Washington University in St. Louis
Washington University Open Scholarship

Engineering and Applied Science Theses & Dissertations

Engineering and Applied Science

Winter 12-15-2014

The Effects of Gamifying Optional Lessons on Motivation

Aaron Zemach
Washington University in St Louis

Follow this and additional works at: http://openscholarship.wustl.edu/eng_etds

Part of the Graphics and Human Computer Interfaces Commons

Recommended Citation

Zemach, Aaron, "The Effects of Gamifying Optional Lessons on Motivation" (2014). *Engineering and Applied Science Theses & Dissertations*. 15.
http://openscholarship.wustl.edu/eng_etds/15

This Thesis is brought to you for free and open access by the Engineering and Applied Science at Washington University Open Scholarship. It has been accepted for inclusion in Engineering and Applied Science Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Computer Science and Engineering

Thesis Examination Committee:
Dr. Caitlin Kelleher, Chair
Dr. Tao Ju
Dr. Robert Pless

The Effects of Gamifying Optional Lessons on Motivation

by
Aaron Zemach

A thesis presented to the School of Engineering
of Washington University in St. Louis in partial fulfillment of the
requirements for the degree of
Master of Science

December 2014

Saint Louis, Missouri

Contents

List of Figures ... iv

List of Tables ... v

Acknowledgments ... vi

Abstract ... vii

1 Introduction ... 1

1.1 Gamification ... 1

1.2 Educational needs ... 1

2 Previous work ... 3

3 Looking Glass ... 5

4 Formative evaluation ... 7

4.1 System 1: Badges ... 8

4.1.1 Evaluation ... 10

4.1.2 Lessons learned ... 11

4.2 System 1: Achievements ... 12

4.2.1 Evaluation ... 15

4.2.2 Lessons learned ... 16

4.3 System 3: Challenges ... 17

4.3.1 Evaluation ... 22

5 Summative Evaluation ... 24

5.1 Materials ... 24

5.2 Data ... 29

5.2.1 Initial experience fulfilling ... 29

5.2.2 User intention to return ... 30

5.2.3 Observed behavior ... 31

5.2.4 Additional data ... 32

5.3 Participants ... 32

5.4 Methods ... 32

6 Results and Discussion ... 35

6.1 Differences in measures ... 35

6.1.1 Initial experience fulfilling ... 35

6.1.2 Intention to return ... 36

6.1.3 Observed behavior ... 38

6.2 Relations between measures ... 39

6.1.1 Initial evaluation vs. Intention to use ... 39

6.1.2 Intention to use vs. actual use ... 40

7 Conclusion 42
7.1 Future work 42

References 44

List of Figures

Figure 3.1: The Looking Glass Code Editor....5
Figure 4.1: The badge interface, integrated into the code editor....9
Figure 4.2: Detail of the badge interface for the “Storyteller” badge....9
Figure 4.3: Notificatino for demonstrating the skill for opening a template....10
Figure 4.4: The achievements interface, integrated into the code editor....13
Figure 4.5: Listing of a user’s earned achievements....14
Figure 4.6: Achievements available for a user to earn....15
Figure 4.7: A code snippet containing the “do together” construct....18
Figure 4.8: Puzzle interface presenting code statements in scrambled order....19
Figure 4.9: Unscrambling a code puzzle....20
Figure 4.10: Confirming an unscramble animation matches the correct animation....21
Figure 4.11: Puzzle completion success....22
Figure 5.1: The main challenges page gamification features....25
Figure 5.2: Locked challenges....27
Figure 5.3: Community leaderboard....27
Figure 5.4: Suggestions & Dares interface....28
Figure 5.5: The non-gamified interface....29
Figure 6.1: Means of ARCS scores for non-gamifed and gamified interfaces....36
Figure 6.2: Means of TAM scores for non-gamifed and gamified interfaces....37

List of Tables

Table 5.1: Looking Glass Challenge System Gamification Features 26
Table 6.1: ARCS mean differences between non-gamified and gamified interfaces 35
Table 6.2: TAM mean differences between non-gamifed and gamified interfaces 37
Table 6.3: Behavior mean differences between non-gamified and gamified interfaces 38
Table 6.4: Pearson's *r* coefficient for ARCS & TAM correlations 40
Table 6.5: Correlations between intention to use and behavioral measures 41

Acknowledgments

I would like to thank my research adviser Dr. Caitlin Kelleher for her support, guidance, and mentorship throughout this and my previous work. Her encouragement, advice, and knowledge have not only shaped my research, but ignited a passion for HCI work that has changed my professional and personal goals for the better.

My sincerest thanks for everyone on the Looking Glass team for their assistance and friendship throughout this work. I would especially like to thank Jordana Kerr, Kyle Harms, Reilly Ellis, and Noah Rowlett for their formative work on the Challenge system, as well as Dennis Cosgrove, Mike Pogran, and Michelle Ichinco for their general guidance.

Lastly, a large acknowledgement goes out to my parents for their ongoing support of my education, passions, and dreams.

Aaron Zemach

Washington University in St. Louis
December 2014

ABSTRACT OF THE THESIS

The Effects of Gamifying Optional Lessons on Motivation

by

Aaron Zemach

Master of Science in Computer Science

Washington University in St. Louis, 2014

Research Advisor: Dr. Caitlin Kelleher

Adding video-game elements to non-video-game interfaces ("gamification") has become a common engagement strategy over the past several years in the domain of education. While prior studies have found that adding game elements to mandatory educational materials can increase students' motivation to complete the materials, there has yet to be a study to investigate if game elements can make users more likely to engage with optional educational materials. In this study, we investigate whether users of a gamified educational interface are more motivated than users of a non-gamified interface to voluntarily complete educational materials. We found users of a gamified interface to spend more time using the system, as well as reporting higher intentions to return to the system, supporting gamification as a method for encouraging independent learning.

Chapter 1

Introduction

1.1 Gamification

It is often cited that 10,000 hours of deliberate practice allow an individual to develop mastery of a certain skill [10]. As Jane McGonigal points out, 10,000 hours is also the amount of time the modern young adult will have spent playing video games by his or her early twenties [24]. It is clear that video games are incredibly engaging systems, and as such, much ongoing work tries to determine how the engaging qualities of video games can be applied in non-gaming interfaces. As early as 1980, researchers investigated how "the features that make computer games captivating [can] be used to make other user interfaces interesting and enjoyable" [23].

One recently popular approach is gamification—the addition of video-game elements such as points, badges, or achievements to non-game interfaces in order to "invoke gameful experiences and further behavioral outcomes" [17]. Gamification has been the topic of a booming amount of research over the past five years. While in 2010 almost no research papers had "gamification" as a keyword, in 2013 over 180 papers were published on gamification [16][17]. Applications for gamification have been broad: helping patients through rehabilitation from injury [23]; promoting healthy lifestyle behaviors [9]; encouraging contributions to collaborative websites [12]; and increasing transactions in an online commerce service [15], to name a few.

1.2 Educational needs

At the same age that students are investing much of their time into video gaming, they are also missing out on opportunities in the classroom. While middle school has been found to be a crucial time to interest students in math and science topics [11], middle school curriculums often lack classes for specialized math and science topics, such as programming [29]. Programming environments aimed at students—such as Looking Glass, Scratch, and Alice—provide an opportunity for independent learning of these concepts. However, because learning in such programs is typically self-directed, there must be sufficient motivation for students to pursue such learning opportunities to gain the exposure in the first place. Gamification is one potential method for providing such motivation.

We investigate whether the addition of gamification elements into a novice programming environment provides additional motivation for users to learn new programming concepts. After determining the most effective way to integrate gamified elements into an open-ended programming environment, we ran an empirical study comparing a gamified interface to a non-gamified version. We found that users with the gamified interface spent significantly longer on gamified lessons, and had significantly higher intentions to return to the lessons, than those with the non-gamified version. These results support the use of gamification as a means of encouraging independent learning.

Chapter 2

Previous Work

Previous research on gamification shows a mix of positive and negative results [17]. Several studies show gamified interfaces can increase user motivation. A study of question-and-answer website Stack Overflow observed that the presence of badges increased user behavior for incentivized activities. [1] Another study found that recruitment e-mails for a research activity involving gamified elements had higher response rates than e-mails for the same activity without the gamified elements. [14] However, this study also found the quality of responses for those who used the gamified system of lower quality than those with a standard interface. Similarly, a study by Nokia found that badges in a mobile photo sharing application had the potential to increase motivation to share photos for some users, but might decrease the quality of photos shared [25]. Therefore, although it is with some cautions related to quality, prior work shows that gamification can successfully be used to motivate users towards incentivized activities.

Gamification has also previously been used to train users on how to use open-ended software systems. In one study, users learned Photoshop through a series of gamified tutorials centered on completing jigsaw puzzles using Photoshop tools [8]. Users were later observed to demonstrate the new skills that were taught as they moved into more complicated tutorials. In another study, a series of AutoCAD tutorials which had gamified elements were completed 7 times more than the same tutorials without game elements [22]. Gamification was observed to be effective in these open-ended contexts. However, in these studies, completing tutorials was mandatory. Especially in open-ended software, in which users may have a range of usage goals, it is uncertain whether a rigidly defined series of tutorials adequately supports these goals, or if the tutorials are instead tangential to the main software experience.

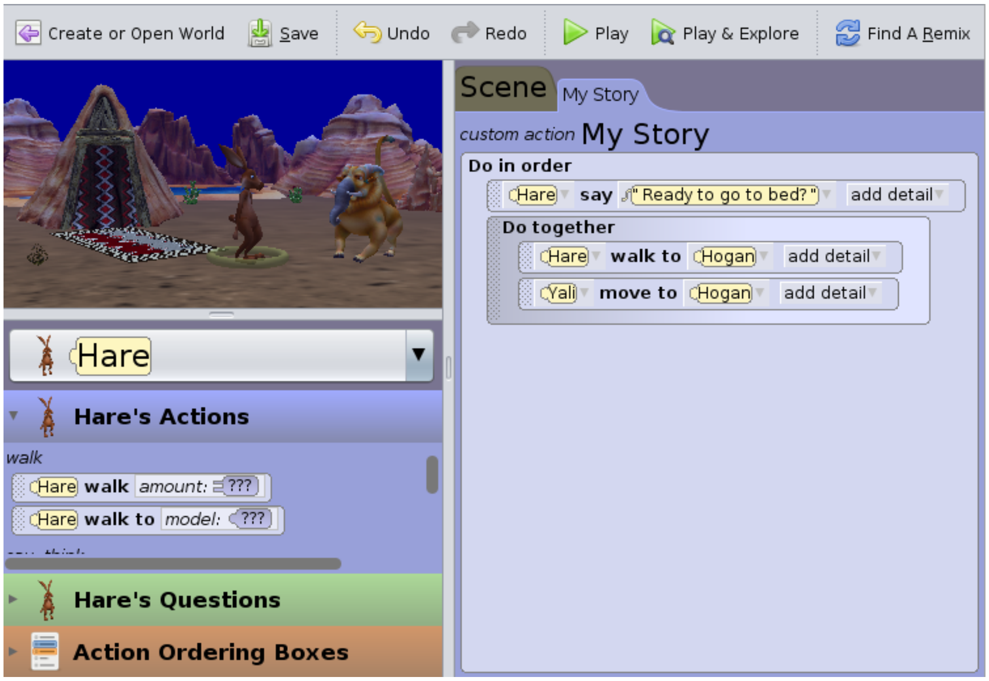
Prior work involving gamification to support education also lacks investigation of independent motivation—the gamified system is typically classroom material completed as part of a required exercise, or with some other tie to course content. For instance, in a study of more than 1000 students, adding achievements to a website that helped students study course material increased student participation with the website [7]. However, use of the website was rewarded with class credit, as well as helping students study for their midterm exam. This provided students with additional motivation that would not exist in a truly independent context. In another study, students in an algorithms class had the option to use a website to study course material, and did not receive course credit [13]. For a subset of students, inclusion of badges in the interface positively affected behaviors such as time management and carefulness of responses. Once again however, there was an external factor related to course performance that might not represent how students would use the website if it was positioned outside of a course.

To summarize, prior work has shown positive results for using gamified interfaces to motivate user interactions. However, prior work on gamification of educational interfaces has largely been focused on course-based materials, and has not looked at how gamification might be used to motivate learning outside of the classroom. As opposed to prior work in which system usage was mandatory, we investigate whether gamified elements in educational environments have motivating effects on users, when usage of the system is not required.

Chapter 3

Looking Glass

In order to investigate the effects of gamified elements on motivation to use educational software, we required educational software to gamify. As the basis of our interface, we used the Looking Glass programming environment. Looking Glass is a novice programming environment that teaches programming by allowing students to make an animated story involving 3D characters. Users drag and drop statements (called "actions"), functions (called "questions") and control flow statements (called "action ordering boxes") to animate the actions of the characters and objects in a 3D scene. Statements range from high-level storytelling actions such as "say" or "walk" to low-level concepts such as joint manipulation. Control flow statements include constructs for iteration, parallelism, and conditional blocks. Figure 3.1 shows the main Looking Glass code editor interface.



```
custom action My Story
Do in order
  Hare say "Ready to go to bed?" add detail
  Do together
    Hare walk to Hogan add detail
    Yali move to Hogan add detail
```

```
Hare walk amount: ???
Hare walk to model: ???
```

Figure 3.1: The Looking Glass Code Editor

In addition, Looking Glass is tied to a community where users can share their animations (called "Worlds"), 3D scene setups without any actions (called "Templates"), and short actions for others to incorporate into their own animations (called "Snippets"). Each user on the community has a profile which showcases the content that he or she has created.

We found Looking Glass to be appropriate in our investigation for several reasons. First, Looking Glass is aimed at middle school students, the crucial age to engage students in science and mathematics topics. Additionally, Looking Glass is intended to be used outside of the classroom, making it ideal for investigating independent motivation. Lastly, although Looking Glass supports powerful programming concepts such as nested loops, custom procedures, and recursion, users do not always discover these abilities on their own. Gamification provides one potential strategy to teach these advanced skills to users. By incentivizing the learning of these skills, gamified elements have the potential to engage users with new and unfamiliar programming concepts. Such a gamified interface could help students independently learn programming skills which are not being taught in school, if students voluntarily choose to engage with the interface.

Chapter 4

Formative evaluation

We wanted to add gamification to Looking Glass as a means of guiding users from novice to experienced programmers. We developed a list of skills that we felt were important to learning Looking Glass, divided into three categories:

- **IDE skills:** Skills required to use the Looking Glass IDE, such as sharing a final animation to the online community
- **Programming skills:** The mastery of various programming concepts, such as using control loops correctly or building a custom function
- **Animation skills:** Understanding techniques to make advanced animations, such as manipulating a character's joints

Our goal was to use gamification design features to motivate users to learn unfamiliar skills, and thus gain experience with using Looking Glass, programming, and making 3D animations respectively—though it should be noted that the IDE skills and animation skills are important mainly as prerequisites for more advanced programming skills.

At the same time, the open-ended storytelling aspect is a crucial part of the Looking Glass experience. Users may have different goals and desired behaviors when using Looking Glass, and we did not want the system to affect or prevent those goals from occurring. This contrasts with the previously discussed work on gamifying introductions for open-ended interfaces. The prior work typically gamified a defined series of tutorials. Instead, we believed that it is important that a gamified interface within an open-ended system supports multiple user goals, where users' behaviors are not limited by the system.

4.1 System 1: Badges

We began with several hypotheses for how gamified elements might best be integrated into an open-ended programming environment. From each of these hypotheses, we were able to make design decisions about the gamified system.

1. If gamified elements co-exist with code editing, users will be able to form strong connections about the skill being taught and its purpose in Looking Glass.

Our system awarded "badges" to users as they programmed their own stories. The skills required for badges can take place during the course of regular editing. Our hope was that this would guide users without diminishing their autonomy or creativity, and better emphasize the importance of the skills. We felt this integrated gamified elements nicely with storytelling goals. The integration of a badge information pane with the code editor is illustrated in Figure 4.1.

2. Rewarding users as they work will be fun and exciting, enticing users to earn additional rewards.

As users demonstrate the skills that badges require, a notification popup fills the screen with praise for the user. Once the user clicks, the notification goes away and he or she can return to code editing. We hoped this popup would capture the user's attention and excite the user for what he or she had just accomplished. This popup is illustrated in Figure 4.3.

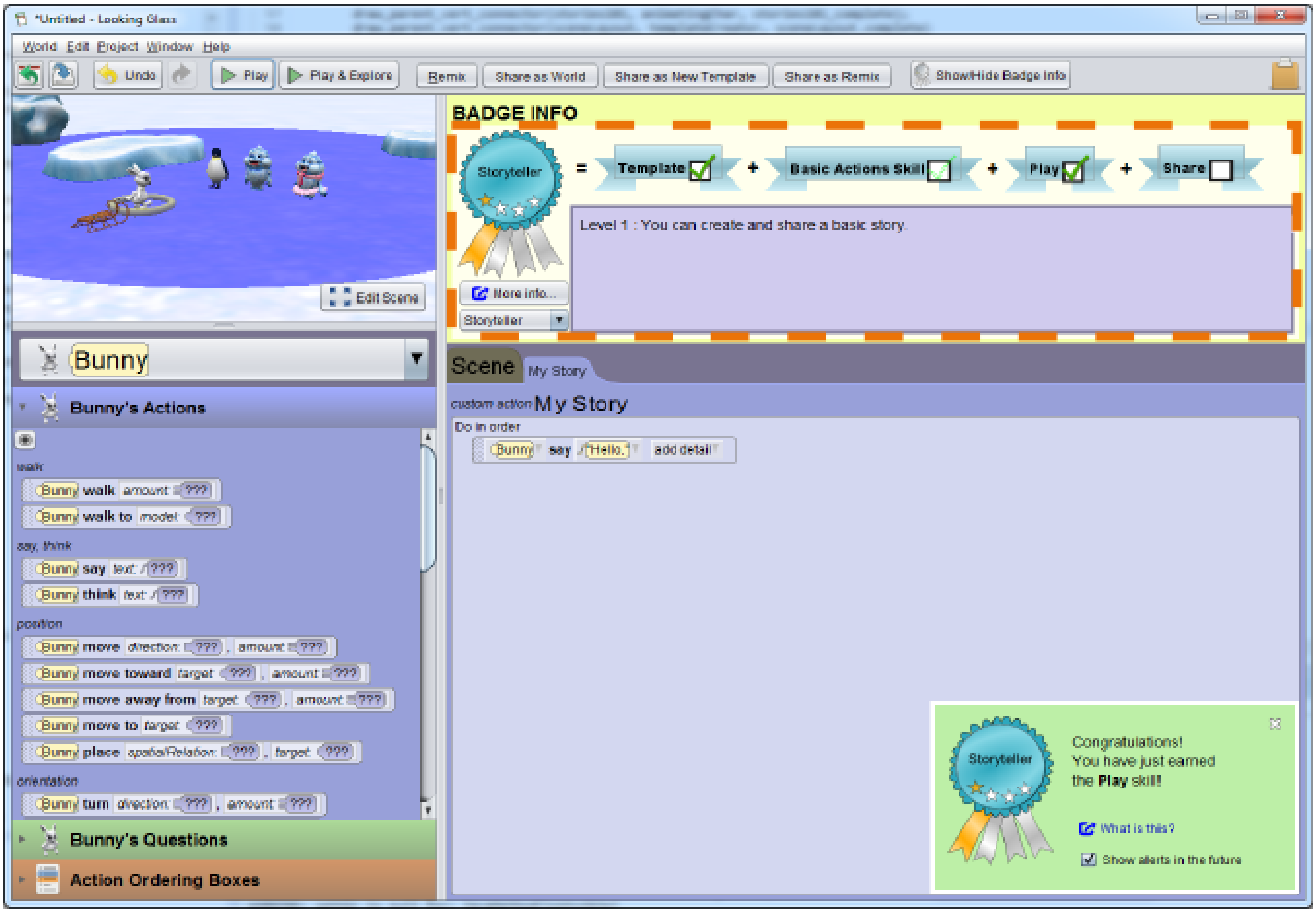
3. If each reward has multiple steps, users who have completed part of a reward will work harder to finish it, encouraging the learning of more skills.

Each badge required two or more component skills to be earned. We hoped that this would provide a "foot in the door," so users who had completed some of a badge's requirements would have additional motivation to complete the rest of the requirements. Our interface illustrating these multiple steps is shown in Figure 4.2.

4. If gamified elements capture users' attentions right away, they will be intrigued and want to return to the gamified system as they continue using Looking Glass.

The first badge available, “Storyteller,” awarded the first skill to the user immediately upon entering the IDE (a skill for starting a new world). We hoped this would inform users of the badge system’s existence, and intrigue them about the rest of a badge’s steps.

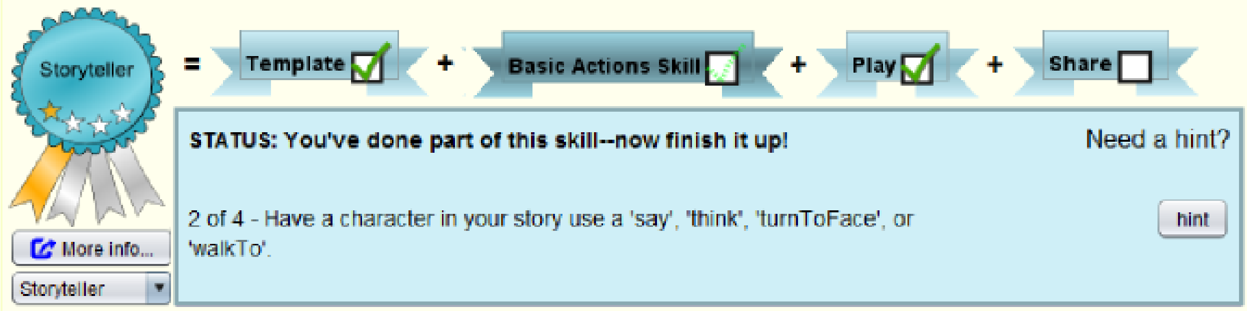
The other three requirements for the “Storyteller” badge (adding 2 procedures to their program from a set of 4 choices, playing their animation, and sharing their animation to the online community) were common Looking Glass activities that we hoped would guide users gently through their first Looking Glass experience, while still allowing for creative storytelling. At first, the user can only work on “Storyteller,” but as badges are earned, new badges become available for the user to work on, enforcing a progression from simple to complex skills.



Level 1 : You can create and share a basic story.

Congratulations! You have just earned the **Play** skill!

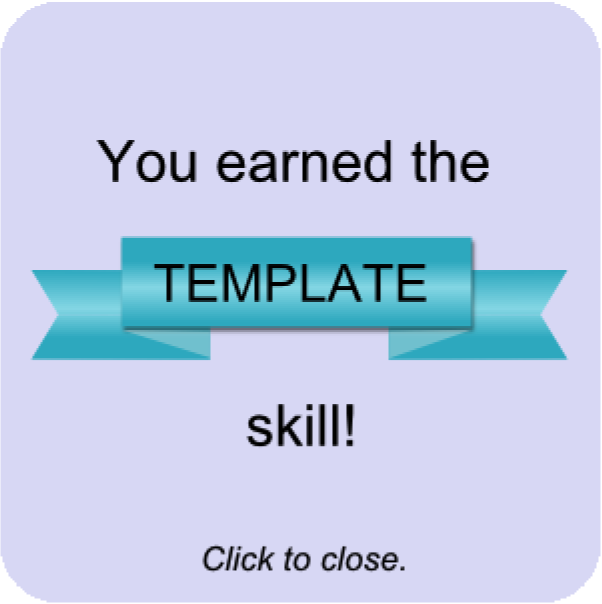
Figure 4.1: The badge interface, integrated into the code editor



STATUS: You've done part of this skill--now finish it up!

2 of 4 - Have a character in your story use a 'say', 'think', 'turnToFace', or 'walkTo'.

Figure 4.2: Detail of the badge interface for the "Storyteller" badge



You earned the TEMPLATE skill!

Click to close.

Figure 4.3: Notification for demonstrating the skill for opening a template

4.1.1 Evaluation

We tested the badge system with 10 users (6 male, 4 female) between the ages of 10 and 15 at the Saint Louis Science Center. Participants used Looking Glass with the badge system for 15-25 minutes and received a $5 Amazon gift card for their participation. Participants were not given any information about the badge system initially, as we were interested to see if participants would try to earn a badge on their own. Participants were asked various questions about the system as they worked, and eventually were asked to complete specific tasks related to using the system if the participant had not performed the tasks independently.

Most of the users were able to understand that a badge was something that they could earn for accomplishing certain actions. Some users understood that the component skills were the steps towards earning a badge, but others would make up their own explanations for how a badge was earned:

> *"I need to get the chickens to the mountain." (A4)*
> *"You get a badge when you make a full story" (A7)*

Users typically completed component skills incidentally, and they would sometimes be confused when the popup announcing their completion appeared, and dismiss it. A few users said that they found the popup "frightening." Most significantly, every user worked towards the previously described "Storyteller" badge, but not a single participant chose to complete the final requirement for the badge, sharing their animation to the online community, without being specifically asked to do so. This was true even if the user knew that he or she needed to share to earn the full badge, indicating a lack of interest in earning the badge. In general, the badge system mainly distracted the users from their primary interest, storytelling.

4.1.2 Lessons learned

From this evaluation, there were multiple lessons learned:

1. Requiring several disparate steps to earn a badge is confusing.

Although many users could correctly explain the relationship between a skill and a badge, several participants were confused that badges could only be earned indirectly through completing multiple tasks. This connection was so unclear that multiple participants invented their own reasoning for how a badge was earned. Additionally, we did not observe that having already completed some of the requirements for a badge provided any sort of "foot-in-the-door" motivation to complete the remainder of the requirements.

2. Disabling the whole screen is too disruptive.

The large notification popup was designed to make sure users could not ignore the fact that a badge requirement had been completed. However, since users in general did not intend to complete a requirement when taking an action, the sudden popup was overwhelming and confusing. After dismissing the first popup, participants would generally dismiss future popups immediately, without reading them to see which requirement was fulfilled.

3. Awarding the first requirement immediately devalued the importance of badges.

We had all users begin their sessions by opening a 3D scene template, which immediately triggers a notification popup in the code editor. However, since the entire editor was novel to the participant, the information of the popup (for a badge system that they did not even know existed) proved irrelevant, and the user immediately closed the popup so that story editing could begin. After this, the badge system was often ignored entirely, as it was viewed as extraneous to the main purpose of Looking Glass.

4.2 System 2: Achievements

The lack of motivation to earn badges and confusion over how to earn badges indicated that the badge system required a redesign. Based on what we learned from the badge system, we based our second system off of new hypotheses, and formed designs around them in the following ways:

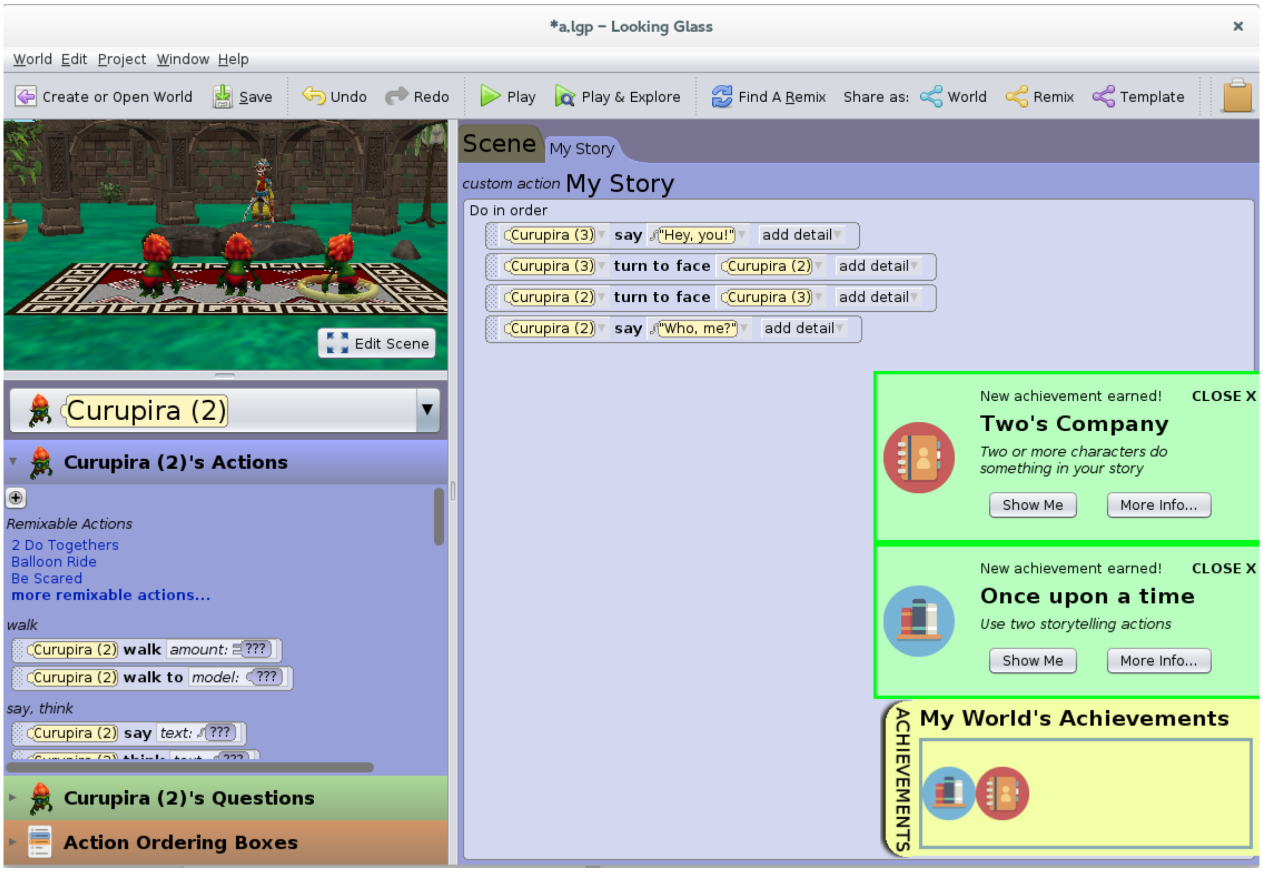
1. Having one skill per reward will help users better understand how the reward is earned.
The paradigm of completing multiple steps to earn a single reward was removed. Instead, each skill was branded as an individual "achievement" that a user could earn. For example, the previous requirements for the Storyteller badge were broken into separate achievements: one for using 2 of 4 specified procedures, one for playing the animation, and one for sharing the animation to the community. We hoped this would help users better understand how the achievement was earned, better motivating them to earn additional achievements.

2. For notifications to be effective, they must be attention grabbing without being distracting.
Instead of a constantly present information panel, a small overlay at the bottom of the code editor showed images for the achievements a user had earned so far during their editing session. When the user displayed a new skill, a small notification would enter from the side of the screen and sit above

this box until the user chose to dismiss it (Figure 4.4). Clicking on the notification or overlay below it would open up a panel that listed all the achievements a user could complete, including how to complete them (Figure 4.6), as well as a tab listing the achievements a user had already completed (Figure 4.5). We hoped this new system would be less distracting while the user was editing code, while still providing an exciting notification when a skill was earned.

3. The best time to introduce users to the reward system is not immediately, but instead after they already have started to grasp the main underlying system.

We removed the skill for opening a scene template. Now, users could only earn their first achievement after having added a few procedures to their code. We hoped that this would allow users time to acclimate to the main Looking Glass system before the achievement system was introduced, helping them better understand what the achievement system was and how it integrated with the main Looking Glass experience.



custom action My Story

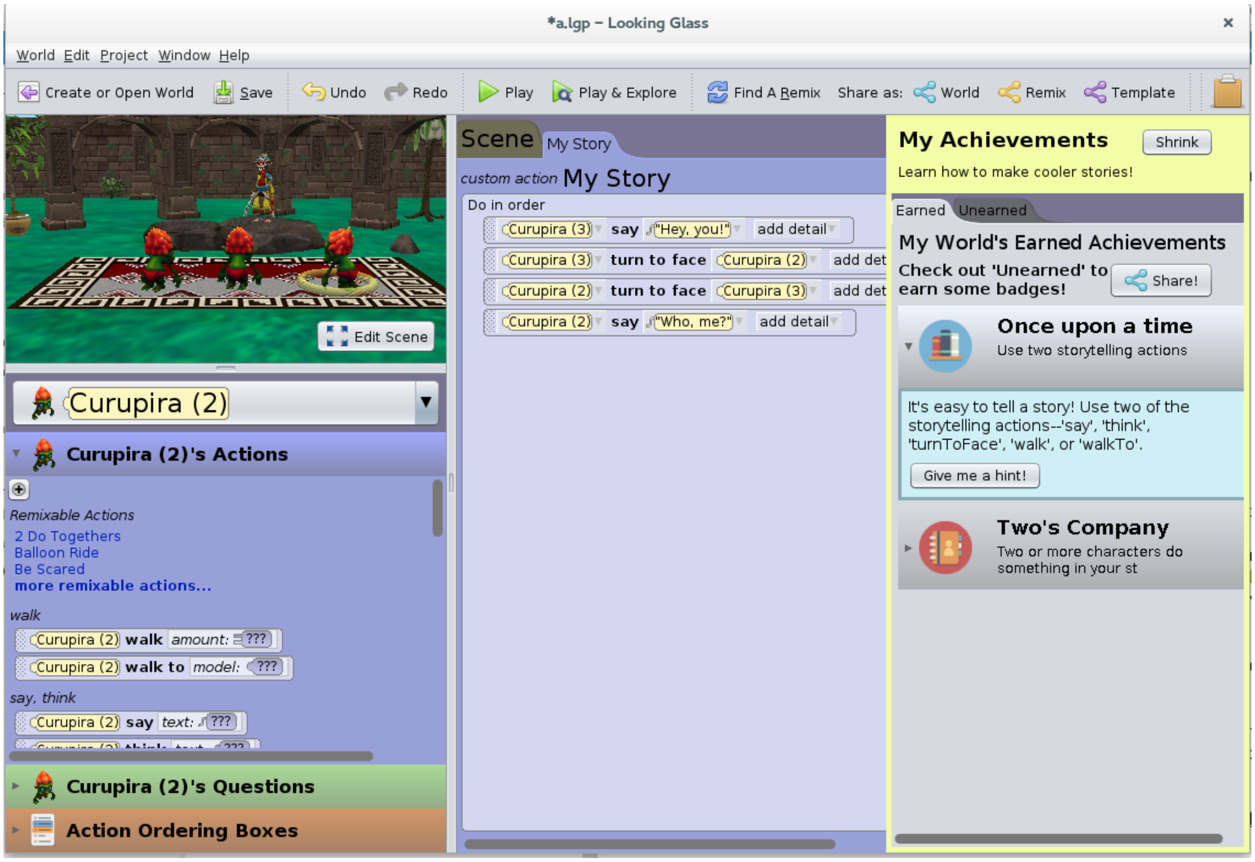
```
Do in order
    Curupira (3) say "Hey, you!" add detail
    Curupira (3) turn to face Curupira (2) add detail
    Curupira (2) turn to face Curupira (3) add detail
    Curupira (2) say "Who, me?" add detail
```

```
Curupira (2) walk amount: ???
Curupira (2) walk to model: ???
Curupira (2) say text: ???
```

New achievement earned! CLOSE X
Two's Company
Two or more characters do something in your story
Show Me More Info...

New achievement earned! CLOSE X
Once upon a time
Use two storytelling actions
Show Me More Info...

Figure 4.4: The achievements interface, integrated into the code editor



Curupira (2) walk amount: ???

Curupira (2) walk to model: ???

Curupira (2) say text: ???

Curupira (3) say "Hey, you!" add detail

Curupira (3) turn to face Curupira (2) add det

Curupira (2) turn to face Curupira (3) add det

Curupira (2) say "Who, me?" add detail

Check out 'Unearned' to earn some badges!

Once upon a time

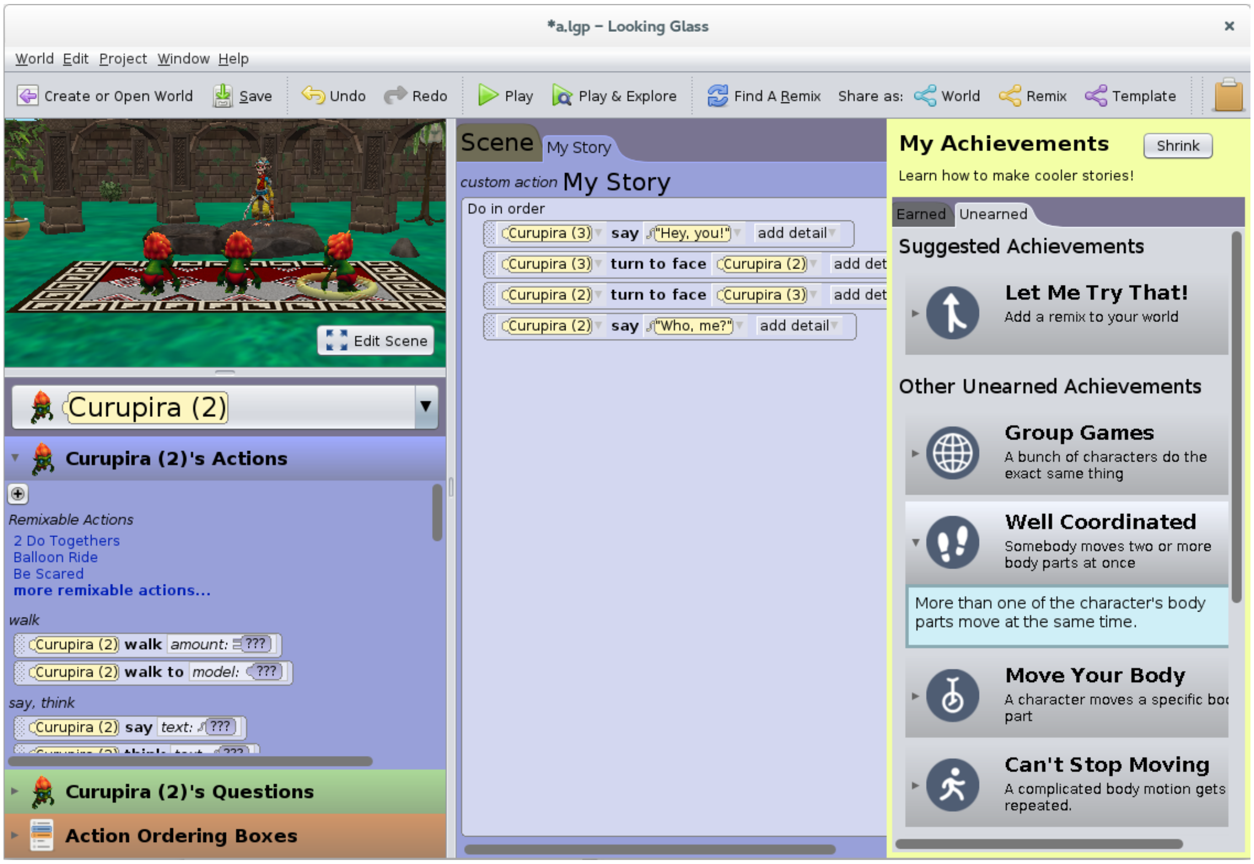
Use two storytelling actions

It's easy to tell a story! Use two of the storytelling actions--'say', 'think', 'turnToFace', 'walk', or 'walkTo'.

Two's Company

Two or more characters do something in your st

Figure 4.5: Listing of a user's earned achievements



Curupira (3) say "Hey, you!" add detail
Curupira (3) turn to face Curupira (2) add det
Curupira (2) turn to face Curupira (3) add det
Curupira (2) say "Who, me?" add detail

Curupira (2) walk amount: ???
Curupira (2) walk to model: ???
Curupira (2) say text: ???

Learn how to make cooler stories!

Suggested Achievements

Let Me Try That! Add a remix to your world

Other Unearned Achievements

Group Games A bunch of characters do the exact same thing

Well Coordinated Somebody moves two or more body parts at once

More than one of the character's body parts move at the same time.

Move Your Body A character moves a specific bod part

Can't Stop Moving A complicated body motion gets repeated.

Figure 4.6: Achievements available for a user to earn

We hoped that, as seen with badges, users would earn their first achievement incidentally during regular editing. Even if they did not notice the notification box initially, eventually they would be intrigued and investigate the system further. Upon noticing that they had already earned one or more achievements, the user would explore what additional achievements could be earned, and then begin trying to learn more skills. Since the achievements could be earned in any order, the user would be able to choose the skills that were most relevant to him or her, providing few limitations on the storytelling experience.

4.2.1 Evaluation

We tested the achievement system with 9 users (6 male, 3 female) between the ages of 10 and 14 at the Saint Louis Science Center. Once again, participants used Looking Glass for 15-25 minutes and received a $5 Amazon gift card for their participation. Participants were observed using Looking

Glass with the achievement system, asked questions about the system, and sometimes asked to perform a task such as finding an achievement that looked interesting and earning it.

As expected, all participants earned at least one achievement through their regular code editing. All participants understood that achievements were awarded for actions they took while editing their animations, and generally were able to articulate a description of how to earn a specific achievement in their own words. Occasionally, participants would still misattribute why they earned an achievement, especially if they did not initially notice the notification when it first appeared. For example, when one user was asked how he earned two achievements, one for using 2 of the 4 specified actions, and another for using a control flow loop for parallel execution, he explained, *"I started my story, and every character in my story does something" (B9)*. It seems likely he was cuing off the creative titles attributed to these two achievements ("Once Upon a Time" and "All Together Now", respectively) instead of their actual content.

When notifications appeared, users generally stopped within a few seconds to observe them, but often returned to editing their animations without investigating further. In addition, similarly to the previous system, users only earned achievements incidentally—not a single participant navigated to the achievement info panel, read the description for an achievement, and then took the actions to earn it, without being prompted to do so. Even for users who were prompted to earn a specific achievement, users were mainly interested in leaving the achievement panel and returning to story editing. The achievement popups were mainly viewed as interruptions to code editing, as opposed to rewards.

4.2.2 Lessons learned

There were several improvements with the achievement system from the badge system. Participants generally were better at explaining how to earn an achievement, although there was still some confusion. The response to notifications for completing a skill was greatly improved, with participants often choosing to pause their code editing to observe and read the notifications, which minimized the interruption of their work. However, there were still some areas where the system failed:

1. Giving rewards for actions that could happen incidentally is poorly suited for teaching skills.

When users begin earning rewards for incidental actions, the cause and effect relationship of skills and achievements is broken. Users do not choose to seek out new skills to earn achievements. For those achievements that are earned just through typical editing, the educational aspect that is central to teaching skills is lost.

2. Incidental rewards are not adequately tied to the skill they represent.

The system again would sometimes fail to adequately communicate the skill associated with each the achievements. It is necessary for this association to be emphasized if users are expected to figure out how to actualize a skill with their code.

4.3 System 3: Challenges

Based on these observations, it seemed that incidental rewards were a bad way to motivate exploration of new programming skills. We determined that a reward system would need to:

1. Involve a purposeful decision to achieve a reward
2. The reward must be closely tied to its skill, that is, the reason why it was earned.

Based on these goals, we looked at other work that was being done on adding code scramble puzzles to Looking Glass. For a puzzle, a short animation is played to a user. Then, the user is presented the code statements that comprise that animation in a scrambled order, and attempt to rearrange them to match the correct animation.

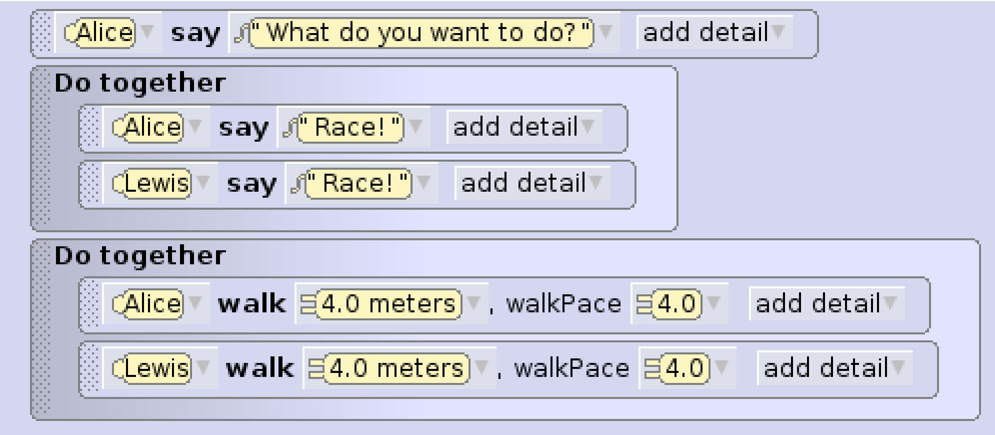
Using the puzzle system as the basis for gamified lessons seemed promising based on our new goals.

- Puzzles are separate from the code editing interface, so rewarding puzzle completion would not be incidental during editing. Users must instead make a purposeful decision to begin a puzzle, which we hope would eliminate the confusion of incidental rewards.
- Each puzzle could be tied to one specific skill, to better emphasize what programming knowledge a user was gaining. We hoped this would prevent against users in formative testing who misattributed why a specific reward was earned.

We called gamified puzzles “challenges.” We felt this system tied back to our original goals well—by completing challenges for successively harder skills, users can be taken from Looking Glass novices to experienced users. At the same time, since the challenges are a separate activity from code editing, completing a challenges would not degrade the main Looking Glass storytelling experience. Lastly, puzzles can be customized by choosing the code which is scrambled, allowing lessons to be better customized to a user’s goals than rigid tutorials.

For the challenge system, a database of short animations made in Looking Glass (“snippets”) is queried with scripts describing each skill we would like users to learn. A snippet that displays a given skill is located, and it is presented to the user as a scramble puzzle. Upon completing the puzzle, the user has finished the challenge and earns a badge.

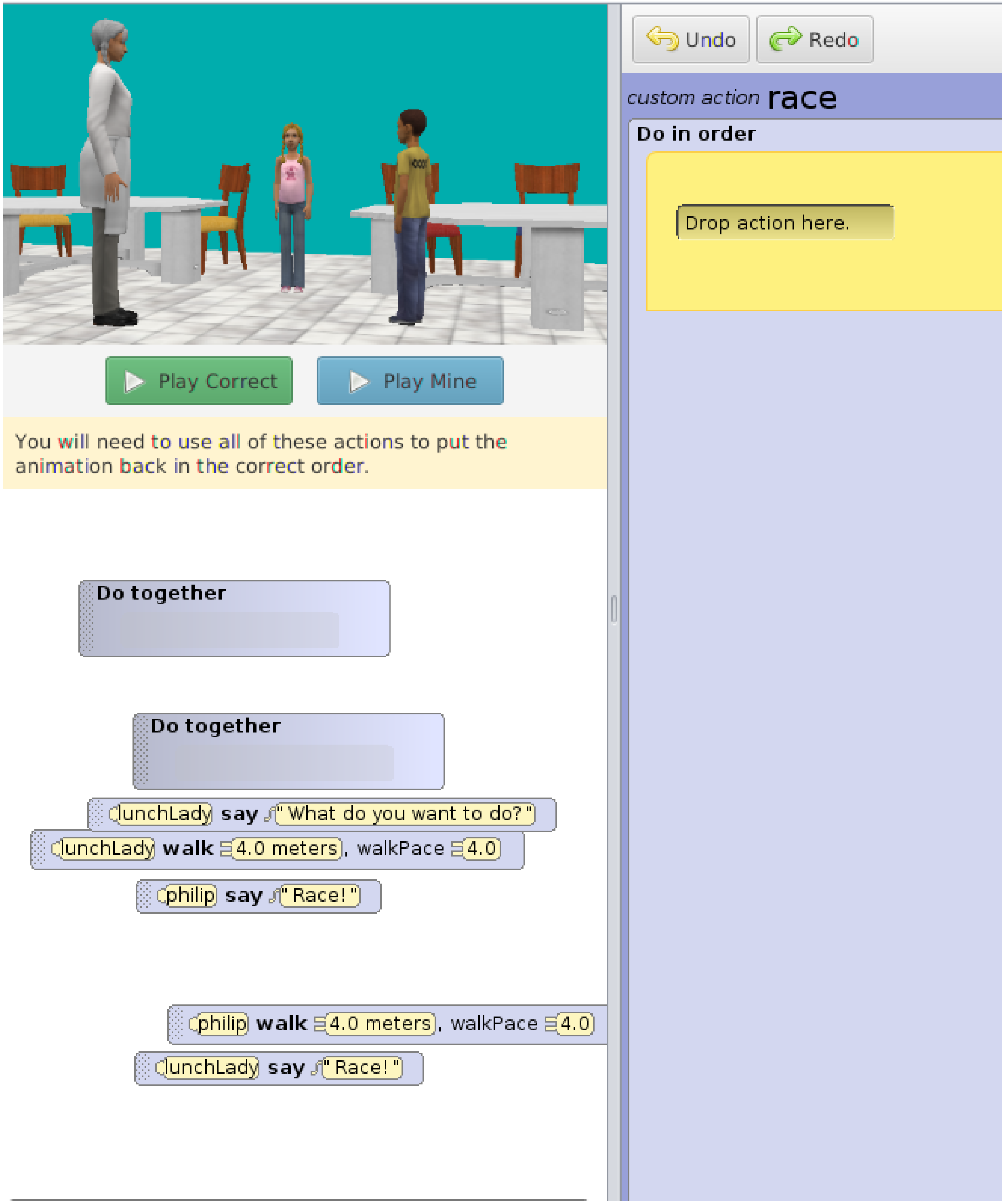
For example, for the skill of using the “do together” parallelism construct, the system might locate the snippet shown in Figure 4.7



```
Alice say " What do you want to do? "
Do together
    Alice say " Race! "
    Lewis say " Race! "
Do together
    Alice walk 4.0 meters , walkPace 4.0
    Lewis walk 4.0 meters , walkPace 4.0
```

Figure 4.7: A code snippet containing the “do together” construct

The puzzle system would then scramble the order of the statements, as shown in Figure 4.8.

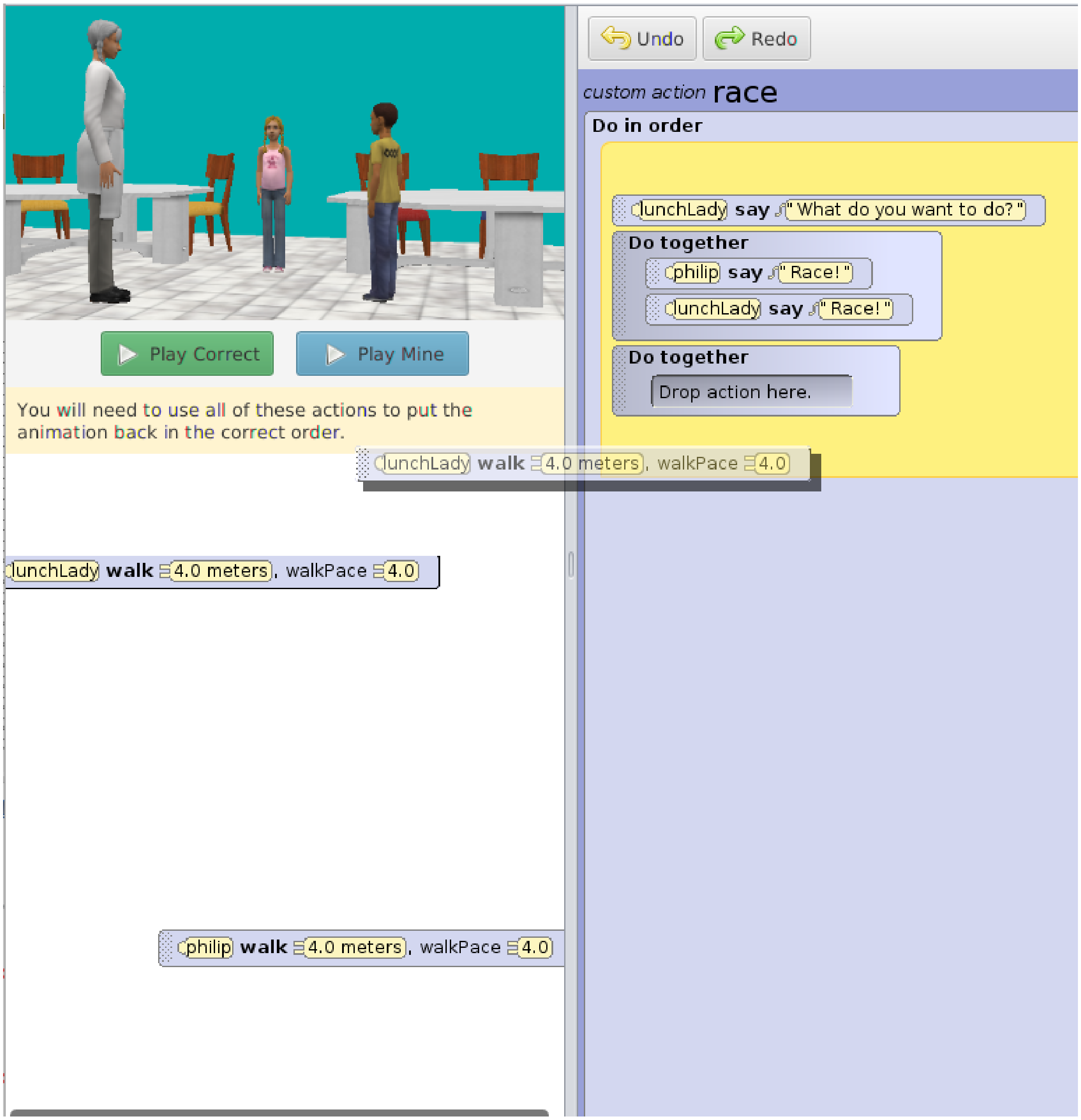


You will need to use all of these actions to put the animation back in the correct order.

```
Do together

Do together

lunchLady say "What do you want to do?"
lunchLady walk 4.0 meters, walkPace 4.0
philip say "Race!"

philip walk 4.0 meters, walkPace 4.0
lunchLady say "Race!"
```

Figure 4.8: Puzzle interface presenting code statements in scrambled order

The user tries to rearrange the statements into the proper order (Figure 4.9) and can play his or her animation and the correct animation at any time to see how close the arranged code is to the correct code (Figure 4.10).



You will need to use all of these actions to put the animation back in the correct order.

```
custom action race
Do in order
    lunchLady say "What do you want to do?"
    Do together
        philip say "Race!"
        lunchLady say "Race!"
    Do together
        Drop action here.
```

```
lunchLady walk 4.0 meters, walkPace 4.0
lunchLady walk 4.0 meters, walkPace 4.0
philip walk 4.0 meters, walkPace 4.0
```

Figure 4.9: Unscrambling a code puzzle

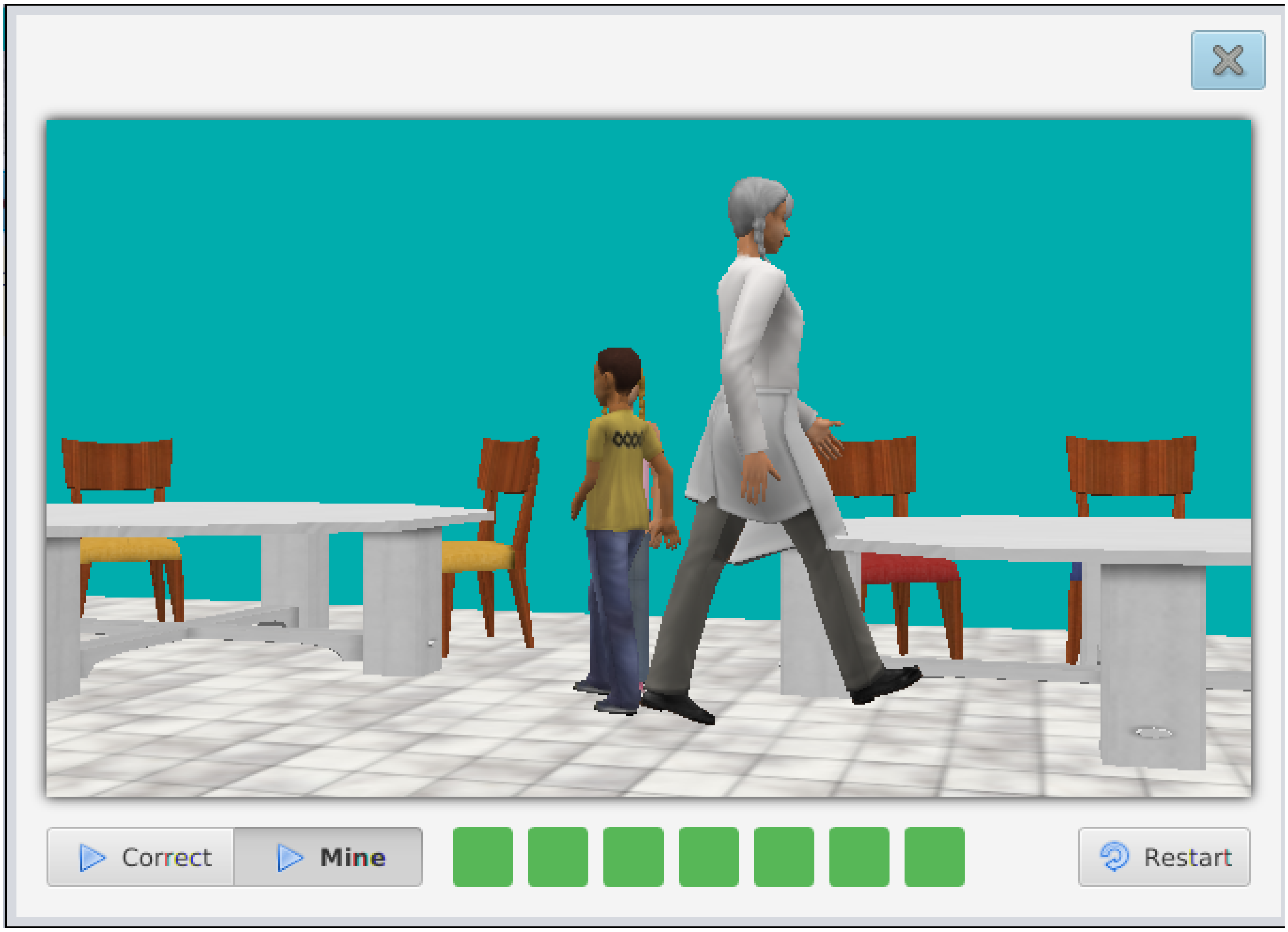
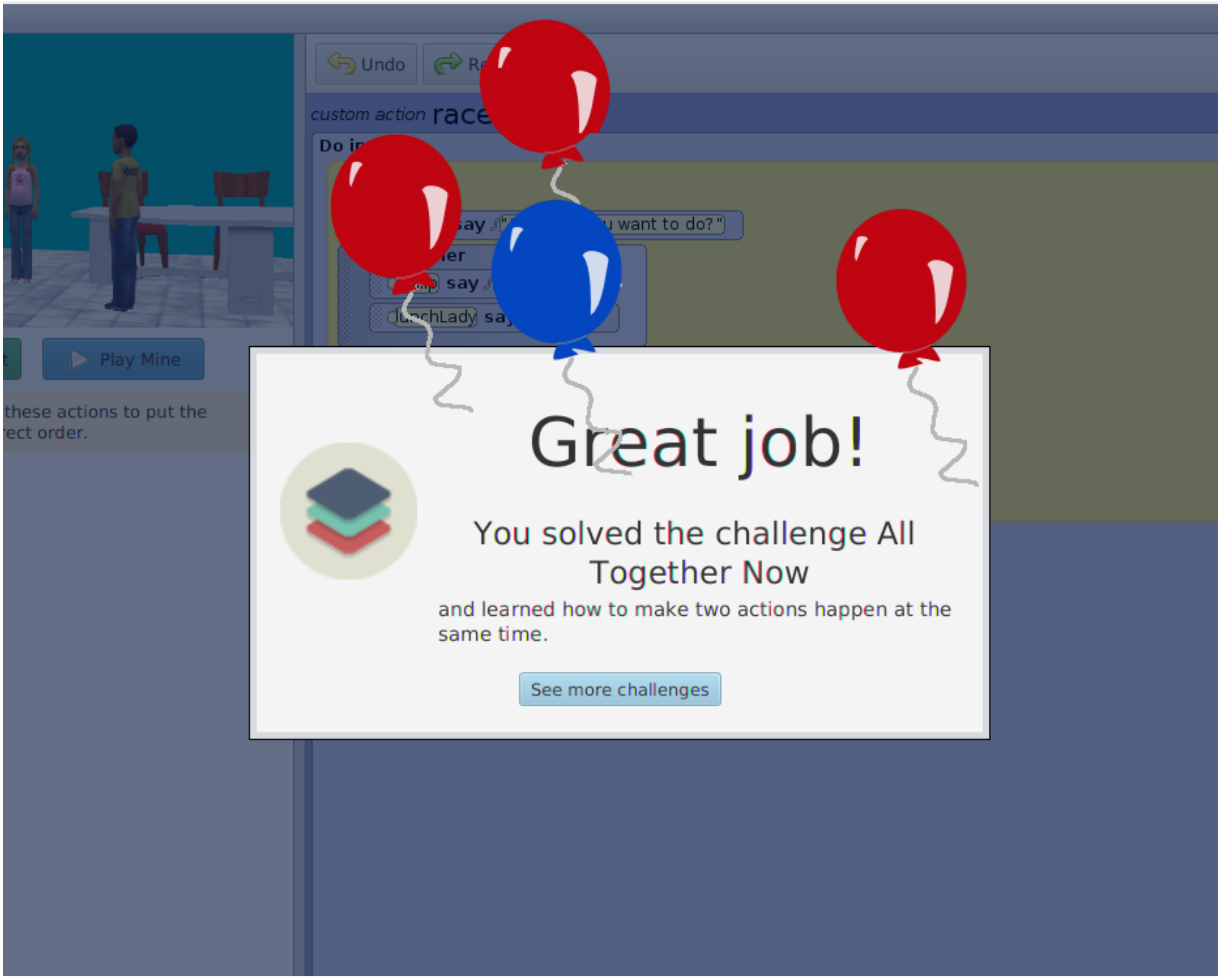


Figure 4.10: Confirming an unscrambled animation matches the correct animation

Once the code matches with the correct code, the system informs the user that the challenge has been completed, shown in Figure 4.11.



Great job!

You solved the challenge All Together Now

and learned how to make two actions happen at the same time.

Figure 4.11: Puzzle completion success

We hoped that users would easily understand the challenge system based on minimal visual guides within the interface. We expected users to be interested in completing the puzzles, and that when they returned to the main challenges interface after the first puzzle was completed, the rewards they had received from completion would motivate them to pursue additional challenges.

4.3.1 Evaluation

We tested the challenge system with 8 users (5 male, 3 female) between the ages of 10 and 15 at the Saint Louis Science Center. Once again, participants used Looking Glass for 15-25 minutes and received a $5 Amazon gift card for their participation. Participants were asked to solve one or two challenges without receiving instructions, responded to verbal questions while solving the challenges, and then were allowed to edit in Looking Glass with the remainder of their time.

All of the users who tested the system were able to locate a challenge they were interested in, begin the challenge, and attempt to unscramble the code statements without any instructions. Additionally, users recognized that by solving challenges, they were learning how to use the program, such as one participant who said that by solving challenges, "You learn to use the program better" (C5) or another who said that solving a challenge "Helps you learn how to do stuff" (C3).

Although no users chose to return to the challenges selection screen during their free editing time, several named gamified aspects that would entice them to return to challenges if they had a longer time to use the system. For instance, one participant singled out a leaderboard tracking how many challenges members of the Looking Glass community had completed, saying that if he had Looking Glass on his personal computer, "I would sit down and to all of [the challenges] to be on top" (C6).

Based on this formative evaluation, we determined that the challenge system fulfilled our goals of having a system that could teach users new programming and animation skills, and would motivate users to engage with the system through gamified interface elements.

Chapter 5

Summative Evaluation

Though the gamified challenge system showed high usability and promising motivational qualities during formative testing, the limited length of the user tests (no more than 25 minutes) left our main question unanswered: Are users of a gamified interface more motivated than users of a non-gamified interface? If the gamified interface does motivate users to choose to complete optional challenges, then gamification would appear to be a successful strategy to enable independent learning. We designed a summative evaluation to investigate these questions.

5.1 Materials

In order to evaluate our research question, we needed both a gamified and non-gamified version of the challenge system. For the gamified version, we wanted to include multiple gamification features that might provide additional motivation for users to return to challenges. To design these features, we referred work by Hsu et al. who perform a literature review to identify 3 categories of gamification components, each with a set of design factors [19]:

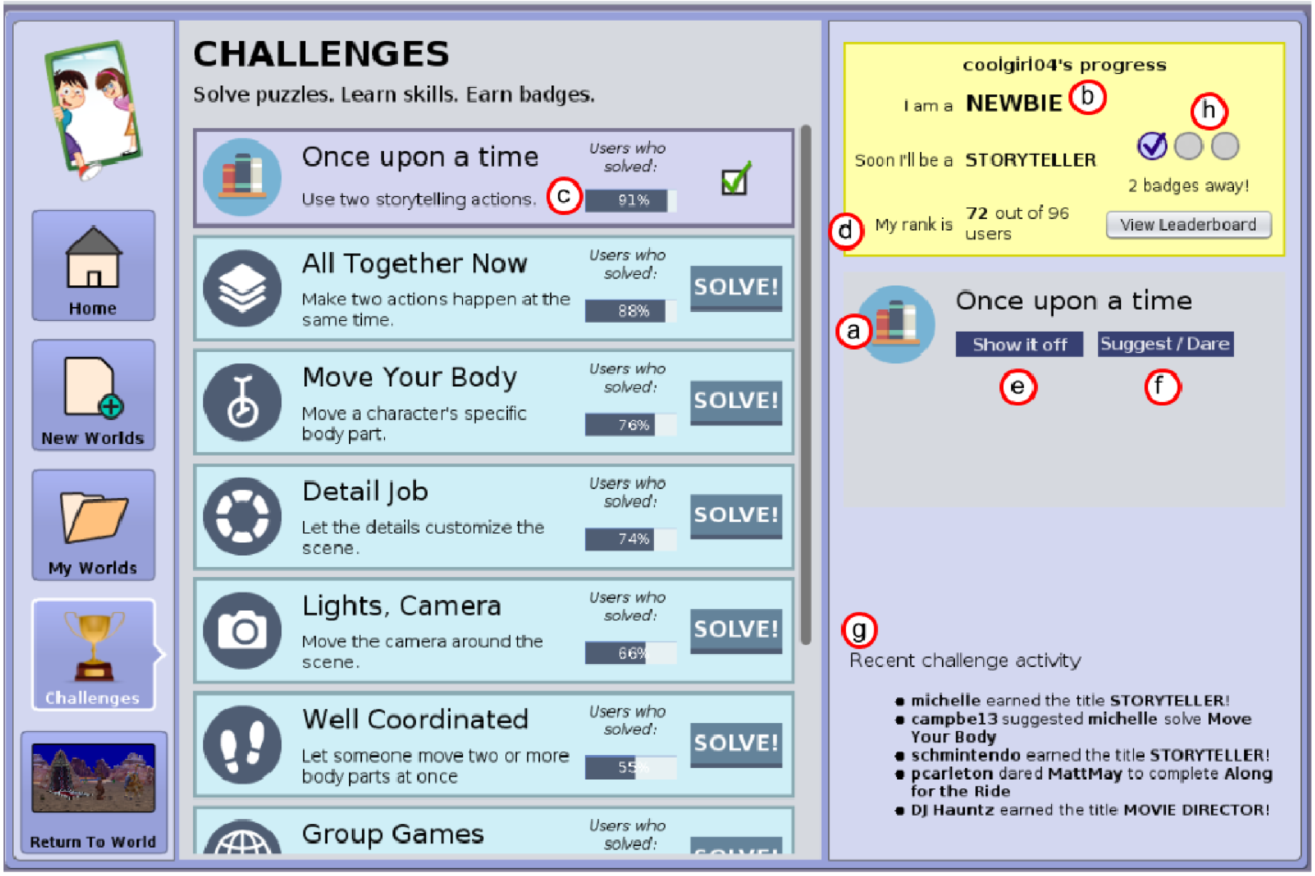
- **Achievement**
    - *Rewards*: Satisfy and motivate users
    - *Goal setting*: Provide interest and enjoyment in chasing goals
    - *Reputation*: Provide an estimation of recognition from other users
    - *Status*: Feed a need for recognition, fame, and attention from other users

  Interpersonal relationship
    - *Instruction*: Help users master the system in an efficient way
    - *Competition:* Enable users to compete with other users
    - *Altruism*: Allow users to bridge and maintain relationships

  Role playing
    - *Group identification*: Communicate a shared set of activities that bind users together [2].
    - *Self-expression*: Express users' autonomy and originality

- *Time pressure*: Impose time limits for certain behaviors

We incorporated design features related to each of the 9 related factors into the experimental challenges interface. The features included are shown in Table 2.1 and illustrated in Figures 5.1 – 5.4. The process of unscrambling a puzzle is identical between the control and experimental versions.



CHALLENGES

Solve puzzles. Learn skills. Earn badges.

Once upon a time — Use two storytelling actions. — Users who solved: 91%

All Together Now — Make two actions happen at the same time. — Users who solved: 88%

Move Your Body — Move a character's specific body part. — Users who solved: 76%

Detail Job — Let the details customize the scene. — Users who solved: 74%

Lights, Camera — Move the camera around the scene. — Users who solved: 66%

Well Coordinated — Let someone move two or more body parts at once — Users who solved: 55%

Group Games — Users who solved:

coolgirl04's progress

I am a NEWBIE

Soon I'll be a STORYTELLER

2 badges away!

My rank is 72 out of 96 users

Once upon a time

Recent challenge activity

- **michelle** earned the title **STORYTELLER!**
- **campbe13** suggested **michelle** solve **Move Your Body**
- **schmintendo** earned the title **STORYTELLER!**
- **pcarleton** dared **MattMay** to complete **Along for the Ride**
- **DJ Hauntz** earned the title **MOVIE DIRECTOR!**

Figure 5.1: The Main Challenges page gamification features: a) Badge, b) Title, c) Completion percentage, d) Community leaderboard ranking, e) Badge showcasing, f) Suggestions/Dares, g) Community news feed, h) Progress towards title

Table 5.1 Looking Glass Challenge System Gamification Features

Feature	**Description**	**Factor(s) engaged**
Badges	Earnables for completing a puzzle with an associated image & name	Rewards
Titles	Special designations for the user, such as labeling the user as a “Storyteller” or “Movie Director”	Rewards Reputation Status Group ID Self-expression
Title progress	Checked and unchecked circles indicate how many challenges must be completed to earn the next new title	Goal setting
Written praise	A congratulatory message given after completing a puzzle	Rewards
Visual effects	After completing a puzzle, balloons fly through the screen	Rewards
Locked puzzles	More advanced puzzles start off locked before prerequisite puzzles have been solved	Goal setting
Completion percentages	Each puzzle informs what percentage of users in the community have completed the puzzle	Competition
Community leaderboard	Ranking of all community users by how many puzzles they have completed, listing everyone’s username, the number of puzzles they have completed, and their current title	Reputation Competition
Community news feed	A list of challenge-related events for all community members (e.g. “michelle earned the title STORYTELLER”)	Status
Communicating skill learned	Each puzzle begins & ends by reminding the user what programming/Looking Glass skill they are learning (e.g. “You completed the puzzle and learned how to get two actions to happen at the same time”)	Instruction
Dares	A user can dare another user to complete a certain puzzle, which appears in the community news feed & a special notification for the dared user	Competition
Suggestions	A user can suggest another user complete a certain puzzle, presumably as a suggestion to learn a given skill	Altruism
Badge showcasing	A user can choose to showcase a certain badge on his or her community profile	Self-expression
Limiting sessions	After 5 minutes working a challenge, the interface suggests the user takes a break from working on a specific puzzle; after 7.5 minutes, the interface requires the user to take a break	Time pressure

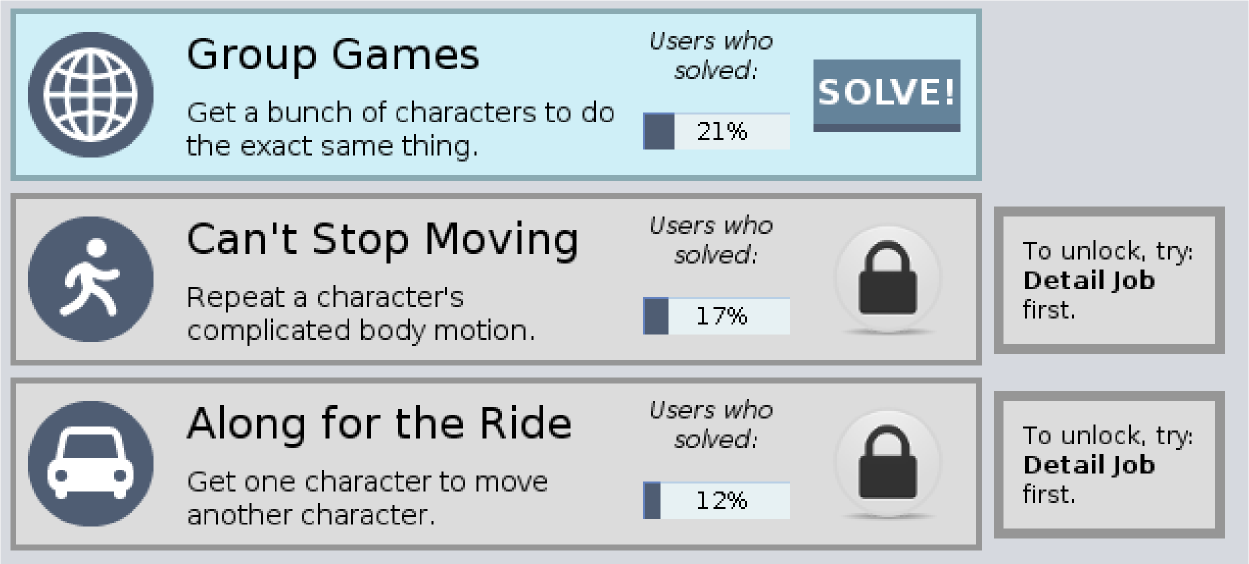
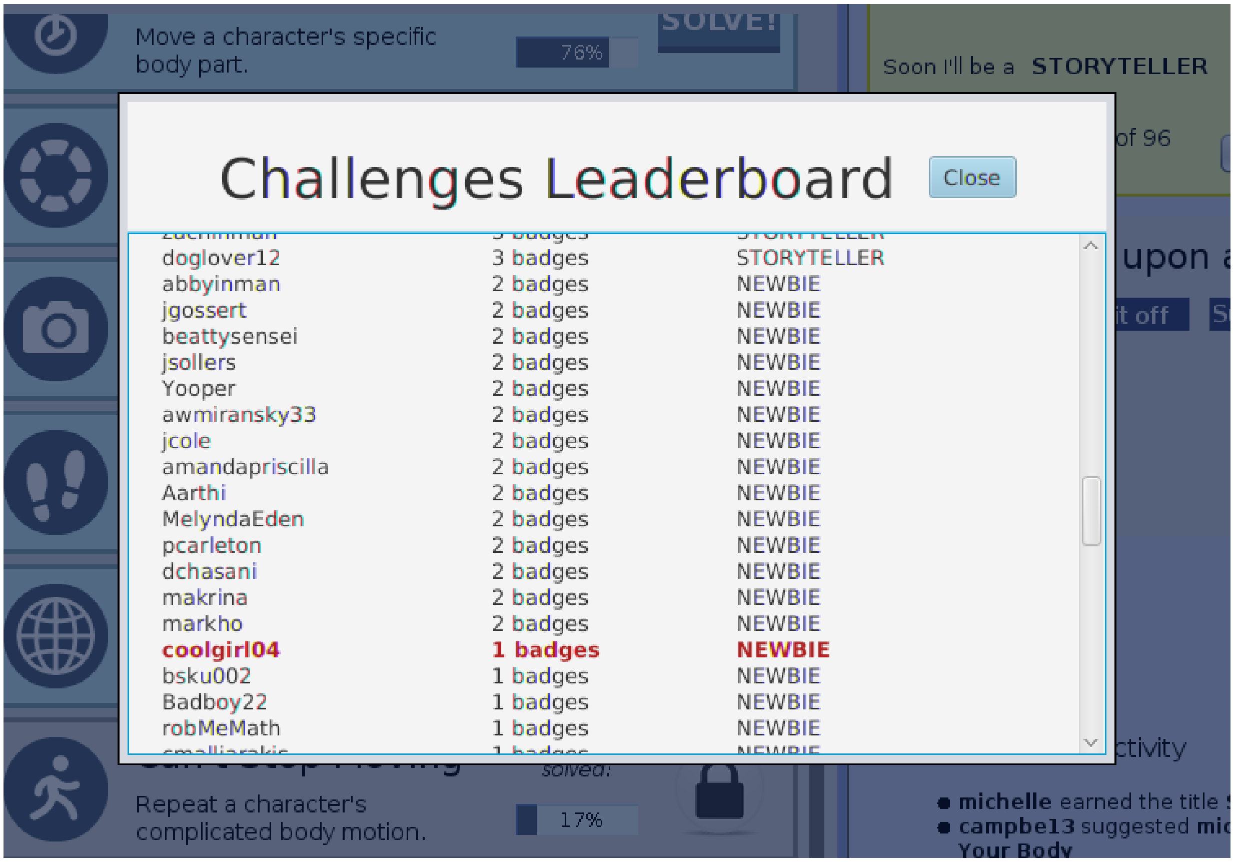
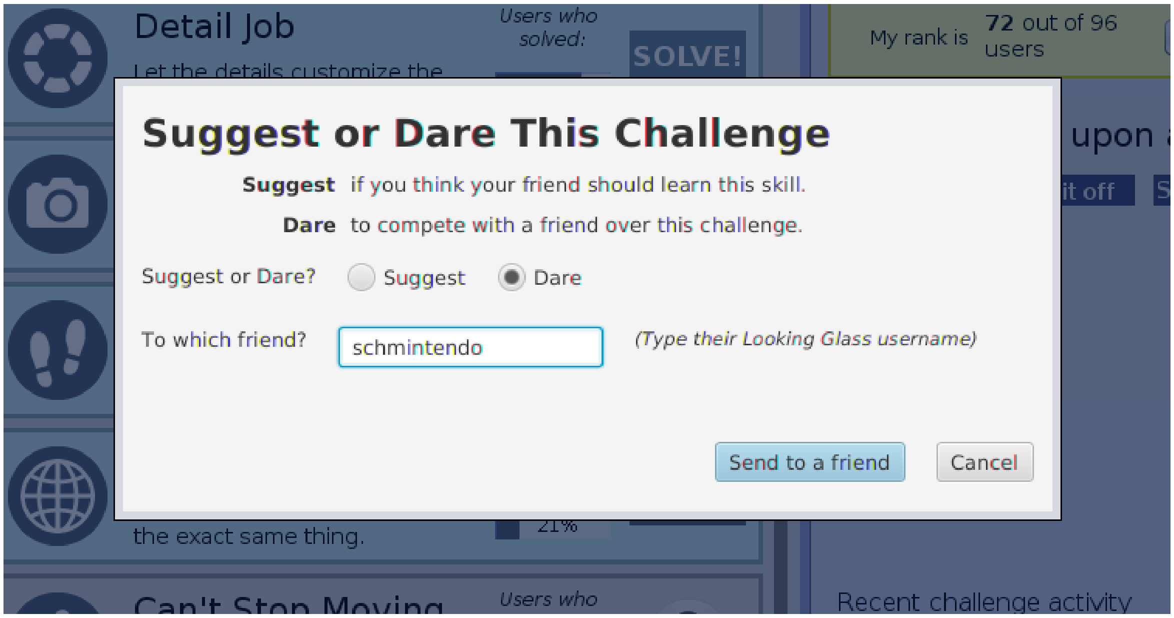

Group Games
Get a bunch of characters to do the exact same thing.
Users who solved:
21%
SOLVE!
Can't Stop Moving
Repeat a character's complicated body motion.
Users who solved:
17%
To unlock, try: **Detail Job** first.
Along for the Ride
Get one character to move another character.
Users who solved:
12%
To unlock, try: **Detail Job** first.

Figure 5.2: Locked Challenges


Move a character's specific body part.
76%
SOLVE!
Soon I'll be a STORYTELLER
of 96
Challenges Leaderboard
Close
upon a
it off
ctivity
michelle earned the title
campbe13 suggested mi
Your Body
solved:
Repeat a character's complicated body motion.
17%

doglover12	3 badges	STORYTELLER
abbyinman	2 badges	NEWBIE
jgossert	2 badges	NEWBIE
beattysensei	2 badges	NEWBIE
jsollers	2 badges	NEWBIE
Yooper	2 badges	NEWBIE
awmiransky33	2 badges	NEWBIE
jcole	2 badges	NEWBIE
amandapriscilla	2 badges	NEWBIE
Aarthi	2 badges	NEWBIE
MelyndaEden	2 badges	NEWBIE
pcarleton	2 badges	NEWBIE
dchasani	2 badges	NEWBIE
makrina	2 badges	NEWBIE
markho	2 badges	NEWBIE
coolgirl04	**1 badges**	**NEWBIE**
bsku002	1 badges	NEWBIE
Badboy22	1 badges	NEWBIE
robMeMath	1 badges	NEWBIE

Figure 5.3: Community Leaderboard



Suggest or Dare This Challenge

Suggest if you think your friend should learn this skill.

Dare to compete with a friend over this challenge.

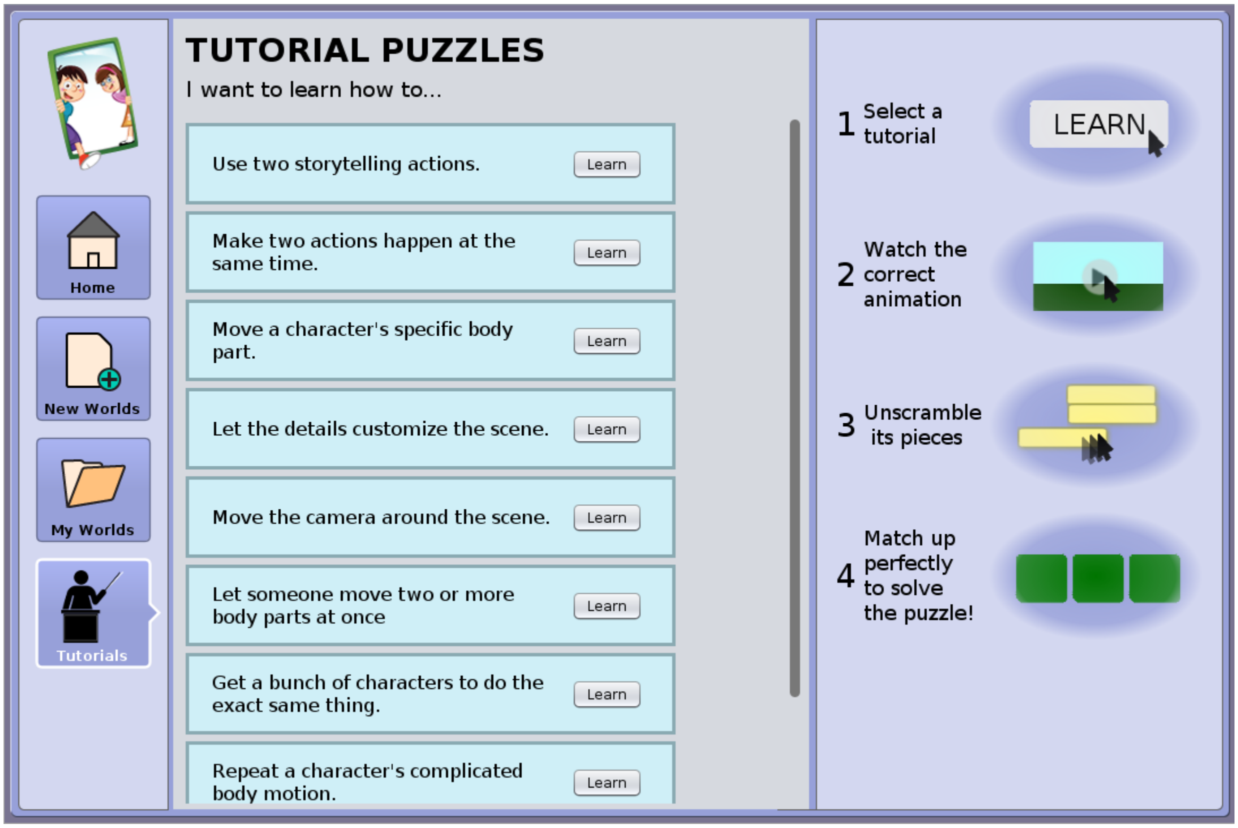
Suggest or Dare? ○ Suggest ● Dare

To which friend? schmintendo *(Type their Looking Glass username)*

Figure 5.4: Suggestions & Dares interface

To design a non-gamified control system, we determined aspects that are necessary for any challenge-based educational system. For our non-gamified educational system, we felt it sufficient to include a listing of the challenge skills a user could learn, a button to start each challenge, and short visual directions for how to solve a challenge. This system is absent of almost all of the gamified elements discussed above, yet still provides an effective introduction to the same programming skills as the gamified version. Because of the instructional nature of this system, "challenges" are instead called "tutorials."

This non-gamified version is still a reasonable interface to teach new programming concepts to Looking Glass users, as the puzzle content is identical. The only difference is the absence of the gamified interface elements. Thus, by comparing how users interact with the gamified and non-gamified versions of the interface, we can determine if there are any effects of adding the gamified elements.



TUTORIAL PUZZLES

I want to learn how to...

- Use two storytelling actions.
- Make two actions happen at the same time.
- Move a character's specific body part.
- Let the details customize the scene.
- Move the camera around the scene.
- Let someone move two or more body parts at once
- Get a bunch of characters to do the exact same thing.
- Repeat a character's complicated body motion.

1 Select a tutorial

2 Watch the correct animation

3 Unscramble its pieces

4 Match up perfectly to solve the puzzle!

Figure 5.5: The non-gamified interface, dubbed the "Tutorials" interface

5.2 Data

Our goal is to evaluate differences in user motivation between those using the gamified and non-gamified interface. However, exact motivation is an internal, unobservable quality. Therefore, we are required to use other measures as a means of getting at motivation. We chose to ask three questions that we believe relate to a user's motivational state:

1. **Is the initial experience of using the system fulfilling?** Before we investigate whether users want to return to the system, we want to know if the first exposure to a system is fulfilling enough to potentially lead to desire to return.
2. **Do users intend to return to the system in the future?** Having users self-report their intentions to return to the system might reflect their true motivational state.
3. **Do users actually return to the system in the future?** Differences in observable user behavior may be the results of what the user feels motivated to do.

We will discuss how we measured each of the above areas in turn.

5.2.1 Initial experience fulfilling

Our first question is whether a user's first exposure to a system is fulfilling enough that the user believes there might be a reason for him or her to return to it in the future. To measure this, we used the ARCS model, which was developed by John M. Keller in 1979 as a "model for motivational design" [20], that is, a systematic approach to assist teachers in designing motivational lessons and materials for their students. By reviewing psychological research on motivation, Keller determined four situational factors which must be present for a student to become motivated to begin an activity and remain motivated throughout it:

1. **Attention** -- Does the material grab the student's attention?
2. **Relevance** -- Can the student see how the material is relevant to his or her own life?
3. **Confidence** -- Does the student believe he or she has a reasonable chance at completing the material?
4. **Satisfaction** -- Does the student feel accomplished after completing the material?

By measuring these four factors, we hope to determine whether users feel sufficiently fulfilled by their first exposure to the challenge system that they would consider returning in the future.

ARCS has previously been used to evaluate the effects of gamification on educational materials, although only in a context of mandatory use. Proske et al. studied 175 college students and found students who completed game-based practice of essay-writing skills were likely to rate their Attention and Confidence significantly higher, and Satisfaction marginally higher, than non-game training [27]. Klein & Freitag found in a study of 91 college students that those who studied lecture content with a board game ranked all four ARCS categories significantly higher than those who studied with a worksheet [21].

We used a survey called the Instructional Material Motivational Survey (IMMS) developed by Keller to measure all four categories of ARCS. By giving the survey to users who have been introduced to the puzzle system, we are able to evaluate whether the interface is succeeding at engaging the four ARCS components.

5.2.2 User intention to return

Another indicator of a user's motivational state might be his or her own self-reported intention to return to a system in the future. Users who find the challenge system motivating might be more likely to report that they intend to return to the challenge system. However, simply asking users to report their intention does not help us understand the reasons for that user's response.

To better understand these reasons, we used the Technology Acceptance Model (TAM) for computer systems. In the model, developed Fred Davis suggested a user's intention to use a computer system is mainly influenced by three factors:

- The perceived usefulness (PU) of the system [3][28]
- The perceived ease of use (PEOU) of the system [3][28]
- The perceived enjoyment (PE) from using the system [6].

By measuring these three factors, along with intention to use, we can get a better idea of why users differ in their self-reported intention.

There is some previous work that used TAM in a similar context. Herzig et al. found that business training software designed as a video game was rated significantly higher for perceived enjoyment and perceived ease of use, and slightly lower in perceived usefulness, than non-game software [18]. Although this was in the context of business training, Sung Youl Park found that TAM was successful at predicting intention to use educational software among 623 college students [26], showing TAM to be valid in educational contexts as well.

5.2.3 Observed behavior

Although motivation is an internal state, one would expect a user's motivation towards using a system to be expressed through their actual behaviors when using the system. Therefore, our third method for examining motivation is to track users' behavior when actually using Looking Glass. For this, we tracked all of a user's interactions while using Looking Glass. Specifically for motivation, we are interested in:

- Tracking each time a user chooses to complete a puzzle
- How long a user spends working on the puzzle

We were also interested in whether any interface interactions were tied to increased puzzle behavior. Firstly, we were interested if interactions with the gamified elements had any ties to puzzle behavior. For example, it might be that users who were interested in their leaderboard rank solved more puzzles than those who were not interested. To capture this, we logged any mouse clicks on gamified interface elements, as well as whether the user's mouse was hovering over gamified interface elements.

Additionally, we track a user's actions while building their animations, as well as the content of the animations the user was working on. This allows us to investigate whether there are any significant differences in code editing between users in the study.

5.2.4 Additional data

In addition to data related to motivation, we collected three additional sources of data:

- Demographic data, such as age, gender, and previous programming experience, to see if any of these factors affected motivation
- Intrinsic Motivation Inventory (IMI), which asked users evaluate the quality of their experience using Looking Glass, to see if gamified elements had any impact on the overall Looking Glass experience
- Qualitative opinion information, for information on why users chose to attempt a specific puzzle, and which elements of the gamified or non-gamified interface they found most engaging.

5.3 Participants

We recruited 44 participants (36 male, 8 female) between the ages of 10 and 15 (μ =12.20, σ = 1.34) through the Academy of Science of St. Louis mailing list. Each participant received a $10 Amazon.com gift card in recognition of his or her participation. Participants ranged in programming experience from no programming experience to programming in their free time. All participants rated their computer skills as "Fair" or above.

5.4 Methods

Study sessions lasted for 90 minutes and were held in a computer lab on the Washington University in St. Louis Danforth campus. Participants first completed a computing history survey that collected various demographic data and information about programming experience. After completing the survey, participants were randomly assigned to a computer that was part of either the control (gamified) or experimental (non-gamified) condition. Computers were arranged in such a way that each participant could only see screens of others within the same condition.

Once at their computer, participants watched a short video that provided an overview of how to create animations in Looking Glass. Specifically, participants saw how to view a character's actions, add actions to a program, and play an animation. We believed it would be important for participants to have a basic understanding of the Looking Glass system to relate the content of puzzles to animating stories. Participants then were asked to come up with a username and were told to use it to log in to the Looking Glass community. The computers were not hooked up to the community, but formative testing showed that attitudes about the gamified elements were tied to whether a user felt his or her progress would be relatable back to himself or herself. After this point, no additional assistance was given by the experimenter for using Looking Glass or solving puzzles, except to restart the system if it crashed. With this restriction, the experimenter never mentioned puzzles, tutorials, or challenges during the session.

We wanted to make sure that all participants had a basic understanding with the puzzle system, as this would ensure that voluntary usage of the system was not affected by discovery or unfamiliarity with the system, but instead by actual motivation. In addition, this allowed all users to evaluate their initial assessment of the puzzle system, regardless of whether they returned to it voluntarily. Each user was given a sheet of steps that asked them complete two simple puzzles—one teaching basic Looking Glass storytelling actions, and another for the Do Together parallelism construct. Note that these steps contained no instructions for how to complete a puzzle, nor explained any reason for completing the puzzles. After completing these two puzzles, participants filled out the IMMS to evaluate their feelings about the attentional qualities, relevance of, confidence towards, and satisfaction with the puzzles. During the completion of surveys, Looking Glass was closed.

Next, we wanted to examine voluntary choice to complete puzzles in a tightly controlled setting. This allows us to look at differences in puzzle choice between the gamified and non-gamified interfaces in the presence of as few external factors that may affect motivation as possible. To achieve this, participants were given the option to complete between 0 and 2 additional puzzles, and asked to explain their choice of quantity. Participants were informed that there was no penalty for choosing no puzzles, and that they could return to do additional puzzles later. We limited each attempt to complete a puzzle to 7.5 minutes based on our pilot study, in which users often had difficulty exiting a puzzle, even when it no longer was enjoyable. After completing the chosen number of puzzles, participants wrote down which aspects of the challenge or tutorial system most encouraged them to complete more puzzles, and then filled out a TAM evaluation, evaluating their intention to return to the system in the future.

Lastly, we wanted to examine voluntary choice to complete puzzles in a setting that approximated real-world usage of Looking Glass. In actuality, there are no restrictions on what actions a user can take while using Looking Glass. Therefore, we had the remainder of the session be free time for users to either work on a Looking Glass animation or return to challenges or tutorials. Participants were informed that they were free to do either code editing or puzzles in whatever combination they would like for however long they would like. Participants worked up until there were 5 minutes remaining in the session, at which point their animations were saved, and they were asked to fill out the IMS survey.

Chapter 6

Results and Discussion

6.1 Differences in measures

Our study aimed to answer the question: Are users of the gamified interface more motivated than users of the non-gamified interface? We examined motivation in three ways: Initial evaluation of the system, self-reported intention to use the system, and observed usage of the system. We will discuss our findings for each of these three areas in turn.

6.1.1 Initial experience fulfilling

To begin, we investigate whether users in the gamified and non-gamified conditions had differences in their initial evaluations of the puzzle system. We began by performing a one-way MANOVA analysis of the four ARCS subcategories between the gamified and non-gamified conditions, which was found to be significant (F(4, 37) = 2.997, *p* = 0.031). Based on this, we performed Kruskall-Wallis tests to determine the significance of the mean differences between the ARCS categories, as the tests are less sensitive to the highly non-normal survey data than ANOVA. Mean values were normalized from a 7-point Likert scale, with a maximum value of +3 and a minimum value of -3. In Table 6.1 we report these mean differences and *p* values. The mean differences are additionally illustrated in Figure 6.1.

Table 6.1: ARCS mean differences between non-gamified and gamified interfaces

	μ (control)	**μ (experimental)**	**diff.**	***p***
Attention	0.56	1.05	0.49	0.07*
Relevance	0.90	0.82	0.08	0.90
Confidence	1.92	1.99	0.07	0.408
Satisfaction	0.56	1.22	0.66	0.034**

*: p < 0.10 **: p < 0.05

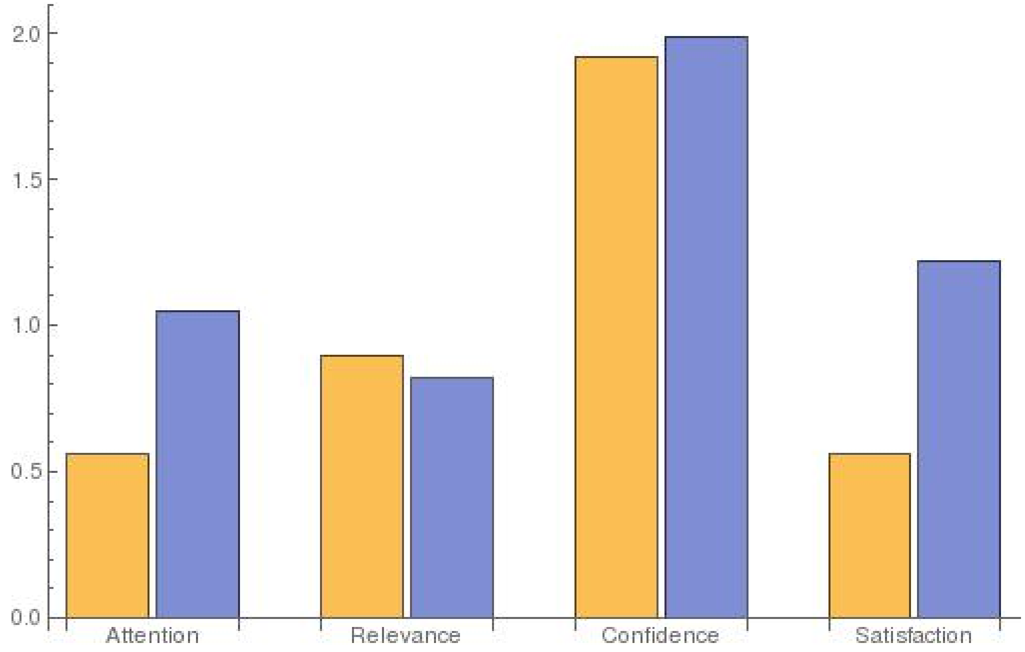


Figure 6.1: Means of ARCS scores for non-gamified (orange) and gamified (blue) interfaces.

Based on this analysis, we see a borderline significant difference in evaluations of attention, and a significant difference in evaluations of satisfaction. These are both logical differences: by adding visually appealing and enticing interface elements, the interface would be assumed to hold the attention of users more effectively; and by adding additional rewards for puzzle completion, one would expect users to feel more satisfied after completing puzzles. The two insignificant differences are also logical: since the puzzles between both conditions were identical, one would expect neither condition to find the system more relevant to their needs, nor would it make sense for one condition to feel more confident about completing the puzzles than the other condition. From these observations, it appears that participants with the gamified interface ranked their initial experience with the puzzle system as more fulfilling than those without it.

6.1.2 Intention to return

Next, we look at users' self-reported intentions to return to the puzzle system, and the factors that affect their intentions. We again performed a one-way MANOVA on the TAM factors between the gamified and non-gamified conditions, finding it to be significant (F(4, 37) = 3.103, $p = 0.027$). Based on this, we performed Kruskall-Wallis tests to determine the significance of the mean differences between the TAM factors. Mean values were again normalized from a 7-point Likert scale, with a maximum value of +3 and a minimum value of -3. In Table 6.2, we report these mean differences and *p* values. The mean differences are additionally illustrated in Figure 6.2.

Table 6.2: TAM mean differences between non-gamified and gamified interfaces

	μ (control)	**μ (experimental)**	**diff.**	***p***
Ease of use	1.38	1.72	0.34	0.40
Usefulness	1.55	2.07	0.52	0.262
Enjoyment	0.89	1.62	0.73	0.014**
Behavioral intention	0.50	1.52	1.02	0.0295**

*: p < 0.10 **: p < 0.05

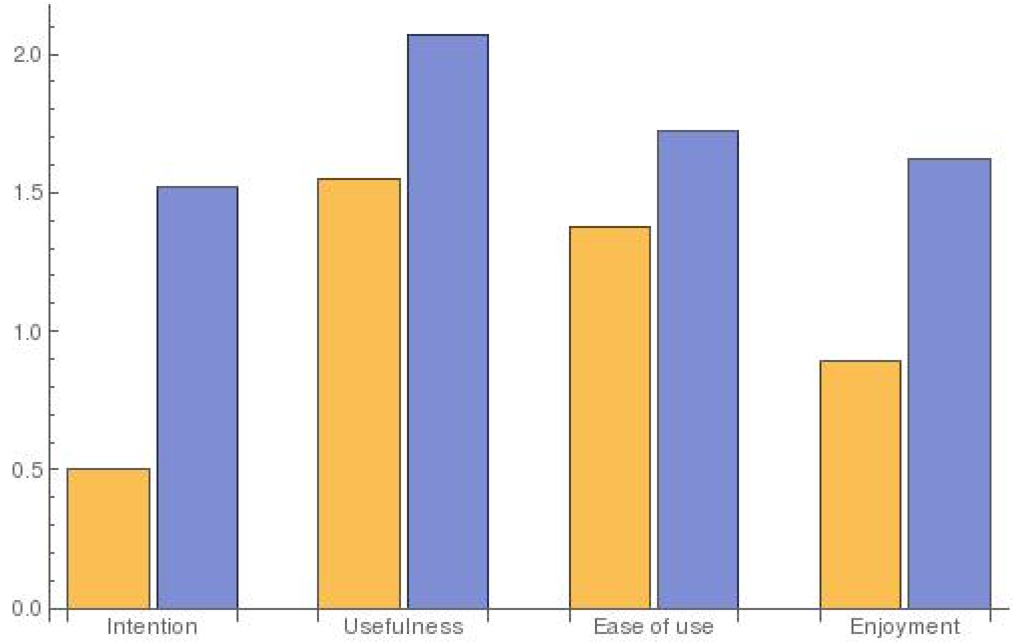


Figure 6.2: Means of TAM scores for non-gamified (orange) and gamified (blue) interfaces.

Most importantly, we see a significant difference with behavioral intention to use, with those in the gamified condition reporting their intention to return as higher than those in the non-gamified condition. This suggests that those in the gamified condition are in fact more motivated than those in the non-gamified condition to return to the puzzle system.

For the three factors that affect intention to return, we see a significant difference only in system enjoyment. This seems logical based on the system differences: the interface with more visually appealing elements and additional rewards is likely to be a more enjoyable experience than the interface without. The two non-significant factors also make sense: there would not be any expected difference in the ease of use of the two systems, nor the usefulness, as the puzzle system and puzzle content are identical for both.

6.1.3 Observed behavior

Our third way of investigating motivation is through the observed behavior of users. For our analysis, we considered differences in the means for four pieces of behavioral data:

- Recall that before their free editing time, participants were given the option to complete between 0 and 2 additional puzzles. We recorded how many puzzles they completed at this point, and report it as "Up to 2 challenges."
- During the users' free editing time, we tracked how many different puzzles were attempted. We report this as "Attempted."
- The start and end times for each puzzle attempt during free time were logged. We summed all of a user's editing time and report it in seconds as "Puzzle time."
- We divided the amount of time spent editing puzzles by the total amount of free time a user had to obtain the percentage of free time spent on puzzles, reported as "Percent."

We performed a one-way MANOVA analysis comparing these behaviors between the gamified and non-gamified interfaces, and found it to be significant (F(4, 37) = 2.997, p = 0.031). Based on this, we performed an ANOVA on each of the four measures, which we report in Table 6.3.

Table 6.3: Behavior mean differences between non-gamified and gamified interfaces

	µ (Control)	µ (Experimental)	Difference	F	*p*
Up to 2 challenges	1.38	1.65	0.27	1.681	0.202
Attempted	1.29	2.22	0.93	2.985	0.091
Puzzle time	145.39	427.00	281.61	7.150	0.012
Percent	0.05	0.16	0.11	6.841	0.011

From our analysis, we see no significant difference between the number of puzzles chosen during the task when up to two could be chosen, and only a marginally significant difference between the number of puzzles chosen during the free editing time. However, we do see significant differences between the amount of time spent solving puzzles during free time, as well as the percentage of free time that was spent on puzzles. This seems to indicate that those in the gamified condition had a greater motivation to work on puzzles, as expressed by spending more time on these puzzles.

6.2 Relations between measures

Beyond examining differences between measures, we further look at the relationship between these three measures. We initially posited that all three of our measurements were methods of examining the hidden state of user motivation. It is reasonable to expect that, were this the case, relationships would be observed between the three measures. Specifically, one would expect that the initial evaluation of the system to be related to the future intention to use the system; and that the intention to return would be related to the actual observed system usage. We will look at both of these questions in turn.

6.2.1 Initial evaluation vs. Intention to use

It seems reasonable to hypothesize that a user's initial evaluation about whether an interface suits his or her needs would be correlated with that user's desire to return to that system later. Therefore, one might expect that the factors measured in the ARCS evaluation would be related to the factors measured in the TAM evaluation. To investigate whether this was observed, we performed Pearson correlation tests between each factor of the two measures, and report our results in Table 6.4.

Table 6.4: Pearson's r coefficient for ARCS & TAM correlations

	PEOU		PU		PE		BI	
	r	*p*	*r*	*p*	*r*	*p*	*r*	*p*
A	*0.22*	*0.14*	0.53***	0.0002	0.69***	1.6x10^{-8}	0.45***	0.002
R	*0.2*	*0.17*	0.59***	0.00002	0.46***	0.001	0.41***	0.004
C	*0.22*	*0.13*	*0.001*	*0.99*	0.4**	0.005	*0.19*	*0.21*
S	*0.05*	*0.2*	0.45***	0.002	0.63***	4.3x10^{-7}	0.51***	0.0004

*: p < 0.05;   **: p < 0.01;   ***: p < 0.005;   *italics*: p > 0.10

Firstly, we can observe that three factors from ARCS, attention, relevance, and satisfaction, have significant correlations with behavioral intention to use the system in the future, each with *p* < .002 and *r* > 0.40. This supports that the initial experience using the challenge system is related to the later reported intention to use the system.

Additionally, we can observe several additional significant relationships between ARCS and TAM. Many of these significant relationships seem very logical: it makes sense that the more attention grabbing material is, the more a user would enjoy it; also that the material which the user is confident that he or she can complete is more enjoyable; and the relevance of material would make sense to be related to how useful it is perceived to be. Unfortunately, several of the other areas lack as simple an explanation for the observed differences. For these, we can simply note that the factors which are indicative of a positive initial experience are related to the factors which influence intent to return to a system.

6.2.2 Intention to use vs. actual use

Another relationship we would hope to validate is a connection between intention to use a system and the actual observed usage of the system. Since we posit that self-reported intention reflects a user's motivation to use a system, and system usage is a realization of that motivation, a relationship between the two would provide evidence that both are truly representative of user motivation. To investigate this we performed an ANOVA for each of the behavioral measures with respect to reported intention to use, and report our results in Table 6.5.

Table 6.5: Correlations between intention to use and behavioral measures

	r	*p*
Up to 2 challenges	0.474	0.001
Attempted	0.418	0.005
Free time	0.379	0.011
Percent	0.382	0.011

As expected, self-reported intention to use is significantly correlated with all four of the behavioral measures. Firstly, this provides evidence that self-reported intention to use is a valid predictor of actual system usage within this context. Secondly, this relationship provides additional support that both of these two measures are accurate reflections of a user's motivational state.

Chapter 7

Conclusion

In this study, we investigated whether users of a gamified interface are more motivated than users of a non-gamified interface to complete optional lessons. We used an open-ended novice programming environment as our basis, since motivating voluntary use presents an opportunity for independent learning of programming. We performed three rounds of formative testing to design this system, in which we found it problematic to integrate gamified elements into a regular code editing system, but more successful to separate the gamified elements into a related but distinct system.

We observed increased motivation to use the gamified system in three ways. First, we found that users of the gamified interface ranked their initial exposure to the system as more fulfilling than those with the non-gamified interface. Second, we found users of the gamified interface had a higher self-reported intention to return to the system than those with the non-gamified interface. Third, those with the gamified interface were observed spending longer using the system than those with the non-gamified system. Further, we found the initial evaluation to be correlated with self-reported intention, and the self-reported intention to be correlated with actual observed system usage. These results support the addition of gamified elements as a means of motivating users to use a gamified educational system.

7.1 Future Work

Although this study was able to investigate how participants chose to voluntarily interact with the gamified lessons, it was limited to a 90 minute session. It would be additionally informative to investigate how users interact with the puzzle system over a period of several days, including whether the inclusion of gamified elements has an effect on how often users would return a system in general. In addition, it would be important to note how the attitudinal aspects discussed in this study vary over long-term use of the application, such as whether certain aspects have stronger influences earlier or later in the lifetime usage of material.

While many participants in the experimental condition noted interface elements that encouraged them to solve more puzzles, half of the participants with the gamified interface did not list any gamified elements. Similarly, previous studies have observed that gamification tends to only affect a subset of study participants. Further research into why some users are motivated by gamified elements while others are not, and how to identify such users, would help understand how to best engage users with gamified interfaces. Similarly, due to the small sample of responses (*N* = 11), conclusions cannot be drawn about the motivational qualities of individual interface elements. A larger sample size may be able to tie responses to these individual elements to behavioral or attitudinal observations.

Lastly, we informally observed in our formative and summative testing that exposure to new programming skills through puzzles led to use of these skills during open-ended code editing. Since this study provided evidence that gamified elements motivate users to solve more puzzles, formal evaluation of puzzles' educational effectiveness will further confirm that a gamified puzzle system provides an effective means for independently allowing middle school students to learn programming skills.

References

[1] Anderson, Ashton, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. "Steering user behavior with badges." In *Proceedings of the 22nd international conference on World Wide Web*, pp. 95-106. International World Wide Web Conferences Steering Committee, 2013.

[2] Antin, Judd, and Elizabeth F. Churchill. "Badges in social media: A social psychological perspective." In *CHI 2011 Gamification Workshop Proceedings (Vancouver, BC, Canada, 2011).* 2011.

[3] Davis, Fred D. "A technology acceptance model for empirically testing new end-user information systems: Theory and results." PhD diss., Massachusetts Institute of Technology, 1985.

[4] Davis, Fred D. "Perceived usefulness, perceived ease of use, and user acceptance of information technology." *MIS quarterly* (1989): 319-340.

[5] Davis, Fred D., Richard P. Bagozzi, and Paul R. Warshaw. "User acceptance of computer technology: a comparison of two theoretical models." *Management science* 35, no. 8 (1989): 982-1003.

[6] Davis, Fred D., Richard P. Bagozzi, and Paul R. Warshaw. "Extrinsic and intrinsic motivation to use computers in the workplace1." *Journal of applied social psychology* 22, no. 14 (1992): 1111-1132.

[7] Denny, Paul. "The effect of virtual achievements on student engagement." In*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 763-772. ACM, 2013.

[8] Dong, Tao, Mira Dontcheva, Diana Joseph, Karrie Karahalios, Mark Newman, and Mark Ackerman. "Discovery-based games for learning software." In*Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2083-2086. ACM, 2012.

[9] Durga, Shree, Magy Seif El-Nasr, Mariya Shiyko, Carmen Sceppa, Pamela Naab, and Lisa Andres. "Leveraging Play to Promote Health Behavior Change: A Player Acceptance Study of a Health Game." In *Virtual, Augmented Reality and Serious Games for Healthcare 1*, pp. 209-230. Springer Berlin Heidelberg, 2014.

[10] Ericsson, K. Anders, Ralf T. Krampe, and Clemens Tesch-Römer. "The role of deliberate practice in the acquisition of expert performance." *Psychological review* 100, no. 3 (1993): 363.

[11] Gill, Judith. "Shedding Some New Light on Old Truths: Student Attitudes to School in Terms of Year Level and Gender." (1994).

[12] Grant, Scott, and Buddy Betts. "Encouraging user behaviour with achievements: an empirical study." In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pp. 65-68. IEEE, 2013.

[13] Hakulinen, Lasse, Tapio Auvinen, and Ari Korhonen. "Empirical study on the effect of achievement badges in TRAKLA2 online learning environment." In*Learning and Teaching in Computing and Engineering (LaTiCE), 2013*, pp. 47-54. IEEE, 2013.

[14] Halan, Shivashankar, Brent Rossen, Juan Cendan, and Benjamin Lok. "High Score!-Motivation Strategies for User Participation in Virtual Human Development." In *Intelligent Virtual Agents*, pp. 482-488. Springer Berlin Heidelberg, 2010.

[15] Hamari, Juho. "Transforming homo economicus into homo ludens: A field experiment on gamification in a utilitarian peer-to-peer trading service."*Electronic commerce research and applications* 12, no. 4 (2013): 236-245.

[16] Hamari, Juho, Jonna Koivisto, and Tuomas Pakkanen. "Do Persuasive Technologies Persuade?-A Review of Empirical Studies." In *Persuasive Technology*, pp. 118-136. Springer International Publishing, 2014.

[17] Hamari, Juho, Jonna Koivisto, and Harri Sarsa. "Does Gamification Work?--A Literature Review of Empirical Studies on Gamification." In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 3025-3034. IEEE, 2014.

[18] Herzig, Philipp, Susanne Strahringer, and Michael Ameling. "Gamification of Erp systems-exploring gamification effects on user acceptance constructs." In*Multikonferenz Wirtschaftsinformatik*, pp. 793-804. 2012.

[19] Hsu, Shang Hwa, Jen-Wei Chang, and Chun-Chia Lee. "Designing attractive gamification features for collaborative storytelling websites." *Cyberpsychology, Behavior, and Social Networking* 16, no. 6 (2013): 428-435.

[20] Keller, John M. Charles M. Reigeluth, ed, *Instructional-design theories and models*. Hillsdale, N.J: Lawrence Erlbaum Associates, 1983. pp. 383 - 434

[21] Klein, James D., and Eric Freitag. "Effects of using an instructional game on motivation and performance." *The Journal of Educational Research* (1991): 303-308.

[22] Li, Wei, Tovi Grossman, and George Fitzmaurice. "Gamicad: a gamified tutorial system for first time autocad users." In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 103-112. ACM, 2012.

[23] Malone, Thomas W. "What makes things fun to learn? Heuristics for designing instructional computer games." In *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pp. 162-169. ACM, 1980.

[24] McGonaigal, Jane. "Gaming can Make a Better World..." (2010).

[25] Montola, Markus, Timo Nummenmaa, Andrés Lucero, Marion Boberg, and Hannu Korhonen. "Applying game achievement systems to enhance user experience in a photo sharing service." In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pp. 94-97. ACM, 2009.

[26] Park, Sung Youl. "An Analysis of the Technology Acceptance Model in Understanding University Students' Behavioral Intention to Use e-Learning."*Educational Technology & Society* 12, no. 3 (2009): 150-162.

[27] Proske, Antje, Rod D. Roscoe, and Danielle S. McNamara. "Game-based practice versus traditional practice in computer-based writing strategy training: effects on motivation and achievement." *Educational Technology Research and Development* 62, no. 5 (2014): 481-505.

[28] Venkatesh, Viswanath, and Fred D. Davis. "A model of the antecedents of perceived ease of use: Development and test*." *Decision sciences* 27, no. 3 (1996): 451-481.

[29] Wilson, Cameron, Leigh Ann Sudol, Chris Stephenson, and Mark Stehlik. "Running on empty: The failure to teach K-12 computer science in the digital age. Association for Computing Machinery." *Computer Science Teachers Association* (2010).

Gamified Optional Lessons & Motivation, Zemach, M.S. 2014