include trivia games and typing tutors.

While these pre-defined templates are entertaining, user generated content provides a unique method of motivating individuals. Studies identifying motivations of individuals creating user generated content show a variety of reasons for its development (Daugherty et al. 2008; Leung 2009). Rather than using content developed by instructors, we place content gathering into user hands. We harness this motivation to teach students the identification and classification task.

Methods

CIRCLE implements retrieval learning through the use of games in two primary ways. First, students gather content, identify specimens, and build classification structures. Students score points in this phase by contributing to the content structures. We call this formative gamification. Second, game templates are created by developers. These templates specify the rules for a particular game, while remaining context independent. Students reinforce their learning by playing these games which are generated by integrating specific content into the game template. We call this summative gamification.



POINTS	Gathered	Elaborated	Identified	Constructed	Played	Total
👤	1	4	0	1	3	9
👤	4	5	1	0	1	11
👤	3	2	1	0	2	8
👤	0	8	2	0	2	12

Figure 55. Students score points which provide motivation and individual accountability. The player in the fourth row has not gathered any specimens, but has 12 points, the most on the team.

Implementing Classification Acquisition

There are five primary student activities in CIRCLE. Content acquisition, trait elaboration, hypothesis formation, and tree construction are independent activities while game play is part of all the others. The system tracks students and records their use of the system, time on task during each activity, and interactions with each other. Students are assigned collaborative roles according to the activities. Throughout each of these stages, points associated with these activities are kept as a running score, see Figure 55. We call this formative gamification.

Content Acquisition

Gatherers primarily collect photographs that are uploaded to a central server for inspection by the team. Camera phones with GPS are preferred, since this simplifies logging the time and place the specimens are collected.

Trait Elaboration

Elaborators suggest traits to observe or experiments to conduct as a means towards identification. Gatherers perform these experiments and observations, returning to the logged location if necessary. The team can also vote on experiments they feel are most important to completing the identification.

Tree Construction

Classification trees are constructed where traits and experiments are shown as branches of the tree, and specimens are the leaves. Students in the role of ‘constructors’ take suggestions from team members on how to arrange the structure leading to a dichotomous key.

Hypothesis Formation

Hypotheses are offered by team members in the ‘identifier’ role as to the general classification or precise identity of the specimens. In the absence of an expert, such as a classroom teacher, hypotheses are voted on in a parimutuel-like system, where a probability of identification is associated with each competing classification.

Game Play

A knowledge base of logging data is collected during the four phases, see table 1. This data is used to populate game templates, described below. Players select a game to play and their collected content is used to create a game unique to their information, including all experiments and observations suggested and performed, all results of these experiments and observations, the moderation scores of the elaborations, all hypotheses suggested whether correct or not, the identification of each individual specimen, and the specific set of tests required to identify the specimens.

Example

After a gatherer has submitted photographs of a rocky outcrop, an elaborator might suggest a 'hardness' experiment using a 'glass plate' to classify a mineral. Team members moderate the experiments and voting takes place to reach team consensus on the important trait elaborations, such as an acid test be performed on a neighboring rock. The constructors take these pieces and interactively shape them into a classification tree. Then student identifiers suggest "Limestone" for a particular node of the tree as a hypothesis based on the experimental results. The combination of these efforts represents the learning outcomes of the team

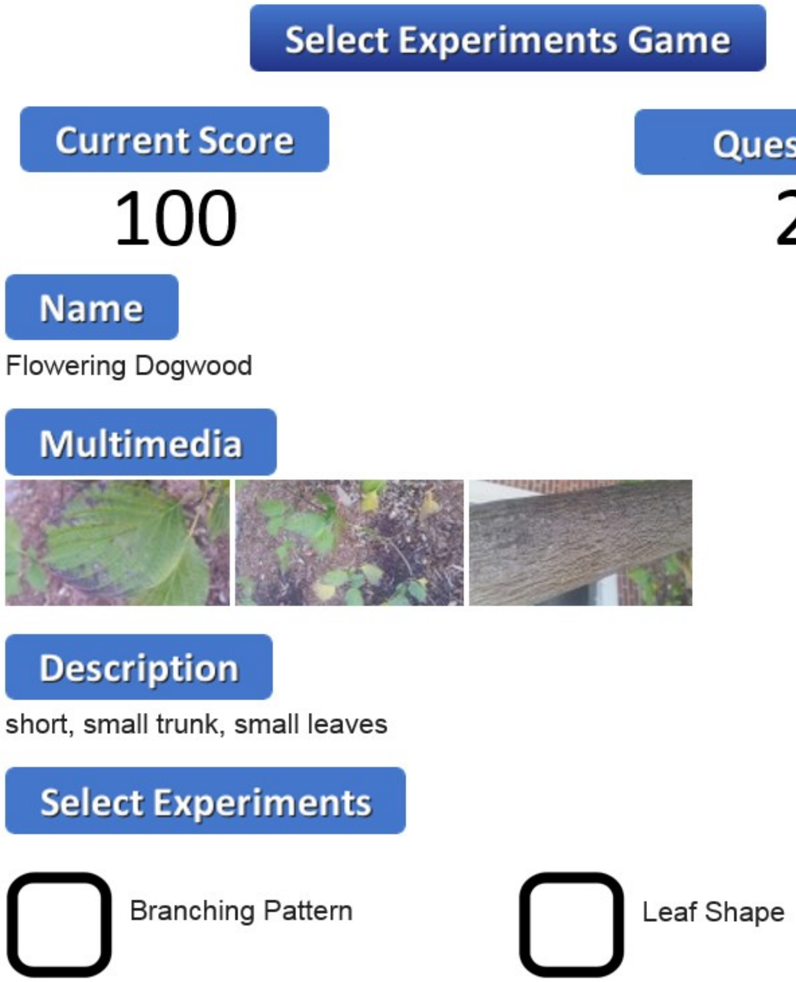
Table 5. Data gathered in CIRCLE by stage.

Stage	Data Gathered
Gathering	• Text description of the specimen • Multimedia artifacts of the specimen (ie audio, video, and images) • Text description of the location of the specimen • GPS coordinates of the specimen (if available)
Elaborating	• Name of the elaboration • Instructions for how to perform the elaboration • Basic description of the elaboration. • Moderation score for this elaboration • Results of the elaboration (ie. Number, text, picture, audio clip, and/or video)
Classifying	• Paths of elaborations and results that lead to particular specimens • All individual actions taken to construct the tree
Identifying	• Correct and incorrect hypotheses suggested by players • Reasons for choosing a particular hypothesis • Votes indicating which hypothesis players feel is the correct one
Other	• Real-time conversations though a chat interface • Asynchronous communication through a bulletin board system • Client-side actions that players have taken (eg mouse clicks, buttons pressed)

Implementing Gaming Templates

When classification trees are completed and validated they become input to game templates. These templates are essentially a system of game rules as predetermined by developers. Context is then applied to these templates based on the knowledge base developed during formative gamification.

The “Select Experiments” game and the “Identification” game are two template examples. During the “Select Experiments” game (Figure 56), students are given a description, name, and multimedia artifacts related to a previously identified specimen. They gain and lose points based on their selection of appropriate experiments and observations to identify that specimen. In the identification game, students are given the description, location, multimedia artifacts, and relevant experimental and observational results and receive points for each correctly identified specimen.



Select Experiments Game

Current Score
100

Question
2

Name
Flowering Dogwood

Multimedia

Description
short, small trunk, small leaves

Select Experiments

□ Branching Pattern □ Leaf Shape

Figure 56. A mockup of the select experiments game. The name, multimedia, description, and experiments to select are changed for each question and match specific specimens the group has identified. In this example, a player must click the empty boxes, selecting whether they feel that leaf shape or branching pattern is more important to identifying Flowering Dogwood.

Results and Discussion

CIRCLE holds great promise at improving student understanding of the classification and identification tasks. Future development will focus on the creation of more game templates. In addition to the “Select Experiments” game and the “Identification” game many more opportunities for game templates exist. Some ideas include:

- Multiplayer – Players take turns selecting experiments or identifying specimens.

- Timed – Players are timed while they select experiments and identify specimens. The fastest time with the highest accuracy gets the most points.

- Memory Game – The system places gathered multimedia specimens onto cards with two of each specimen face down. Players need to find the two cards that contained the same specimen. The fewest number of moves would receive the most points. This could be combined with a multiplayer template such that the individual who found the most pairs wins.

- Classi-Caching - (see Geocaching) - Users attempt to find other examples of the same specimen out in the real world. If they find an example, they mark the location in the game. Other players need to go to that location and either agree or disagree that it is the same as the original specimen. Points are given for the number of matching specimens found and level of agreement on whether the specimen matches the in-game specimen.

- Audio hide and seek – In this audio-only template, students would search for a sound that was similar to a specimen that was already collected. Players receive points for how close the sounds matched.

Conclusions and Future Work

CIRCLE was used by both STEM experts and undergraduate ecology students in 2014 (Borchert et al. 2015). The actions of each participant was logged to accurately capture their sequence. In addition, the participants were observed and formative feedback was solicited. Finally, a 10-item System Usability Scale (SUS 1986) and 4-item open ended questionnaire was administered as a means of gathering information about the ease of use of CIRCLE.

The open ended questions included “What things were good about CIRCLE? Why?”, “What was bad about CIRCLE? Why?”, “What should be changed about CIRCLE?”, and included an area to add additional comments. In the SUS instrument, users expressed an opinion on a five point Likert scale from Strongly Agree to Strongly Disagree. The responses to the open ended questions were then correlated with individual SUS questions. For example, SUS question #5 is “I found the various functions in CIRCLE were well integrated“ and user comments included (referring to using the color red as a visual cue in CIRCLE) “Red prompts for what to do next”

By contrast, SUS question #6 is “I thought there was too much inconsistency in CIRCLE” which prompted one user to add “Unable to go back and change things”, and another to suggest “add home button”. A total of about 20 concrete and implementable suggestions were provided.

The next version of CIRCLE will address many of the 20 suggestions offered by the pilot study group, and our next experiment will administer another SUS questionnaire, with the responses tracked by SUS question number. In this way, we can methodically track and measure the improvements in the next version using changes in SUS scores.

Future development will focus on improving usability, so that players learn to love this innovative approach towards retrieval learning.

EXPERIMENTS

A series of studies were performed to determine which educationally important variables could be measured using CIRCLE, if CIRCLE could be used to help students learn how to classify and identify real world objects, and if game templates could be created from user generated content to harness the power of retrieval learning. Development of the software began in January 2014. Beta testing of the first prototype occurred in May 2014.

Beta Test One - STEM Education Participants - Spring 2014

Twelve graduate students and faculty members in STEM Education tested the first prototype of CIRCLE in Spring 2014 (IRB Protocol: SM14234). These participants were members of a group which met weekly to discuss journal articles related to STEM Education. They came from a wide variety of disciplines including Biology, Chemistry, Physics, and Math. Some of the graduate students were taking the course for credit.

Participants began by completing the informed consent document. Students and faculty were placed into groups of three individuals. CIRCLE projects were created before the experiment to save time. Individuals were handed a piece of paper with their username and password.

A set of nine candies were used as specimens to be identified and classified: Cinnamon Bears, Three Musketeers, Pretzel M&Ms, Dark Chocolate M&Ms, Mike and Ike's, Rolos, Snickers, Spree, and Twix. Approximately 20 samples of each candy were opened and placed into a plastic bag. Each bag was labeled with a letter. In addition to the physical specimens, nutrition facts and an ingredient list were printed and placed next to the bags. The two M&M brands were selected to ensure the participants had a reason to read the ingredient list.

Participants used a provided laptop and any personally-owned mobile devices to access and use CIRCLE. Groups self-selected into the gatherer and elaborator roles. The gatherer took pictures and entered a description and location of the specimen. The elaborators suggested tests for the gatherer to perform to properly classify and identify the candy. All group members then suggested identities of the specimens and worked together to create a classification structure containing all of the candy.

After using CIRCLE, participants were given an adapted version of the ten item System Usability Scale (Brooke 1996) where the word System was replaced with CIRCLE (Appendix A). In addition to the SUS, four open ended questions were included: "What things were good about CIRCLE?", "What was bad about CIRCLE? Why?", "What should be changed about CIRCLE?", and "Any further comments?"

The results of this beta test were positive. The mean SUS score was 66.0 with a standard deviation of 15.4, which is in-line with other web based applications (Bangor, Kortum, & Miller 2008). Suggested features from the open ended questions included the ability to delete specimens and hypotheses. Minor fixes were applied, but overall the participants enjoyed the experience.

Table 6 shows the instructor-available log data gathered during the first beta test. Unidentified specimens are the specimens with the name "Unknown". The "Identified Specimens" entry corresponds to specimens identified by the group through the CIRCLE voting process. Chat messages refers to the number of real time chat messages sent by individual players. Experiments refers to the number of suggested experiments in the project. Multimedia per specimen is the average number of images, videos, or audio files associated with each specimen.

The logging data shows a relatively large number of still unidentified specimens. Some groups were able to complete the identification task for some of their specimens. Tree construction was limited by time constraints. There were no instructions on the number of multimedia to take of the specimens, so groups either took none or one picture of each specimen. The activity was face-to-face, so the number of chat messages is very low.

Table 6. Group level summary statistics in beta test one (n = 4).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Identified Specimens	6	1.5	1	1.91	0	4
Unidentified Specimens	41	10.25	7.5	5.85	7	19
Chat Messages	10	2.5	1.5	3.32	0	7
Experiments	14	3.5	3.5	0.577	3	4
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Multimedia Per Specimen		0.657	0.587	.254	0.455	1

Experiment One - Ecology Course - Fall 2014

After making improvements to the software, an experiment was conducted using a more realistic use case - identifying and classifying trees on a college campus (IRB Protocol: SM15030). CIRCLE was embedded into a unit on phenology in a General Ecology course. For the phenology project students examined a tree on campus to determine when and how fast its leaves dropped during the fall. The students were asked to identify and classify both their own phenology tree and a series of five to six other trees on campus of their own choosing. The instructor placed the forty-seven students into groups of two to four using the Comprehensive Assessment of Team Member Effectiveness (Layton, Loughry, Ohland & Ricco 2010). Variables used to balance the groups included: GPA, gender, race, schedule, and goal grade in the course.

Efforts were made to minimize a single minority or single female being placed in a group. Students were given a worksheet of instructions to follow (Appendix C). They had two class periods to create their accounts and CIRCLE projects, walk to their chosen trees, gather multimedia of the trees, suggest and perform experiments, identify, then classify the trees. These tasks were completed using CIRCLE on their own mobile devices, working together in face-to-face groups. A technician was available with spare laptops and mobile devices.

Students created their own account, which included the completion of a short demographic survey. These results are shown in Table 7. At the conclusion of the unit, students were asked to complete the same SUS plus open ended questions survey used during the beta test, plus a modified version of the twenty item Social Space Scale (Kreijns et al. 2004) (Appendix B).

Additional logging data was collected during the first experiment (Table 8). A bug during beta testing resulted in the number of results per experiment not being saved to the server. This was fixed in the experiment, so the number of results in the project is shown in the table. The classification tree interface was also playable during the experiment. Scratch trees refers to the number of trees beyond the primary tree that the group could use in a "scratch paper" like interface. Tree modifications indicates the number of times the tree was changed. This includes adding a result, replacing an empty node with a specimen or experiment, moving, or deleting nodes. Out degree refers to the number of outgoing edges from a node. The out degree of the primary tree is equal to the number of edges on the primary tree. Primary Tree Depth is the number of edges from the root node to the leaf on the longest path in the tree. Asynchronous communication was added for this study. Board Messages Posted is the number of board messages written by players. Board Messages Read are the number of times all messages were read by players. Experiment Node Children is the average number of child nodes for the experiment nodes in the tree. This is equivalent to the number of results in the tree. Experiments Per Specimen in Tree is the total number of edges (which is equivalent to the number of results in the tree) divided by the number of specimens in the tree.

Table 7. Experiment one demographics (n=55).

Category	**Total (%)**	**Category**	**Total (%)**
Gender		**Race**	
Male	19 (34.5%)	Asian	2 (3.6%)
Female	26 (47.3%)	American Indian or Alaska Native	1 (1.8%)
Prefer not to answer	10 (18.2%)	White	39 (70.9%)
GPA		Black or African American	1 (1.8%)
4.0-3.5	24 (43.6%)	Two or more racial categories	1 (1.8%)
3.5-3.0	17 (30.9%)	Prefer not to answer	10 (18.1%)
3.0-2.5	13 (23.6%)	**Predicted Grade in Course**	
2.5-2.0	1 (1.8%)	A	40 (72.7%)
Grade Level		B	14 (25.5%)
Freshman	7 (12.7%)	C	1 (1.8%)
Sophomore	10 (18.2%)	**Ethnicity**	
Junior	19 (34.5%)	Not Hispanic or Latino	41 (74.5%)
Senior	19 (34.5%)	Hispanic or Latino	1 (1.8%)
		Prefer not to answer	13 (23.6%)

There are a number of conclusions that can be drawn from the data. The low number of board and chat messages is unsurprising because individuals were able to communicate face-to-face. Teams were told to take at least three pictures of each specimen which is borne out in the data. It was clear that students didn't understand the nature of scratch trees. They either used many trees (44) or none at all. This feature ended up being removed in future versions. Experiment Node Children and Results Per Specimen in Tree are different because students could assign or not assign a specimen to an empty node. The data suggest that not many students placed specimens into the empty nodes in the tree. A number of issues surfaced when looking at the classification structures. Groups worked hard trying to complete the classification task (one group had 286 tree modifications), but only one group finished with a valid tree. The primary reason for this difficulty was that students didn't understand the nature of the suggest and perform experiment tasks. While the intent was for students to suggest individual experiments with multiple possible results, they tended to create either multiple experiments with at most two results or a single experiment with all possible results. So, for example, rather than creating three separate experiments for 1) leaf type, 2) branching pattern, and 3) leaf shape, one group only created a single experiment called observation with

> "The leaf shape is oval.
> The branching pattern is opposite.
> The leaves are compound."

as one of the results. This limited the depth of the classification structure, as seen by the primary tree depth. While these experimental results are valid and useful, they are not easily used by the CIRCLE interface. Experiment two (described below) uses some of these experimental results with great success.

User issues are borne out in the statistics. The average number of experiments per specimen was much too low to create valid classification trees (mean = 1.03). In order to distinguish between different specimens in CIRCLE, there must be different experiments with a variety of experimental results. Once they attempted to create the classification structures, half of the groups had an experimental node degree of two or less. This means that there were only one or two results added to an experiment node. There were very few results to distinguish between specimens on the classification structure.

There were some potential limitations of this experiment. Most notably, NDSU's racial diversity is quite low, which is borne out in the demographic data. Further experimentation will be needed with a more diverse population to ensure that it is usable for those groups as well. In addition, classrooms in different disciplines will need to be tested as well. There are opportunities for identification and classification throughout the sciences that could benefit from CIRCLE.

Table 8. Group level summary statistics in experiment one (n=16).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Identified Specimens	145	9.063	9	5.67	0	20
Unidentified Specimens	113	7.063	8.5	5.50	0	16
Chat Messages	30	1.875	1	2.28	0	7
Experiments	90	5.625	5.5	3.46	1	14
Results	244	15.25	14	11.92	1	51
Scratch Trees	133	8.313	5.5	10.40	0	44
Tree Modifications	1336	83.5	55.5	83.61	0	286
Out Degree of Primary Tree	91	5.688	3	7.53	0	29
Depth of Primary Tree	16	1	1	0.63	0	2
Board Messages Posted	5	0.313	0	0.79	0	3
Board Messages Read	5	0.313	0	0.79	0	3
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Multimedia Per Specimen		3.301	3.237	0.29	2.8	3.81
Results Per Specimen in Tree		1.03	1	0.09	1	1.28
Experiment Node Children		4.69	4	4.36	0	14

The Social Space scores were generally positive. The mean score for the positive group behavior subscale of the social space survey was 33.24, with a standard deviation of 7.70 (n=47). For comparison, the best possible score, meaning the most positive group behaviors, is 50. The mean score for the negative group behavior subscore was 15.68, with a standard deviation of 5.70 (n=47). The best possible score for the negative group behavior subscale is 10.

The mean SUS score for the first experiment was 41.1, well below average for an application like CIRCLE. Student answers to the open ended questions provide insights into the low SUS score and logging data. Answers were categorized to identify the most important issues to address. The top five answers to “What was bad about CIRCLE?” were related to tree construction, not liking working in groups, being unsure what to do next, difficulty in navigating, and bugs in the software. The top five answers to “What should be changed about CIRCLE?” were: improve directions, improve tree construction, make the interface more user friendly, and remove group work. Despite student objections, group work was not removed. This is a hallmark of the CIRCLE experience and is grounded in educational theory. The other issues were matched with specific developer tasks in an effort to improve system usability.

Tree construction improvements included 1) a complete rewrite of the classification tree interface as described in the implementation; 2) the creation of a tutor dialog box that appears when students unlocked a component of their classification tree: specimens, experiments, and results; 3) creation of a tutorial video for developing a classification tree; 4) inclusion of embedded node data, allowing users to see the information related to a particular node in the tree. For example, clicking on a specimen would show its multimedia, name, and experiments in the information panel.

Directions were improved by implementing a series of tutor messages for each stage of the CIRCLE process, with special emphasis on interface elements. This is related to the issue of being unsure what to do next. In addition to the tutor messages, context-sensitive red text was used to indicate what the user needed to do next. For example, when students are required to vote for a particular specimen, red text is shown next to those specimens in the list.

Difficulty in navigation was improved by creating the cookie crumb navigation bar described in the implementation section. Prior to this, the navigation buttons were always on the bottom, and it was not always clear to students how to navigate the interface. A graphical site map (Peruri, in preparation) was also developed.

User friendliness was increased by adopting the JQueryUI framework. This allowed the developers to quickly create more graphically pleasing and consistent interfaces without the assistance of a trained artist.

In addition to these written comments, it was noted by the instructor that students were not sure why they were performing the tree classification task. This was primarily because students were performing identifications using Internet-supplied dichotomous keys. This issue was alleviated by swapping the identification and classification tasks, so that classification came first. This also allowed for the ability of the classification tree to handle two specimens with the same experimental results, giving them both the same identification through the use of specimen groups as described in the implementation.

Beta Test Two - STEM Education Participants - Spring 2015

After the implementation of these changes, a second beta test was conducted to identify technical issues in the software (IRB Protocol: SM14234). Nineteen graduate students and faculty members from the same group used during the first beta test volunteered to use CIRCLE. Some participants had used CIRCLE before. For others, the experience was brand new.

Nine candies were chosen for classification and identification: Peanut M&M, Almond M&M, Skittles, Twix, Rolos, Almond Joy, Snickers (x2), Milky Way. Two specimens of Snickers were used to simulate the case where multiple specimens with the same identity are found and need to be classified and identified. Participants self-assigned into separate gathering and elaborating roles. The gatherer took pictures of the specimens, wrote a short description for the specimen, and completed suggested experiments. Elaborators suggested experiments. After elaboration, participants were told to collaboratively construct classification trees, identify the candy, and play games based on their chosen identities and classification trees. At the completion of the task, they were asked to complete the system usability scale survey with open ended questions.

Table 9 shows group level logging data for the second beta test. This test marked the change from the D3.js version of classification to the GoJS version described in the implementation. Thus, tree modification logging was replaced with Diagram Saves. Diagram Saves occurred whenever the player added or removed an experiment from the tree, results for the experiment are added as leaves automatically in the GoJS version.

This cohort had the same issues with experiments and results found in experiment one. A very low number of experiments per specimen, along with not well specified experiments and results, ended with a difficulty in completing the classification task. Since the classification task was uncompleted, no groups reached the identification or game play tasks. Chat and board messages were low because experiments were conducted face-to-face in a single classroom session. Groups were not given directions for how many images to take, so they ended up taking between zero and two images per specimen, which matches our expectations.

Table 9. Group level summary statistics in beta test two (n=8).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Identified Specimens	0	0	0	0	0	0
Unidentified Specimens	70	8.75	9	1.75	5	11
Chat Messages	19	2.375	0.5	4.03	0	11
Experiments	32	4	4.5	1.77	2	6
Results	84	10.5	9	6.97	5	27
Diagram Saves	248	31	32	11.43	14	44
Out Degree of Primary Tree	92	11.5	9.5	8.37	2	30
Max Depth of Primary Tree	20	2.5	2.5	0.93	1	4
Board Messages Posted	4	0.5	0	1.07	0	3
Board Messages Read	5	0.625	0	1.41	0	4
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Multimedia Per Specimen		0.80	0.75	0.68	0	1.89
Experiment Node Degree		5.01	3.7	2.84	2	9

Additional individual level logging was included in the second beta test (Table 10). This logging looked at the tutoring system's effectiveness (Tutor Dialogs Closed and Tutor Duration) and the site map's usage (Site Map Clicks). Tutor messages were read, on average, approximately 4.3 seconds. While some of the messages were short, that implies that a number of the dialogs were closed without reading them. In addition, the site map functionality was not used, with only 2 of the 16 individuals clicking the site map button.

Table 10. Individual level summary statistics in beta test two (n=16).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Tutor Dialogs Closed	836	52.25	50.5	6.97	42	62
Site Map Clicks	9	0.56	0	2.00	0	8
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Tutor Duration (ms)		4299	3705	1640	2555	7451

Experiment Two - Campuswide Volunteers - Summer 2015

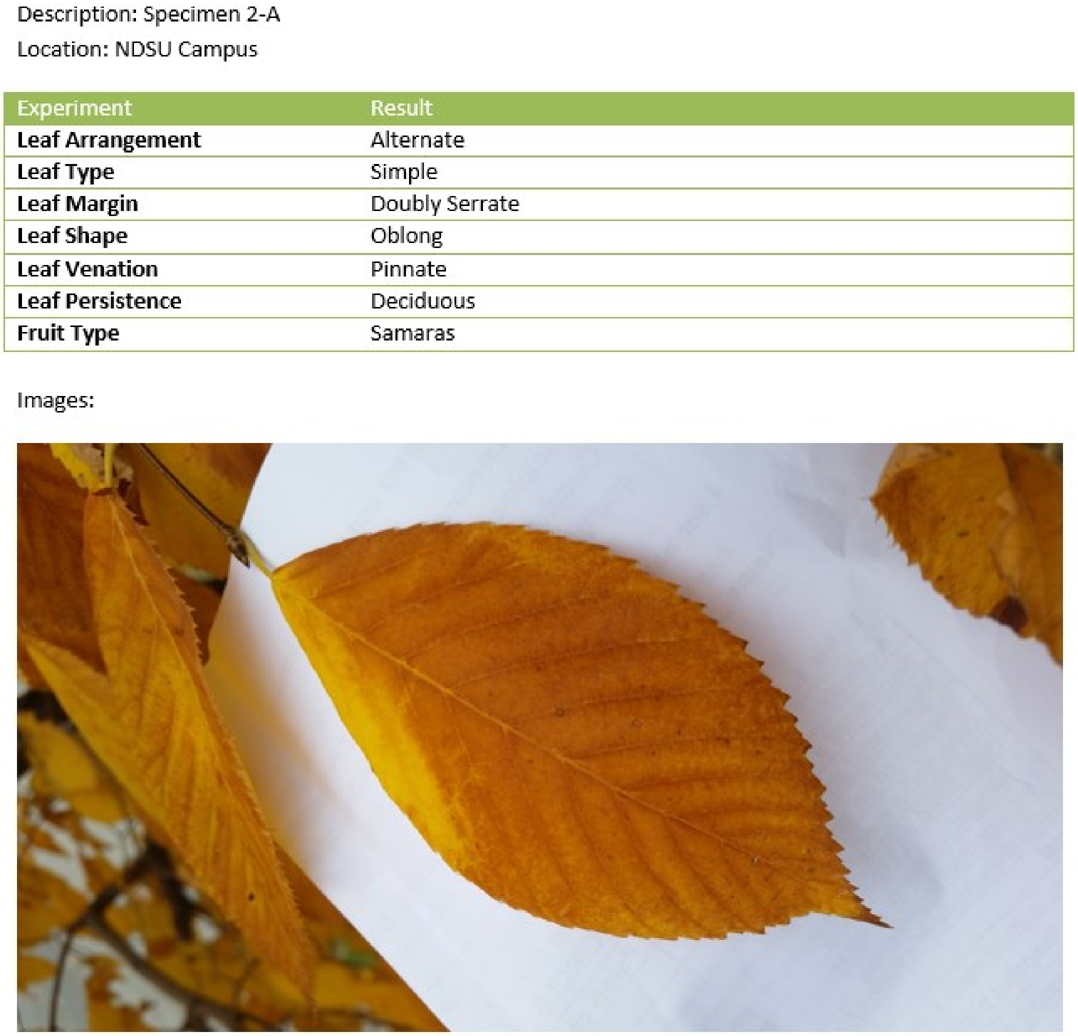
Many fixes were applied to CIRCLE between Fall 2014 and Summer 2015. The main addition was the inclusion of the ability to add experimental groups. An experimental group refers to a pair of individuals that will work on CIRCLE together on a specific topic. For these groups, pre-selected experiments and experimental results are loaded into the interface. In this specific case, rather than having novice students suggest experiments and elaborations, CIRCLE provided the experiments and results for them. Students were allowed to add their own results, but a full list of results with associated help messages was included. Logging was also included to determine participant time on task.

A second experiment was conducted using these experimental groups (IRB Protocol: SM15145). Participants were recruited for the experiment through mailing list messages to all graduate students in the College of Science and Mathematics and to a broader list of individuals interested in research participation throughout the university. They were directed to an online form which asked potential users to electronically sign an informed consent document and complete a demographic survey and availability chart (See Appendix D). Students were scheduled to work in pairs for four hours on all five CIRCLE tasks (gathering, elaborating, classifying, identifying, and playing). A developer was available for technical support, but did not interact with the students unless help was specifically requested. Users were allowed to use their own mobile device, or they could use the provided netbook, Android tablet or iPod Touch. They were encouraged to use a laptop for the classification task because the classification structure fit better on a larger screen. Peer help and gamification were disabled for this study to focus on the use of CIRCLE as a tool, rather than as a gamified application.

Table 11. Experiment two demographics (n=30).

Category	**Total (%)**	**Category**	**Total (%)**
Gender		**Race**	
Male	14 (46.7%)	Asian	12 (40%)
Female	15 (50%)	Black or African American	3 (10%)
Prefer not to answer	1 (3.3%)	White	14 (46.7%)
GPA		Prefer not to answer	1 (3.3%)
4.0-3.5	16 (53.3%)	**Major By College**	
3.5-3.0	0 (0%)	Science and Math	10 (33.3%)
3.0-2.5	11 (36.6%)	Engineering and Architecture	7 (23.3%)
2.5-2.0	3 (10%)	Health Professionals	5 (16.7%)
Grade Level		Arts, Humanities & Social Science	4 (13.3%)
Sophomore	5 (16.7%)	Human Development & Education	2 (6.7%)
Junior	5 (16.7%)	Agriculture	1 (3.3%)
Senior	8 (26.7%)	Business	1 (3.3%)
Graduate Student	12 (40%)	**Ethnicity**	
		Not Hispanic or Latino	26 (86.7%)
		Prefer not to answer	4 (13.3%)

Test subjects were given a preselected set of trees, including images, experiments, and experimental results (Figure 57). Images were taken from student data gathered during the first experiment. Experiments and results were gathered from online sources coupled with data from the first experiment to create the preselected set. They were also given a binder of pages taken from a paper-based field guide (Martine 1998) corresponding to the correct identifications for the specimens. Students were asked to create a classification tree that didn't show any errors and play the select experiments game in order to complete the project.



Description: Specimen 2-A
Location: NDSU Campus

Experiment	Result
Leaf Arrangement	Alternate
Leaf Type	Simple
Leaf Margin	Doubly Serrate
Leaf Shape	Oblong
Leaf Venation	Pinnate
Leaf Persistence	Deciduous
Fruit Type	Samaras

Images:

Figure 57. Example page from a specimen binder. It shows the specimen description and location, observations and results of performing those observations, and the first of three or four images.

At the completion of the project, participants were required to complete the post assessment survey that included the same SUS and social space measures given in the first experiment, to measure usability and sociability changes. Once the survey was completed, they were paid for their time.

Group level statistics for the second experiment were the same as those gathered in the second beta test (Table 12). One group was unable to complete the identification task because one of the group members didn’t show up for followup meetings. Otherwise, all groups were able to identify all the specimens. Again, the experiments were performed face-to-face, so the number of chat and board messages were low. Experiments were provided by the game, therefore, all groups had the same number of experiments. Players were able to add their own results. One of the experimental results in the binders mentioned an “odd pinnately-compound” leaf arrangement, but the CIRCLE interface only listed “simple” and “compound” leaf arrangements. Most groups manually added the “odd pinnately-compound” leaf arrangement. The classification tree structures improved dramatically from prior experiments with a large enough depth to accurately identify all of the specimens. In the binder, there were either three or four images per specimens, they generally took a picture of each image in the binder.

In addition to the individual level logging done in the second beta test, statistics gathered during the second experiment included time on task and game scores. When the client sends a *BeginTask* event, a LogElement is stored on the server, these LogElements coupled with those stored when the player disconnects were used to establish the amount of time the players spent on each task. In addition, since these students were able to complete the select experiments game, the high scores for the game templates were included.

Again, the site map interface was not well utilized, with only 13 clicks, with most students not clicking the button at all. Students in this experiment read the tutor messages longer than in the second beta test ($p<0.003$ on a two-tailed t-test). The maximum possible score in the game is dependent on the number of specimens a group has gathered. Two groups were unable to complete the playing task because of lack of participation.

Table 12. Group level summary statistics in experiment two (n=15).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Identified Specimens	226	15.067	14	8.54	0	43
Unidentified Specimens	14	0.933	0	3.61	0	14
Chat Messages	7	0.467	0	1.06	0	4
Experiments	105	7	7	0	7	7
Results	448	29.867	30	0.64	29	31
Diagram Saves	1023	68.2	31	132.90	8	545
Out Degree of Primary Tree	1714	114.267	38	279.60	21	1108
Depth of Primary Tree	84	5.6	6	1.35	4	8
Board Messages Posted	3	0.2	0	0.41	0	1
Board Messages Read	4	0.267	0	0.59	0	2
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Multimedia Per Specimen		2.73	3.07	0.75	1	3.36
Experiment Node Degree		5.07	4.11	2.37	2.22	10

Limitations of this experiment relate to the time period the experiment was run. Since the experiment was run during the summer semester, it meant that there was a more diverse population, but it also included a very large gap in student scholastic ability as measured by grade point average. Students in summer school are either particularly motivated or need the courses in order to pass, which is shown in the GPA data in the demographics table. More experimentation of other groups is necessary.

Thirty test subjects used CIRCLE, twenty-eight finished and completed the SUS and Social Space survey. The Social Space scores were again generally positive. The mean score for the positive group behavior subscale of the social space survey was 33.93, with a standard deviation of 8.09 (n=28). The mean score for the negative group behavior subscore was 14.64, with a standard deviation of 3.90 (n=28). Both the positive (p=0.7161) and negative (p=0.3973) were not statistically different from the first experiment.

Table 13. Individual level summary statistics in experiment two (n=30).

Numeric Data	**Total**	**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Tutor Dialogs Closed	1357	45.233	45	8.90	32	70
Site Map Clicks	13	0.433	0	0.73	0	3
Time on Task (Viewing Project)*	46620	1554	903	1436	190	5276
Time on Task (Gathering)*	45537	1518	1397	782	564	3747
Time on Task (Elaborating)*	50272	1676	1243	970	646	3890
Time on Task (Classifying)*	49424	1647	1745	1227	0	4603
Time on Task (Identifying)*	60410	2014	1640	1561	131	6628
Time on Task (Playing)*	49259	1642	1546	1274	0	5033
Time on Task (Total)*	301524	10051	9687	3531	3377	15247
Game Template High Score	20800	867	868	535	0	2115
Average Data		**Mean**	**Median**	**Std Dev**	**Min**	**Max**
Tutor Duration (ms)		8561	6649	5219	3127	26170

* Time on Task measurements are in seconds

The mean SUS score for this study was 68.0, with a standard deviation of 15.3, this is significantly better than the score in first experiment (p < .001 on a two tailed between subjects t-test). Comments were generally favorable. “Circle is a very educational website. It is good for us to learn more about tress[sic].” and “It's innovative and people can work together and interact.”

Negative comments were centered around the help system, classification tree, and game play. Participants felt that there were too many unavoidable tutors, so it may be helpful in future studies to cut back on the number of dialogs, perhaps by having a hint button that users can click instead. Participants also had issues with the classification tree. Animations tended to run fairly slow on older laptops, and there was a very large amount of scrolling required to view the entire tree. Interface improvements need to be identified. Finally, participants felt confused by the select experiments game. It was not immediately obvious to them that their own classification tree was being used to create the games. As a result, they became quite frustrated when they lost points, even though they were generally correct. A better approach will be to only increase student points. Future studies will be performed to alleviate these concerns.

CONCLUSIONS AND FUTURE WORK

CIRCLE addresses three key features missing from current approaches in classification and identification pedagogy. CIRCLE implements a series of collaborative online tasks for identifying and classifying real world objects that are typically done by hand; applies formative and summative gamification through motivating constructivist user-generated content in contrast to memorization-based techniques, and provides an interface for instructors and researchers to view student progress and identify relevant variables associated with improving student outcomes during the identification and classification task.

This project addressed the problem of manual classification through a careful literature review that identifies current theories on student learning including motivation, mobile learning, gamification, and collaborative learning. These theories were implemented into the CIRCLE software through the five primary tasks: gathering, elaborating, classifying, identifying, and playing. Experimental results showed that usability scores improved as changes were applied through the iterative design. We suggest that the gathering/elaborating tasks should be separated from the classification/identification/game play tasks to lessen the cognitive load (Sweller 1994) on students, as the elaborating task in particular is quite difficult.

CIRCLE implements formative gamification through points distributed “during use” and summative gamification through the development of “after use” game templates. We developed a game template named “Select Experiments Game” and tested it during experiment two. Usability scores indicate that students were able to use the games. Written comments indicate that further refinement of point distribution is required. It would also be useful to tell students that their classification structures are used in the creation of the classification / identification games, so that they are not confused by why they are losing points.

CIRCLE tracks a number of potentially educationally relevant variables, as shown in the experiments. Numeric variables included time on task, number of identified and unidentified specimens, experiments, results, chat messages, bulletin board messages posted and read, primary tree depth, degree of the primary tree, diagram saves, tutor dialogs closed, and site map clicks. Mean statistic variables include tutor duration, experiment node degree, and multimedia per specimen. These variables guided the development of the software and may serve as a foundation for future STEM education research identifying how students move from novice to expert procedures and understanding while building classification trees.

Future Work

A number of avenues for future research have been identified. These include the development of classroom materials, creation of more game templates, and the examination and elaboration of STEM education theories based on CIRCLE results,

We contacted three different instructors early in CIRCLE development to gauge interest in the software. The Ecology professor was enthusiastic and used CIRCLE in her classroom as experiment one. Others were reluctant because of the early stage of CIRCLE development. There were too many bugs and user interface issues at the start of the project. These user interface and initial software issues have been corrected. Others did not feel students would use the software without receiving monetary compensation or course credit. Developing curricular materials will be necessary to adequately address this concern. Materials will include lesson plans, video and text tutorials, a more usable instructor interface, and further software developments, including items tracked using our bug tracking system.

Other future development will focus on the creation of more game templates. In addition to the “Select Experiments” game many more opportunities for game templates exist. Some ideas include:

- Multiplayer – Players take turns selecting experiments or identifying specimens.
- Timed – Players are timed while they select experiments and identify specimens. The fastest time with the highest accuracy gets the most points.
- Memory Game – The system places gathered multimedia specimens onto cards with two of each specimen face down. Players need to find the two cards that contain the same specimen. The fewest number of moves would receive the most points. This could be combined with a multiplayer template such that the individual who found the most pairs wins.
- Classi-Caching - (see Geocaching) - Users attempt to find other examples of the same specimen out in the real world. If they find an example, they mark the location in the game. Other players need to go to that location and either agree or disagree that it is the same as the original specimen. Points are given for the number of matching specimens found and level of agreement on whether the specimen matches the in-game specimen.
- Audio hide and seek – In this audio-only template, students would search for a sound that was similar to a specimen that was already collected. Players receive points for how close the sounds matched.
- Tic Tac Toe Quiz (TRIS-Q) (Tsai, Tsai, & Lin 2015) - In this modified Tic-Tac-Toe game, students are given a multiple choice question. When they answer correctly, they can place their own piece on the board. When they answer incorrectly, they must place a

piece for their opponent. Multiple choice questions could be developed from information about the specimens, classifications, and identifications.

Data extracted from CIRCLE promises to enhance our understanding of how students learn, especially when identifying and classifying objects. Observing students while creating trees provides a snapshot of what the students know, over time multiple trees show how students move from novice users to experts. Bulletin board and chat messages could be correlated to student behaviors, both positive and negative. Interventions based on these correlations could improve student outcomes. Data mining techniques could discover non-obvious patterns in student / CIRCLE interaction. Social network analysis applied to CIRCLE relationships could identify outliers, encouraging them to interact with the group.

CIRCLE offers opportunities for a number of different populations. Educational researchers could use CIRCLE as a tool for exploring how individuals gain expertise. One STEM Education researcher has shown interest in using CIRCLE to see if student-built trees mimic expert-produced trees; how quickly students create a usable tree, what reasoning students use when building trees and how is it different from expert reasoning, and determining whether the creation task actually help the student memorize taxonomies (J. Momsen, personal communication, July 24, 2015). Instructors of taxonomy or systematics class would also find the tool useful, being able to keep track of learning progress made by students in their classrooms. Ultimately, CIRCLE holds great promise to discover new theories, elaborate on previous theories, and help students learn how to classify and identify objects.

REFERENCES

App Shopper (2013). Game Fish Identification by Caranx Informatica Ltda. Retrieved from http://appshopper.com/games/game-fish-identification on September 9, 2013.

AmericanTorque.com (2013). Engine Identification Game. Retrieved from http://www.americantorque.com/game/engine-id/ on September 9, 2013.

Asbell-Clarke, J., Rowe, E., & Sylvan, E. (2013). Working through Impulse: Assessment of Emergent Learning in a Physics Game. Paper presented at the Games+Learning+Society 9.0 (GLS 9.0) Conference, Madison, WI.

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. Intl. Journal of Human–Computer Interaction, 24(6), 574-594.

Black, P., & Wiliam, D. (1998). Assessment and Classroom Learning. Assessment in Education: Principles, Policy & Practice, 5(1), 7-74.

Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. 13(6), 4-16.

Borchert, O. (2008). Computer Supported Collaborative Learning in a Multiplayer Online Game. Master’s Thesis. Fargo, ND: North Dakota State University.

Borchert, O., Hokanson, G., Peruri, A., & Slator, B. M. (2015). Here a Game, There a Game: classification, gamification, and retrieval learning. Proceedings of the 1st International Conference on Higher Education Advances (HEAd'15). Valencia, Spain. June 24-26.

Borchert, O., Hokanson, G., Slator, B. M., Vender, B., Yan, P., Aggarwal, V., Kariluoma, M., Marry, A., & Cosmano, B. (2013). A 3D Immersive Virtual Environment for Secondary Biology Education. Paper presented at the Society for Information Technology &

Teacher Education International Conference 2013, New Orleans, Louisiana, United States. http://www.editlib.org/p/48105

Borchert, O., & Slator, B. M. (2014). Building a Mobile Collaborative Learning Environment for the Identification and Classification of Real World Objects. Proceedings of the 1st International Conference on e-Learning e-Education and Online Training. European Alliance for Innovation: Bethesda, MD. Septemper 18-20.

Brachman, R. J. (1983). What is-a is and isn't: an analysis of taxonomic links in semantic networks. Journal Name: Computer; (United States); Journal Volume: 10, Medium: X; Size: Pages: 30-36.

Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). How People Learn: Brain, Mind, Experience, and School. National Academy Press: Washington, DC.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Brooke, J. (1996). SUS-A quick and dirty usability scale. Usability evaluation in industry, 189, 194.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated Cognition and the Culture of Learning. Educational Researcher, 18(1), 32-42. doi: 10.3102/0013189X018001032

Buehl, D. (2000). Backwards Design; Forward Thinking. Retrieved November 24, 2013, 2013, from http://www.weac.org/news_and_publications/education_news/2000-2001/read_backwards.aspx

Casey, S., Kirman, B., & Rowland, D. (2007). The gopher game: a social, mobile, locative game with user generated content and peer review. Paper presented at the Proceedings of the international conference on Advances in computer entertainment technology, Salzburg, Austria.

Chein, M., & Mugnier, M.-L. (2009). Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs. London, UK: Springer-Verlag London Limited.

Coldwell, W. (2013). Where in the world am I? The addictive mapping game that is GeoGuessr. The Independent. Retrieved from http://www.independent.co.uk/life-style/gadgets-and-tech/features/where-in-the-world-am-i-the-addictive-mapping-game-that-is-geoguessr-8641265.html on January 26, 2015.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem. User Modeling 2001. Springer Berlin Heidelberg, 137-147.

Crimson Trails (2013). Geobirds. Retrieved from http://geobirds.com/play on September 9, 2013.

Daugherty, T., Eastin, M., & Bright, L. (2008). Exploring Consumer Motivations for Creating User-Generated Content. Journal of Interactive Advertising, 8(2).

Deci, E. L., Eghrari, H., Patrick, B. C., & Leone, D. R. (1994). Facilitating internalization: The self-determination theory perspective. Journal of personality, 62(1), 119-142.

Deci, E. L., & Ryan, R. M. (2000). The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior. Psychological Inquiry, 11(4), 227-268.

Denis, G., & Jouvelot, P. (2005). Motivation-driven educational game design: applying best practices to music education. Paper presented at the Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining gamification. Paper presented at the Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments.

Deutsch, M. (1949). A Theory of Co-operation and Competition. Human Relations, 2(2), 129-152. doi: 10.1177/001872674900200204

Deutsch, M. (1962). Cooperation and trust: Some theoretical notes, Nebraska Symposium on Motivation, 1962 (pp. 275-320). Oxford, England: Univer. Nebraska Press.

Feltovich, P. J., Spiro, R. J., Coulson, R. L., & Feltovich, J. (1996). Collaboration within and among Minds: Mastering Complexity, Individually and in Groups. In T. Koschmann (Ed.), CSCL: Theory and Practice of an Emerging Paradigm (pp. 25-44). Mahwah, NJ: Lawrence Erlbaum Associates.

Ferguson, W., Bareiss, R., Birnbaum, L., & Osgood, R. (1992). ASK Systems: An Approach to the Realization of Story-Based Teachers. Journal of the Learning Sciences, 2(1), 95-134.

Fielding, R. (2000). Architectural styles and the design of network-based software architectures, Ph.D. thesis, UCI, Irvine, California.

Fisch, S. M. (2005). Making educational computer games educational. Paper presented at the Proceedings of the 2005 conference on Interaction design and children.

Fitz-Walter, Z., Tjondronegoro, D., & Wyeth, P. (2011). Orientation passport: using gamification to engage university students. Paper presented at the Proceedings of the 23rd Australian Computer-Human Interaction Conference.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. 19(1), 1-67.

Gang, Z., Wang, J. T. L., & Ng, P. A. (1996). Curriculum knowledge representation and manipulation in knowledge-based tutoring systems. Knowledge and Data Engineering, IEEE Transactions on, 8(5), 679-689.

Gaonkar, S., Li, J., Choudhury, R. R., Cox, L., & Schmidt, A. (2008). Micro-Blog: sharing and querying content through mobile phones and social participation. Paper presented at the

Proceedings of the 6th international conference on Mobile systems, applications, and services, Breckenridge, CO, USA.

Gee, J. P. (2003). What video games have to teach us about learning and literacy. New York, NY: Palgrave-MacMillan.

Geocaching. (2000). Accessed from http://en.wikipedia.org/wiki/Geocaching on January 30, 2015.

Ginsburg, G. S., & Bronstein, P. (1993). Family Factors Related to Children's Intrinsic/Extrinsic Motivational Orientation and Academic Performance. Child Development, 64(5), 1461-1474.

Goldman, S. V. (1996). Mediating Microworlds: Collaboration on High School Science Activities. In T. Koschmann (Ed.), CSCL: Theory and Practice of an Emerging Paradigm (pp. 45-81). Mahwah, NJ: Lawrence Erlbaum Associates.

Grambo, G. (1995). The Rock and Mineral Board Game. Science Activities: Classroom Projects and Curriculum Ideas, 32(2), 21-27.

Grinter, R. E., Aoki, P. M., Szymanski, M. H., Thornton, J. D., Woodruff, A., & Hurst, A. (2002). Revisiting the visit: understanding how technology can shape the museum visit. Paper presented at the Proceedings of the 2002 ACM conference on Computer supported cooperative work, New Orleans, Louisiana, USA.

Guay, F., Boggiano, A. K., & Vallerand, R. J. (2001). Autonomy Support, Intrinsic Motivation, and Perceived Competence: Conceptual and Empirical Linkages. Personality and Social Psychology Bulletin, 27(6), 643-650. doi: 10.1177/0146167201276001

Harter, S. (1981). A new self-report scale of intrinsic versus extrinsic orientation in the classroom: Motivational and informational components. Developmental Psychology, 17(3), 300-312. doi: 10.1037/0012-1649.17.3.300

Hayes-Roth, F., Waterman, D., & Lenat, D. (1984). Building expert systems. Reading, MA: Addison-Wesley.

Hill, C., & Slator, B. (2009). A Visualization Enhanced Hypertextbook for Computer Science Education. Paper presented at the Twelfth IASTED International Conference on Computers and Advanced Technology in Education (CATE 2009, St. Thomas, US Virgin Islands.

Hokanson, G., & Slator, Brian M. (2013). Development Tools for Content Creation in Virtual Environments. In Proceedings of E-LEARN 2013 - World Conference on E-Learning. (pp. 126-136). Las Vegas, NV, USA: AACE. October 21-24, 2013.

Hokanson, G., Borchert, O., Slator, B. M., Terpstra, J., Clark, J. T., Daniels, L. M., Anderson, H. R., Bergstrom, A., Hanson, T. A., Reber, J., Reetz, D., Weis, K. L., White, R., & Williams, L. (2008).Studying Native American culture in an immersive virtual environment. Paper presented at the Eighth IEEE International Conference on Advanced Learning Technologies (ICALT 2008).

Janis, I. L. (1972). Victims of Groupthink. Boston, MA: Houghton Mifflin Company.

Jenkins, H. (2002). Game Theory: How should we teach kids Newtonian physics? Simple. Play computer games. Technology Review, 29.

Jensen, M. (2013). The Anatomy Game (Web Anatomy II). Retrieved from http://msjensen.cehd.umn.edu/webanatomy_archive/game/ on September 9, 2013.

Johnson, D. W. (1970). Social psychology of education. New York, NY: Holt, Rhinehart and Winston.

Johnson, D. W., & Johnson, R. (1989). Cooperation and Competition: Theory and Research. Edina, MN: Interaction Book Company.

Johnson, D. W., & Johnson, R. T. (1991). Living Together and Alone: Cooperative, Competitive, and Individualistic Learning. Englewood Cliffs, NJ: Prentice Hall.

Johnson, D. W., Johnson, R., & Holubec, E. (1998). Cooperation in the Classroom (Seventh ed.). Edina, MN: Interaction Book Company.

Johnson, D. W., Johnson, R. T., & Smith, K. (2007). The State of Cooperative Learning in Postsecondary and Professional Settings. Educational Psychology Review, 19(1), 15-29.

Kagan, S. (1994). Cooperative Learning. San Juan Capistrano, CA: Kagan Cooperative Learning.

Kankaanranta, M., Neittaanmäki, P., Kuts, E., Islas-Sedano, C., & Sutinen, E. (2009). Let's Play Together with the Camera of Your Mobile Device Design and Use of Serious Games (Vol. 37, pp. 127-141): Springer Netherlands.

Kariluoma, M., Slator, B. M., Vender, B., Borchert, O., Hokanson, G., Yan, P., & Cosmano, B. (2013). The Design of Multiplayer Simulations for Online Game-Based Learning. Proceedings of the IASTED Technology for Education and Learning Conference. Marina del Rey, CA, USA. November 11 - 13.

Karpicke, J. D. (2012). Retrieval-Based Learning: Active Retrieval Promotes Meaningful Learning. Current Directions in Psychological Science, 21(3), 157-163. doi: 10.1177/0963721412443552

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. 29(2), 119-127.

Kids Know It Network (2013). Identify Rock Types Game. Retrieved from http://www.kidsgeo.com/geology-games/rocks-game.php on August 26, 2013.

Kinder Web Games (2013). Letter Learning Game. Retreived from http://kinderwebgames.com/a.html on September 9, 2013.

Klopfer, E., Sheldon, J., Perry, J., & Chen, V. H. H. (2011). Ubiquitous games for learning (UbiqGames): Weatherlings, a worked example. Journal of Computer Assisted Learning, 28(5), 465-476.

Kornell, N. (2009). Optimising learning using flashcards: Spacing is more effective than cramming. Applied Cognitive Psychology, 23(9), 1297-1317.

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-Supported Problem Based Learning: A Principled Approach to the Use of Computers in Collaborative Learning. In T. Koschmann (Ed.), CSCL: Theory and Practice of an Emerging Paradigm (pp. 100). Mahwah, NJ: Lawrence Erlbaum Associates.

Kreijns, K., Kirschner, P. A., Jochems, W., & Van Buuren, H. (2004). Measuring perceived quality of social space in distributed learning groups. Computers in Human Behavior, 20(5), 607-632.

Kreijns, K., Kirschner, P. A., Jochems, W., & Van Buuren, H. (2007). Measuring perceived sociability of computer-supported collaborative learning environments. Computers & Education, 49(2), 176-192.

Kreijns, K., Kirschner, P. A., Jochems, W., & Van Buuren, H. (2011). Measuring perceived social presence in distributed learning groups. Education and Information Technologies, 16(4), 365-381.

Kukulska-Hulme, A., Traxler, J., & Pettit, J. (2007). Designed and user-generated activity in the mobile age.Journal of Learning Design, 2(1), 52-65. doi: doi:10.5204/jld.v2i1.28

Landers, R.N. & Callan, R.C. (2011). Casual Social Games as Serious Games: The Psychology of Gamification in Undergraduate Education and Employee Training. Serious Games and Edutainment Applications, 399–423.

Layton, R. A., Loughry, M. L., Ohland, M. W., & Ricco, G. D. (2010). Design and validation of a web-based system for assigning members to teams using instructor-specified criteria. Advances in Engineering Education, 2 (1), 1-28.

Lee, C. S., Goh, D. H.-L., Chua, A. Y. K., & Ang, R. P. (2010). Indagator: Investigating perceived gratifications of an application that blends mobile content sharing with gameplay. Journal of the American Society for Information Science and Technology, 61(6), 1244-1257.

Lepper, M. R., Corpus, J. H., & Iyengar, S. S. (2005). Intrinsic and Extrinsic Motivational Orientations in the Classroom: Age Differences and Academic Correlates. Journal of Educational Psychology, 97(2), 184-196. doi: 10.1037/0022-0663.97.2.184

Leung, L. (2009). User-generated content on the internet: an examination of gratifications, civic engagement and psychological empowerment. New Media & Society, 11(8), 1327-1347. doi: 10.1177/1461444809341264

Levene, M. (2010). An Introduction to Search Engines and Web Navigation. John Wiley & Sons: Hoboken, NJ. pp. 221-222.

Malone, T. W. (1980). What makes things fun to learn? A study of intrinsically motivating computer games. Palo Alto, CA: XEROX Palo Alto Research Center.

Martine, C. T. (1998). Trees of New Jersey and the Mid-Atlantic States. New Jersey Department of Environmental Protection Division of Parks and Forestry Park Service Forest Research Education Center.

McAuley, E., Duncan, T., & Tammen, V. V. (1989). Psychometric properties of the Intrinsic Motivation Inventory in a competitive sport setting: A confirmatory factor analysis. Research quarterly for exercise and sport, 60(1), 48-58.

Miller, G. A. (2003). The cognitive revolution: a historical perspective. Trends in cognitive sciences, 7(3), 141-144.

Montola, M. (2005). Exploring the edge of the magic circle: Defining pervasive games. Paper presented at the Proceedings of DAC.

Murayama, K., Matsumoto, M., Izuma, K., & Matsumoto, K. (2010). Neural basis of the undermining effect of monetary reward on intrinsic motivation. Proceedings of the National Academy of Sciences. doi: 10.1073/pnas.1013305107

New England Aquarium (2013). Right Whale Identification Game. Retrieved from http://www.neaq.org/education_and_activities/games_and_activities/online_games/right_whale_identification_games.php on August 26, 2013.

Newman, R. S. (1990). Children's help-seeking in the classroom: The role of motivational factors and attitudes. Journal of Educational Psychology, 82(1), 71-80. doi: 10.1037/0022-0663.82.1.71

Nist, L., & Joseph, L. M. (2008). Effectiveness and Efficiency of Flashcard Drill Instructional Methods on Urban First-Graders' Word Recognition, Acquisition, Maintenance, and Generalization. School Psychology Review, 37(3), 294-308.

Norem-Hebeisen, A. A., & Johnson, D. W. (1981). The relationship between cooperative, competitive, and individualistic attitudes and differentiated aspects of self-esteem1. Journal of Personality, 49(4), 415-426.

Palincsar, A. S. (1998). Social Constructivist Perspectives on Teaching and Learning. Annual Review of Psychology, 49(1), 345-375.

Panitz, T. (1999). Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning: ERIC Clearinghouse.

Papert, S. (1998). Does Easy Do It? Children, Games, and Learning. Game Developer, 88.

Pea, R. (2009). The International Society of the Learning Sciences: ISLS Vision. Retrieved December 1, 2013, from http://www.isls.org/ISLS_Vision_2009.pdf

Perry, J., & Rosenheck, L. (2012). UbiqBio: A Playful Approach to Learning Biology with Mobile Games Paper presented at the 2012 Conference for the International Society for Technology in Education, San Diego, California.

Peruri, A. (in preparation). Unpublished Master’s Paper. North Dakota State University, Fargo, ND.

Plant, R. W., & Ryan, R. M. (1985). Intrinsic motivation and the effects of self-consciousness, self-awareness, and ego-involvement: An investigation of internally controlling styles. Journal of Personality, 53(3), 435-449.

Purpose Games (2013a). Brain Structure Identification Retrieved from http://www.purposegames.com/game/brain-structure-identification-game on September 9, 2013.

Purpose Games (2013b). 50 State Identification Game. Retrieved from http://www.purposegames.com/game/50-state-identification-game-quiz on September 9, 2013.

Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106.

Quinlan, J. R. (1987). Simplifying decision trees. 27(3), 221-234.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan-Kaufmann Publishers, Inc.

Redish, E. F. (2004). A theoretical framework for physics education research: Modeling student thinking.arXiv preprint physics/0411149.

Richardson, V. (2003). Constructivist Pedagogy. Teachers College Record, 105(9), 1623-1640. doi: 10.1046/j.1467-9620.2003.00303.x

Ritterfeld, U., Cody, M., & Vorderer, P. (2009). Serious Games: Mechanisms and Effects. New York, NY: Routledge.

Romero, C., Ventura, S., & García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. Computers & Education, 51(1), 368-384.

Ryan, R. M. (1982). Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. Journal of personality and social psychology, 43(3), 450.

Ryan, R. M., Connell, J. P., & Plant, R. W. (1990). Emotions in nondirected text learning. Learning and Individual Differences, 2(1), 1-17.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions.25(1), 54-67.

Ryan, R. M., Koestner, R., & Deci, E. L. (1991). Ego-involved persistence: When free-choice behavior is not intrinsically motivated. Motivation and emotion, 15(3), 185-205.

Ryan, R. M., Mims, V., & Koestner, R. (1983). Relation of reward contingency and interpersonal context to intrinsic motivation: A review and test using cognitive evaluation theory. Journal of Personality and Social Psychology, 45(4), 736.

Saini-Eidukat, B., Schwert, D. P., & Slator, B. M. (2002). Geology explorer: virtual geologic mapping and interpretation. Computers & geosciences, 28(10), 1167-1176.

Schwabe, G., & Göth, C. (2005). Mobile learning with a mobile game: design and motivational effects.Journal of Computer Assisted Learning, 21(3), 204-216.

Sharan, S., & Sharan, Y. (1978). Small-group Teaching. Englewood Cliffs, NJ: Educational Technology Publications.

Shindler, J. (2010). Transformative Classroom Management: Positive Strategies to Engage All Students and Promote a Psychology of Success. San Francisco, CA: Jossey-Bass.

Singer, L., & Schneider, K. (2012). It was a bit of a race: Gamification of version control. Paper presented at the Games and Software Engineering (GAS), 2012 2nd International Workshop on.

Skinner, B. F. (1938). The behavior of organisms: An experimental analysis. Cambridge, MA: B.F. Skinner Foundation.

Slator, B. M., & Faroque, G. (1998). The Agents in an Agent-based Economic Simulation Model. Paper presented at the 11th International Conference on Computer Applications in Industry And Engineering (CAINE-98), Las Vegas, NV.

Slator, B. M., Beckwith, R., Brandt, L., Chaput, H., Clark, J. T., Daniels, L. M., Hill, C., McClean, P., Opgrande, J., Saini-Eidukat, B., Schwert, D. P., Vender, B., White, A. R. (2006). Electric Worlds in the Classroom: Teaching and Learning with Role-Based Computer Games. New York: Teachers College Press. Columbia University. 192 pages.

Slator, B. M., Saini-Eidukat, B., Schwert, D. P., Borchert, O., Hokanson, G., Forness, S., & Karilouma, M. (2011). The eGEO Virtual World for Environmental Science Education. Paper presented at the 2011 GSA Annual Meeting, Minneapolis, MN.

Slavin, R. E. (1995). Cooperative Learning (Second ed.). Boston, MA: Allyn and Bacon.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. International Journal of Artificial Intelligence in Education (IJAIED), 12, 40-62.

SUS. (1986). System Usability Scale. Retrieved from http://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html on January 28, 2015.

Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. Learning and instruction, 4(4), 295-312.

Thom, J., Millen, D., & DiMicco, J. (2012). Removing gamification from an enterprise sns. Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.

Tsai, F. H., Tsai, C. C., & Lin, K. Y. (2015). The evaluation of different gaming modes and feedback types on game-based formative assessment in an online learning environment. Computers & Education, 81, 259-269.

Tzuriel, D. (1989). Development of motivational and cognitive-informational orientations from third to ninth grades. 10(1), 107-121.

Vatrapu, R., Suthers, D., & Medina, R. (2008). Usability, sociability, and learnability: A CSCL design evaluation framework. Paper presented at the Proceedings of the 16th international conference on computers in education (ICCE 2008).

Vender, B., Borchert, O., Dischinger, B., Hokanson, G., McClean, P. E., & Slator, B. M. (2010). JavaMOO Virtual Cells for Science Learning. In Shalin Hai-Jew Editor, Virtual Immersive and 3D Learning Spaces : Emerging Technologies and Trends. IGI Global, 294-211.

Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. (M. Cole, Trans.). Cambridge, MA: Harvard University Press.

Wang, X. H., & Yang, B. (2003). Why Competition may Discourage Students from Learning? A Behavioral Economic Analysis. Education Economics, 11(2), 117-128.

White, A. R., McClean, P., & Slator, B. M. (1999). The Virtual Cell: A Virtual Environment for Learning Cell Biology. Paper presented at the Proceedings of the Tenth International Conference on College Teaching and Learning: Innovation in a Global Learning Community. April.

Wiggins, G., & McTighe, J. (1998). Understanding by Design. Alexandria, VA: Association for Supervision and Curriculum Development.

Wijers, M., Jonker, V., & Drijvers, P. (2010). MobileMath: exploring mathematics outside the classroom. ZDM, 42(7), 789-799.

Wingkvist, A., & Ericsson, M. (2011). A Survey of Research Methods and Purposes in Mobile Learning (pp. 1-17): IGI Global.

Wong, E. H., Wiest, D. J., & Cusick, L. B. (2002). Perceptions of autonomy support, parent attachment, competence and self-worth as predictors of motivational orientation and academic achievement: an examination of sixth- and ninth-grade regular education students. Adolescence, 37(146), 255-266.

Yap, K. K., Huang, T. Y., Dodson, B., Lam, M. S., & McKeown, N. (2010, August). Towards software-friendly networks. In Proceedings of the first ACM asia-pacific workshop on Workshop on systems. ACM, 49-54.

Yuan, Y., & Shaw, M. J. (1995). Induction of fuzzy decision trees. 69(2), 125-139.

Zhu, Q., Wang, R., Chen, Q., Liu, Y., & Qin, W. (2010, December). Iot gateway: Bridging wireless sensor networks into internet of things. In Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. IEEE, 347-352.

Zimmerman, B. J. (1986). Becoming a self-regulated learner: Which are the key subprocesses? , 11(4), 307-313.

Zimmerman, B. J. (1989). A social cognitive view of self-regulated academic learning. Journal of Educational Psychology, 81(3), 329-339. doi: 10.1037/0022-0663.81.3.329

Zimmerman, B. J. (1990). Self-Regulated Learning and Academic Achievement: An Overview. Educational Psychologist, 25(1), 3-17.

APPENDIX A. SYSTEM USABILITY SCALE SURVEY

This survey was given as a post-treatment assessment during all beta tests and experiments.

Username (NOT real name): ________________________

1) I…
    a) Liked using CIRCLE
    b) Thought using CIRCLE was ok
    c) Did not enjoy using CIRCLE
    d) Don’t know how I feel about playing CIRCLE
2) What things were good about CIRCLE? Why?
3) What was bad about CIRCLE? Why?
4) What should be changed about CIRCLE?

For questions 5-14, check a box that corresponds with your opinion of each statement ranging from strongly disagree to strongly agree.

I think that I would like to use CIRCLE frequently

Strongly Disagree   Strongly Agree

______   ______   ______   ______   ______

I found CIRCLE unnecessarily complex

Strongly Disagree   Strongly Agree

______   ______   ______   ______   ______

I thought CIRCLE was easy to use

Strongly Disagree   Strongly Agree

______   ______   ______   ______   ______

I think that I would need the support of a technical person to be able to use CIRCLE

Strongly Disagree   Strongly Agree

______   ______   ______   ______   ______

I found the various functions in CIRCLE were well integrated

Strongly Disagree   Strongly Agree

______   ______   ______   ______   ______

I thought there was too much inconsistency in CIRCLE

Strongly Disagree   Strongly Agree

______ ______ ______ ______ ______

I would imagine that most people would learn to use CIRCLE very quickly

Strongly Disagree Strongly Agree

______ ______ ______ ______ ______

I found CIRCLE very cumbersome to use

Strongly Disagree Strongly Agree

______ ______ ______ ______ ______

I felt very confident using CIRCLE

Strongly Disagree Strongly Agree

______ ______ ______ ______ ______

I needed to learn a lot of things before I could get going with CIRCLE

Strongly Disagree Strongly Agree

______ ______ ______ ______ ______

5) Any other comments?

APPENDIX B. MODIFIED SOCIAL SPACE SURVEY

This survey was given as post assessment during the experiments (Numbers start at 15 because it was given immediately after the SUS with the Any other comments question asked at the end of both surveys)

For questions 15 – 26, select one of the following

(1) not applicable at all

(2) rarely applicable

(3) moderately applicable

(4) largely applicable

(5) totally applicable

15. Group members felt free to criticize the ideas, statements, and/or opinions of others
16. We reached a good understanding on how we had to function
17. Group members ensured that we kept in touch with each other
18. We worked hard on the group assignment
19. I maintained contact with all other group members
20. Group members gave personal information on themselves
21. The group conducted open and lively conversations and/or discussions
22. Group members took the initiative to get in touch with others
23. Group members spontaneously started conversations with others
24. Group members asked others how the work was going
25. Group members felt that they were attacked personally when their ideas, statements and/ or opinions were criticized
26. Group members were suspicious of others

For questions 27 – 34, select one of the following

(1) very rarely or never

(2) rarely

(3) sometimes

(4) often

(5) always or very often

27. Group members grew to dislike others

28. I did the lion’s share of the work

29. Group members obstructed the progress of the work

30. Group members were unreasonable

31. Group members disagreed amongst each other

32. The group had conflicts

33. Group members gossiped about each other

34. Group members did not take others seriously

APPENDIX C. EXPERIMENT ONE INSTRUCTIONS

Below is the sheet of instructions given to students prior to deployment of experiment one.

Name ____ Group #______ Bio/Zoo 364
Identifying trees and Snapshot phenology Fall 2014

Goal. Test an app to enable future ecology students to identify and monitor any tree in the Fargo-Moorhead area for the phenology project.

How. Working in groups, each student will 'gather' and 'elaborate' 4 or more tree specimens, build a classification tree, and support tree identification to species level.

Step 1. Create an account

1. On your cell phone, tablet, or notebook, go to http://circle.cs.ndsu.nodak.edu/
2. Click on 'Play CIRCLE'.
3. Click on 'Create Account'
4. Make sure you select 'Student' as the Account Type and 'General Ecology' as the Course.
5. Complete the rest of the create account form, selecting a username and password that you will remember.
6. Click the Create Account button, making sure that the game created an account for you, then click Back to return to the main login page.
7. Login using the username and password you chose.

Step 2. Create a project for your group

1. Each group will create and contribute to a single project.
2. Have one group member click 'Create project'. Name your project and provide a brief description and keywords, then click Create Project.
3. On the next screen, invite your other group members
4. Each individual student will need to log in on their own cell phone, tablet, or notebook and accept the invitation.

Step 3. Gather specimens

1. Working individually, gather 5 – 6 tree specimens as described below.
2. One of your specimens must be the tree you are observing for the phenology project. Please tag this tree with 'phenologyproject2014'.
3. Click 'Gather specimens'. Provide a brief description of the tree and the tree's location.

Add pictures to visually describe the tree by clicking ‘Select a picture’ > ‘Choose file’. Then take a picture and click ‘Upload.’ For each additional image, you will need to ‘Add more multimedia.’ Include the following images with each specimen:

a. Close up of leaf
b. Close up of bark
c. Close up of leaf branching patterns
d. Close up of a fruit (if present)

4. Repeat this procedure for 4 – 5 additional trees.

Step 4. Elaborate on specimens

1. To unlock ‘Identify a specimen’, all specimens need elaboration and experimentation.
2. You can suggest experiments for specimens that are not your own. You can perform experiments for your own specimens. The more experiments conducted, the better the identification tree you construct.
3. Click ‘elaborate specimen’ > ‘suggest experiment or observation’.
4. Suggested observations include:
   a. What is the shape of the leaves? (e.g., round, pointed, lobed)
   b. What is the branching pattern of the leaves? (e.g., opposite, alternate, whorled)
   c. Are the leaves simple or compound?
5. First, check the list of existing experiments. If you want the person who gathered the specimen to perform one of those experiments, just select the experiment from the list and click Suggest
6. If you don’t see the experiment, click ‘new’ and complete the subsequent boxes.
7. Complete experiments suggested by others. You will need to enter descriptive text and possibly take additional pictures.

Step 5. Identify specimens

1. You need to identify each specimen, including your own. Use the online guide to assist you: http://www.arborday.org/trees/whattree/whatTree.cfm?ItemID=E6A
2. Follow the in-app instructions to either add your support for an existing hypothesis or make a new hypothesis.
3. Finally, vote for the hypothesis you agree with.

Step 6. Construct classification tree

Next, you’ll work together using the experiments you conducted to create a classification tree.

1. From the project view page, click on Construct Classification Tree
2. You are able to click on nodes (the circles) of the tree and perform actions. Start by clicking the empty node and selecting an experiment that roughly divides your specimens in half. (For example, if half of your specimens have an alternate leaf arrangement, and half have an opposite arrangement, use that as the first node)

3. Click on the Experiment circle and add "Results" to that node. (For example, click "Add Result: Alternate" to add the alternate leaf arrangement result to your tree and "Add Result: Opposite" to add the opposite leaf arrangement result to your tree. This creates a new empty node where you can add experiments or specimens.
4. Return to step 6.2 until you have a path that fully describes a particular specimen. For example, if a tree has an alternate leaf arrangement, and long needles, you would create experiments and results for each, then click on the resulting empty node and choose "Set as Specimen: X"
5. You'll need to include all of your specimens in the tree. Just add the necessary experiments and results to create the tree. You may need to return to the elaboration steps and suggest more experiments in order to create a complete tree.

Step 7. Confirm primary tree
To double check to make sure that your tree contains all of the specimens in the project and the paths are all valid, click on the Check Primary Tree button. The app will tell you if you are missing any specimens or if a specimen is using the wrong experiments.

Step 8. Documentation
1. Complete the points table, below, indicating your user name and the points you've earned.
2. Complete the snapshot phenology table using your group's methods for observing and quantifying phenology.

User name ____

Name ____ Group #______

Points earned as part of the CIRCLE testing

Activity	**Points earned**
Suggested a hypothesis	
Moderated an experiment	
Suggested an experiment	
Gathered a specimen	
Performed an experiment	
Changed a classification tree	
Got a correct hypothesis	
Gathered a piece of multimedia	
Total	

Snapshot phenology of local trees

Specimen name	**Date observed**	**% leaves dropped**	**% color change**

APPENDIX D. DEMOGRAPHIC SURVEY AND AVAILABILITY INSTRUCTIONS

This is the form given to students immediately prior to experiment two. Students in experiment one created their own accounts and were asked the same set of questions during account creation.

Demographics Survey

Enter an e-mail you can be reached, if you are selected to participate:

Age:

17 or under

18 or older

GPA

4.0-3.5

3.0-3.5

2.5-3.0

2.0-2.5

1.0-1.5

Below 1.0

Grade Level

Freshman

Sophomore

Junior

Senior

Graduate Student

Gender:

Prefer not to answer

Male

Female

Ethnicity

Prefer not to answer

Hispanic or Latino

Not Hispanic or Latino

Race

Prefer not to answer

American Indian or Alaska Native

Asian

Black or African American

Native Hawaiian or other Pacific Islander

White

Two or more racial categories

Have you used CIRCLE before?

Yes

No

Availability Chart

Place a check next to any hour long period you are available over the next two weeks

Monday

8:00-9:00 ____
9:00-10:00 ____
10:00-11:00 ____
11:00-12:00 ____
12:00-1:00 ____
1:00-2:00 ____
2:00-3:00 ____
3:00-4:00 ____
4:00-5:00 ____

5:00-6:00 ____
6:00-7:00 ____
7:00-8:00 ____

(Repeated for two weeks)

APPENDIX E. SOURCE CODE

The source code for this project is stored in a subversion (SVN) repository. The address is https://circle.cs.ndsu.nodak.edu/svn_circle-client