Comparing Student Engagement in Educational Programming Games With and Without Narrative Elements.

Theo Thesen
THSTHE004@myuct.ac.za
University of Cape Town

Abstract

Introductory programming courses pose a challenge to students and teachers alike. This is due, in part, to the fact that students struggle to maintain engagement with the coursework, and this has been shown in turn to have a negative effect on knowledge acquisition. Game-Based learning systems have emerged as a powerful medium to circumvent this problem, and, as a result, they have seen undergone substantial investigation in both theoretical and practical applications in recent years, yet the question of how they affect student engagement is still under debate. One argument as to why this is the case is that games are too broad of a topic to effectively research and should instead be broken down into core elements of games whose effects on introductory programming can be studied individually. This project presents an analysis of the narrative element of games and introduces it as a valuable aspect of game-based learning systems that has a positive effect on student engagement in an introductory programming course. To put this hypothesis to the test, an educational introductory programming game was developed with toggle-able narrative elements. Experimental research was conducted to measure the difference in student engagement between the narrative and non-narrative versions of the game.

CCS Concepts: • **Computer Science Education** → Teaching Introductory Programming; • **Game-Based Learning Systems** → Narratives; Story Telling; • **Applied Computing** → Interactive Learning Environments.

1 INTRODUCTION

Introductory programming has been shown to be a common point of attrition for students looking to study Computer Science [3, 5]. Statistical analysis of 161 CS1 courses taught in 15 countries concluded that there is a global mean pass-rate of just 67.7% for introductory programming courses [50]. Part of this problem is due to the fact that current pedagogical approaches towards teaching introductory programming struggle to maintain student engagement. These approaches tend to work for some students yet cause many other students to disengage from their coursework [18]. This disengagement has been shown to have a negative effect on knowledge acquisition and information assimilation [34].

Solutions to this problem have been introduced, with Game-based Learning (GBL) systems emerging as a popular mechanism to counteract the problem of student engagement in introductory programming courses [21, 43, 45]. The issue is that although GBL systems as pedagogical tools for teaching introductory programming have seen largely positive results in both theory and practical application, they are highly variable. The reason for this high degree of variation and unpredictability when it comes to measuring the effectiveness of game-based learning systems and pedagogical tools is understood to stem from the fact that games, as systems, are too vaguely defined and encapsulate too broad a range of elements to yield any pertinent and consistent research conclusions[27]. Current literature indicates that an effective strategy to circumvent these varied results is to isolate particular elements of games and test the effect that said particular element has on student engagement in isolation [27, 53]. In this way, hidden variables are accounted for by measuring the differences in student engagement when the selected element is present within the game-based learning solution and when it is not.

1.1 Problem Statement

It has been definitively established that, in certain contexts, digital game-based learning can be an effective tool to increase student engagement[9, 12, 16, 29, 52, 54]. However, there is a distinct lack of research into which elements of games would be best suited to facilitate student engagement. Digital games are typically complicated and varied systems utilising a host of different game design principles that are interconnected, making it difficult to benchmark or evaluate results when conducting research into educational games [27, 53]. The literature indicates that when conducting research on the effectiveness of game-based learning systems, it is better to experimentally evaluate different game design elements in isolation to determine their impact on student engagement.

Narratives are utilized in games, other forms of popular media, and education to tie linked concepts together to form an over-arching thematic concept [20, 39]. They have also been shown to be reliable sources of engagement and emotional attachment [10, 32].

Research indicates that Game-based learning systems work best as a pedagogical tool when utilized hand-in-hand with other, more traditional pedagogical methods [23, 27]. Furthermore, it has been demonstrated that students struggle to maintain engagement with the material taught in introductory programming courses [18].

This suggests that the introduction of narrative elements into game-based learning systems may help to solve some of the problems facing game-based learning systems as a pedagogical tool for student engagement in introductory programming courses. This is because narrative elements have been shown to be reliable sources of sustained, intrinsic engagement, which is an indicator of improved knowledge acquisition in an introductory programming context, [34] and introductory programming courses struggle to maintain [18]. Furthermore, narrative elements are useful as a tool to tie linked concepts together to form a thematic context. Therefore, the introduction of narrative elements into a game-based learning context may allow for teachers to easily associate concepts within the game-based system with concepts taught using other pedagogical strategies.

This research paper attempts to take meaningful steps toward solving the problem of student engagement in introductory programming courses and goes on to posit that narrative elements in game-based learning systems have a positive effect on student engagement in introductory programming courses.

1.2 Aim

To answer the research question, a game-based learning system that has toggle-able narrative elements will be implemented. In being able to play the game with and without narrative elements, the effect that narrative elements have on student engagement in the context of an introductory programming course can be isolated and comparatively evaluated.

The game-based learning system will function without narrative elements and will incorporate some of the elements that game-based learning systems utilize that are known to help people learn introductory programming. For example, displaying visual objects that represent an abstract concept like for-loops or if-statements can help students to conceptually understand what purpose those programming concepts have.

A person playing the game will have to solve introductory programming challenges in order to complete it. These challenges will cover specific introductory Computer Science topics and will get progressively more difficult as a player advances through them.

In narrative mode, the player will be able to interact with the game-world in a meaningful way that will provide further narrative substance to the game and appeal to the players emotional responses while playing, in an attempt to provide intrinsic motivation for a player to continue interacting with the game-based learning system.

2 BACKGROUND

2.1 The Problem of Introductory Programming

Learning introductory programming poses a challenge to many students [19, 25]. In fact, there is an inordinately high attrition rate of 30-40% for Computer Science majors, with the majority of them opting to drop out during introductory programming courses [3, 5]. One understanding of this perceived difficulty is due to programming not being a single skill, but rather an application of a set of skills [3]. There is extensive research in this field - towards identifying, describing and preventing causal factors that contribute towards the perceived difficulty and high drop-out rates of introductory programming courses [46].

Learning introductory programming calls for continued motivation and engagement. This is something that teachers struggle to instill in their students, and students struggle to maintain [18]. Disengagement from introductory programming has been shown to lead to a decrease in student participation, information assimilation and overall knowledge gain [34]. On the other hand, maintained engagement with introductory programming courses tends to result in meaningful acquisition of knowledge [11, 34] and has been shown to have a statistically significant effect on academic achievement [11]. Students tend to approach programming with motivations that stem from different sources [24]. Some students are intrinsically motivated by sheer interest in the coursework whereas others are motivated extrinsically by factors like the desire for financial stability and social pressure. Students that are intrinsically motivated have been found to generally struggle with their programming coursework less than those who are extrinsically motivated [25, 44]. This provides a motive to curate introductory programming courses that are intrinsically motivating in order to increase student engagement and knowledge acquisition within the coursework.

A given student's cognition also comes into play. Learners have been shown to absorb knowledge in different ways. Some students respond positively to static learning environments while others respond to dynamic or interactive learning environments [25]. The challenge in introductory programming courses is then to teach coursework in such a way that these different learning styles are accommodated for so as to maximize knowledge acquisition across different learning styles [49].

Much literature has been written on the topic of how best to design introductory Computer Science courses, accounting for student cognition, learning styles and learning environments [4, 33, 48, 49, 51]. There is still significant debate about best approaches towards teaching introductory programming courses [45], but recent trends in pedagogical strategies for introductory programming have embraced the concept of gamification and game-based learning systems, with largely positive, yet highly variable, results [4, 31].

2.2 Gaming-Based Learning Systems

Gamification is a strategic approach that derives from the domain of game design and involves developing information systems and mechanics that incorporate game design elements, which are intrinsically motivating[21], to aid productivity and engagement within a given system [7, 26]. Although gamification has been extensively researched, it suffers from an imprecise definition. It is described by Deterding, Dixon & Khaled (2011) [15] as a means of applying 'gamefullness', 'gameful interaction' and 'gameful design' in non-game contexts. This contrasts other definitions, such as Al-Azawi, Rula & Al-Faliti (2016) [1], who describe gamification as the process in which a system as a whole is turned into a game. For the purposes of this project, we introduce the term GBL, or Game-Based Learning to refer to the process of creating a game for the express purpose of teaching something. Game-based learning systems differ from conventional gamification practices in that they do not borrow features and mechanics from game design, but rather they leverage video games as a distinct medium of conveying information [1, 14] and take advantage of unique aspects of gaming to create more comprehensive learning environments and increase knowledge acquisition [1].

Research into Game-based learning systems in educational environments indicates that they are an effective pedagogical tool [27, 35, 38, 53]. They are not without their faults though, as game-based learning systems as pedagogical tools in an introductory programming context have seen varied, yet largely positive, results in the literature. This has been attributed to the fact that games, and as a result, gamification and game-based systems are subject to too much variation and too vaguely defined to yield any pertinent research conclusions [15, 27].The literature indicates that an effective means of counteracting these varied results is by isolating one particular element of games and testing the effect that said element has on student engagement in isolation [27, 53].

2.3 Narratives in Game-Based Learning

Narrative, or storytelling, is a spoken or written account of connected events. It serves to tie linked concepts together to create an over-arching thematic context [20, 39]. As an individual experiences a narrative, they progressively construct models of meaning that represent the elements and characters within the narrative based on experiences and knowledge they already have [10]. Critically, an individual finds themselves updating their mental models as a narrative progresses [55]. Narrative elements also motivate people to engage with a system. There is extensive research in this field [10, 40] and substantial conclusions have been drawn about the positive effects of narratives on engagement [10, 32].

Narrative elements are important parts of what constitutes a game. The ability to engage [10], transport [6, 20] and draw emotion [13] out of a player contributes massively to said player's sense of enjoyment and satisfaction while playing [8, 20, 22]. Little conclusive research has been published on the specific effects of narrative and storytelling elements on player engagement within gaming [13, 27]. Many papers, however, suggest there is a strong correlation [13, 37, 42].

One crucial distinction between narratives in gaming and narratives in other forms of media is the fact that a person playing a game with narrative elements is not only being told a story, but they are an active participant in shaping the construction and outcome of the narrative [39]. This means that as a character in a game, a person must actively engage with the game in order to achieve a desirable narrative outcome. Salen & Zimmerman (2004) [41], found that other elements of gaming (goals, conflict) interact with narrative elements and, with the player as a character, result in a greater degree of narrative comprehension and immersion. This is naturally conducive to educational games as pedagogical challenges can be set that, when solved, help the player to actively further the narrative.

Alongside this, narrative elements are already littered throughout pedagogy [2, 28]. For example, story sums in maths classrooms are a manner of leveraging the ability of narratives to tie associated elements together and link them in a grander thematic context [20, 39]. This results in students garnering not only a deeper understanding of the coursework, but also a deeper understanding of the greater scheme into which the coursework fits [28].

The effect of narrative elements in a game-based learning system have not been extensively, or precisely researched. This is due, in part, to the fact that game-based systems are composed of multiple elements and subject to wide variation [21, 23, 53] and narrative elements are an often overlooked aspect of gaming, as other popular forms of media (film, print, etc.) utilize narrative elements in a more obvious and central way [39] and are therefore a more popular medium of study when looking at the effects of storytelling on introductory programming, or more broadly, education [28].

3 SYSTEM & DESIGN

Python is the programming language which students playing the game-based learning system utilize to solve challenges. This choice was made because Python is understood worldwide to be an accessible language to learn due to its perspicuous syntax that places an emphasis on natural language and its general ease of use. Furthermore, the first-year introductory Computer Science course, as well as the extended introductory Computer Science course taught at the University of Cape Town (UCT) use Python as the introductory programming language of choice. Because play-testers and research study participants were sourced from UCT, Python became the obvious programming language of choice to feature in the game-based learning system.

Unity was used to implement the GBL system. Unity is real-time development software that is immensely popular as a video game development platform. The primary motivation for implementing the narrative game-based learning system using Unity instead of a game engine built in Python was because of the general lack of Python game engines that support 3D game design. Furthermore, Python game engines that do support 3D game design were deemed to be too light weight for the task given the time constraints. Unity was considered to be the best choice to make because of the host of tools and features it contains that means less time need to be spent creating a minimum viable product and it allows for more time to be spent developing substantial game mechanics, narrative elements and a fully-realized game world. Opting to use Unity introduced a new technical challenge to overcome due to the fact that Unity is written in the programming language C# (C-Sharp), a Python to C# interpreter had to be implemented. This interpreter take scripts written in Python, converts them to equivalent C# scripts and executes said C# scripts on various game objects defined within Unity. This challenge is covered in greater detail in its own subsection, 3.1.2.

Due to constraints on the amount of time play-testers would spend playing the game-based learning system, and in trying to appeal to the broadest possible range of play-testers, it was decided that a simple and comprehensive narrative was the best choice. This was done in the hopes of making the narrative elements within the GBL system as universally accessible as possible. In the narrative version of the game, a cut-scene is displayed in the beginning which shows an astronaut leaving his family behind on a journey to space. The rocket ship carrying the astronaut is suddenly damaged in a meteor shower and crash lands on an alien planet. After the cut-scene, the astronaut identifies that the internal computer system on the rocket ship has gone down and that they need to fix it to leave the planet and return to their family. The coding challenges are designed to seamlessly fit into the narrative of the story, as the scripts that the player writes become the code which makes the internal computer systems of the rocket ship function again. After the player completes the tasks they are assigned with, they see another cut-scene of the rocket ship, now working, flying away from the alien planet. This narrative was designed to be simple, yet effective and to contain elements of transportation and emotional connection. It was also designed to be modular, in the sense that it can be removed from the GBL system without rendering the system to be unplayable or confusing.

Players experience the game as the astronaut attempting to fix their ship. In the main scene, They have a third-person view of the astronaut and can navigate using the "WASD" keys on the keyboard. The camera can be panned around the astronaut using the mouse. To interact with objects in the game-world, the player presses the "E" key on their keyboard. While solving coding challenges, the player cannot see the astronaut character and they can no longer move or pan the camera. Instead, they are shown a graphical depiction of the coding challenge they are attempting to solve and are provided with a user interface panel on the right-hand side of the screen. This user interface panel functions as the "computer" into which the player writes scripts and runs code. The UI panel contains a text box with two toggle-able buttons labelled "In-scope Elements" and "Description". Clicking the "In-scope Elements" button updates the text box to show the pre-defined functions that the player has access to, how they are called, and what values they return. Clicking the "Description" button updates the text box to describe the tasks that the player needs to solve in order to complete the coding challenge. Below this text box is a user input field where scripts are written and processed. An example script is included in the beginning, to show the player how to interact with the in-scope elements provided to them through code. Below the user input field is another text box that displays simplified terminal output to the player. Finally, there are two buttons labelled "Run Code" and "Exit" beneath the terminal output box. Clicking the "Run Code" button interprets the player written code in the user input field from Python to C# and executes the written C# code on objects defined in Unity. Pressing the "Exit" button returns the player the to main scene.

3.1 Fundamental Programming Concepts

The 2013 ACM Software Development Fundamentals [47] identifies the essential competencies that an undergraduate computer scientist must develop. The competency "Fundamental Programming Concepts" encompasses basic concepts of programming languages that students must become proficient in. For the purposes of this research project, a subset of these fundamental programming concepts were selected to be taught in the narrative game-based learning system. These are listed below:

1. Variables and primitive data types
2. Expressions and assignments
3. Conditional and iterative control structures
   - if-statements
   - for-loops
4. Functions and parameter passing
   - players will not need to write their own functions but will need to call pre-defined functions.

The coding challenges within the GBL system have been constructed in such a way that there is a logical separation between the different fundamental programming concepts that are covered. There are three different programming scenes within the game, each scene ties into the overarching narrative of the game in a unique way and requires the player to write short scripts to solve minor coding challenges. The

"Electricity Scene" is the first coding challenge that the player undertakes. The player is required to utilize their understanding of variables, primitive data types, expressions, assignments and type casting in order to complete the challenge. In terms of the running narrative, this challenge is portrayed as the player attempting to fix the generator aboard the rocket ship and direct incoming power to the correct locations in order the get the electricity working again. The "Water Scene" is the next scene. The player needs to write a script for the ship's water pump that determines if incoming water is clean or not. If it is clean, it can be sent straight to the on-board water tank. If it is impure or too acidic, it needs to be sent to the purification tank first. This scene builds upon the players understanding of conditional expressions and if-statements to solve. In the final scene, the "Engine Scene", the engine is broken down into two sections. Each section has a number of 'chambers' that are responsible for generating energy. Section one requires the player to calculate the total energy produced by the section by iterating through each chamber in the section and calculating the additive total value. Section two requires the player to iterate through each chamber in the section, using a pre-defined function to determine if a given chamber is broken, and if said chamber is broken, fixing it using another pre-defined function. This section relies on the player having some knowledge of for-loops as well as slightly more advanced introductory Computer Science concepts such as calculating additive total values and effectively utilizing if-statements within for-loops.

There is a logical ordering to the scenes. They have been designed to increase in complexity as the player progresses, and they build upon concepts covered in previous scenes. The player has to solve the Electricity scene first, the Water scene second and finish with the Engine scene. This ordinal system with the scenes echoes into the narrative elements too. If the player tries to interact with a scene before they have completed the pre-requisite scene(s), a subtitle will pop up that ties the ordinal nature of the scenes into the story line. For example, if a player attempts to interact with the Water scene before completion of the Electricity scene, the subtitle "Looks like I need to get the electricity up and running again before I can fix the water system." appears.

The visual aspects of the coding challenges have also been considered. Screenshots of each scene are illustrated in Appendix B. For example, the visual elements that accompany the challenge described in the water scene provide a medium to convey to the player what they need to do to complete the scene in a way that is less abstract than the description of the scene.

3.2 Python to C# Interpreter

Since scripts in Unity are written in the language C#, and players will be writing Python code, it became necessary to implement a Python to C# interpreter that allowed a player to write short Python scripts in the game and have them be interpreted and subsequently executed in C#. IronPython [17] is an open-source implementation of the Python programming language which is tightly integrated with .NET (the framework upon which C# is built). Using IronPython, Python programs can integrate with applications written in other .NET programming languages, such as C#. IronPython has been used to interpret the user's Python code during runtime, which allows them to manipulate objects in the Unity environment. Objects in the Unity game-world that can be interacted with via code are attached with specific scripts that specify their name as well as the functions included in the script that the interpreter has access to. This is to prevent players from being able to call functions that induce a crash or undesirable game state and was implemented by utilizing the proxy class structural design pattern, called controllers, that allows the programmer to control which elements and functions of a class another class has access to. The drawback of this structural design pattern is that it introduces many more seemingly useless classes into the project, but for the implementation of project with relatively few classes like this one, it worked well.

3.3 Toggle-able Narrative

To effectively test the impact that narrative elements in game-based learning systems have on student engagement in the context of learning introductory programming, the game was designed in such a way that narrative elements could be toggled on or off. This is first displayed to the player when they open the game, they are taken to a main menu that has two buttons. The first button is labelled "play narrative mode" and the second is labelled "play non-narrative mode". To have the option of playing the game with or without narrative elements allows for user engagement metrics to be compared between the two ways of playing the game and allows for valuable insight into the efficacy of narrative elements as a tool for fostering student engagement.

In narrative mode, a cut-scene is shown before the player is placed into the game-world to provide meaningful context and create emotional attachment between the player and the character they play as. This is also used to establish a narrative reason for the player to complete the challenges. Once the game starts, subtitles are displayed at the bottom as a means of indicating the internal dialogue of the astronaut to the player. Different dialogue options show up when the player interacts with or completes challenges in the game world. The subtitles serve as a mechanism to draw the player in and provide them with a running story-line. Finally, the narrative mode contains special objects on the ship that can be interacted with by pressing the 'E' button on the player's keyboard. When interacted with, special dialogue options appear that function to appeal to the empathetic responses of the player by further fleshing out the nature of or relationships within the world as expressed through the narrative.

In non-narrative mode, There are no cut-scenes, subtitles or interact-able objects that do not directly relate to coding challenges. Note, non-narrative mode is not entirely devoid of narrative elements. Players may infer a non-verbal narrative based entirely on the props, setting and atmosphere of the game. Alongside that, upon completion of all the programming challenges, a cut-scene plays that shows a rocket ship leaving a barren planet. In narrative-mode, this cut-scene is accompanied by dialogue in the form of subtitles. In non-narrative mode, there are no subtitles in the cut-scene.

Minor narrative elements were kept in the non-narrative mode of game for a multitude of reasons. First and foremost, many elements of games bleed into each other. The visual element of games is appealing for reasons other than just contextualizing a narrative, yet, it is very difficult to entirely remove narrative aspects from the visual elements of games. As a player experiences the visual aspect of a game, they receive non-verbal information that leads them to automatically construct a contextual interpretation of the game. This contextual interpretation of the game serves as a rudimentary narrative in the absence of a strictly defined story line. Furthermore, the vast majority of game-based learning systems provide feedback to the player upon completion of tasks or challenges within the GBL system. This is an effective means of motivating the player to complete the tasks assigned to them [1, 36] and creates a satisfaction based reward-system for the player. In keeping these minor narrative elements in the non-narrative version of the game, we are avoiding potentially skewing our data by not considering the other aspects of games that may have an effect on student engagement that a strict narrative may introduce. For example, only including aesthetic, visual elements in the game in narrative mode means that if there is a change in the level of student engagement between narrative and non-narrative mode, it may be due to the effect of visual elements in the game and not the introduction of a narrative to the game-based learning system. Including minor narrative elements is therefore a means of mitigating the unseen variables that may only serve to skew the accuracy of the research.

3.4 System Architecture

There was strict emphasis placed on maintaining simplicity and clarity when it came to the software architecture of the GBL system. This strict emphasis was put in place for two reasons. One, the time constraint to get the GBL system into a place where it could be effectively play-tested was shorter than was ideal. This meant that the game-based learning system had to be as simple and modular as possible to ensure that bugs in the software could be quickly identified and fixed or cut out of the end product. Two, the entire system needed to be modular in order to reliably be able to play the game as intended in narrative or non-narrative mode. This meant that all narrative-related classes and constructs needed to be able to be removed from the game-based learning system without creating an unhealthy game state or introducing bugs.

The Singleton design pattern is a popular design pattern that is known to hurt a project in the long run in most cases. For this reason, it was avoided where possible. The only instance of it being implemented in the GBL system was to ensure that only one object that could play background music existed at a given point in time. The reason this design pattern worked in this situation is because the MusicHandler object was isolated from the rest of the codebase and therefore did not introduce dangerous coupling into the structure of the project.

In trying to avoid using Singletons, a public static class called CustomGameManager was implemented which functioned as a globally accessible object that stored multiple booleans that maintained the state of the game. For example, when narrative mode was selected from the main menu screen, CustomGameManager.NarrativeModeEnabled was set to true and then the main scene was loaded. This functioned in a similar manner to Unity's PlayerPrefs class, except it did not store data between sessions as there was no need to be able to save one's progress when playing the game.

SubtitleManager and SceneManager were the two major classes that handled core functionality within the game. Both of them functioned as the single wherein their assigned tasks could be handled. There is no way to create a subtitle without calling a function of the SubtitleManager and similarly, there is no way to move between scenes without calling a function of the SceneManager. These objects functioned to maintain modularity in the codebase and to allow for easier debugging throughout the project.

Unity's UnityEvent system was utilized as a built-in Observer design pattern to define 'listener' objects that listen for triggers before executing a block of code. For example, the WaterChecker script 'listens' for the event when the ship scene is loaded after the water scene has been completed, and when it is triggered, updates water objects in the scene to look like they are flowing through pipes.

4 METHODS & TESTING

4.1 Experiment Design

The narrative game-based learning system was designed in such a way that a full play-through of the entire game amounted to around 25 minutes on average. This was deemed to be an acceptable amount of time for a player to get a feel for the game-based system as a whole and to absorb the narrative elements in their full capacity without repelling potential research participants due to the promise of long time commitments. To ensure the game-play was kept to roughly 25 minutes, two current first year Computer Science students and one second year student were contacted and asked to gauge the difficulty and amount of time they would take to complete the given tasks within the game.

To gather quantitative information about a given play-testers experience of the game and their sense of engagement, a survey was designed in line with the standards described in the User Engagement Scale [30]. This approach towards measuring user engagement subdivides the notion of engagement into measurable sub-factors. These factors are; *Focused attention*, feeling absorbed into the interaction and losing a sense of time. *Endurability*, the overall success of the interaction and users' willingness to recommend an application to others or engage with it in future. *Felt involvement*, the sense of being "drawn in" and having fun. *Perceived usability*, a negative affect experienced as a result of the interaction and the degree of control and effort expended during the interaction. *Aesthetic appeal*, the attractiveness and visual appeal of the interface. The *Reward* sub-factor is used as a more robust measure that accounts for the conceptual overlap between *Endurability* and *Felt involvement*.

A short-form survey designed around specifications from the User Engagement Scale [30] was developed. This survey accounted for factors that have been identified as contextual artefacts that could potentially sway user results. In accounting for these factors, every participant answered the survey online and participant fatigue was mitigated for by limiting the play-testing sessions to 45 minutes total. The full survey as well as the sub-factor that each question influences are listed in Appendix A. Every question in the survey is coded on a five-point Likert scale ranging from 1 - Strongly Disagree to 5 - Strongly agree. There are, however, questions that are asked in the negative form, in which case, the response needs to be inverted. For example, giving a 2 as a response to the statement "I felt frustrated while playing the game." corresponds to a 4 as a response to the statement "I didn't feel frustrated while playing the game.".

To gather qualitative information about a given play-testers experience of the game, a brief conversation was conducted between the researcher and the research participant, after which the research participant sent a brief email to the researcher detailing the positives and negatives of their experience as well as their perceived sense of engagement.

After signing the informed consent form, each research participant was sent a table of possible slots in which they could sign up to play-test the GBL system. All research was conducted online via Microsoft Teams or Google Meet. Participants were sent a meeting link and a link to download the game approximately half an hour before the slot that the research participant had signed up for. This was done to guarantee that a student did not have the chance to play through the game before the meeting. Once the meeting started, participants were asked to share their screen to ensure that research participants were in fact playing the game for the allotted duration. The research participants were informed that the researcher would not be watching them for the whole duration of their play-testing, but would instead glance at the shared screen every two minutes to confirm that the narrative game-based learning system was still being played and that no game-breaking errors had occurred.

4.2 Ethical Issues

Ethical considerations with regards to the process of testing and developing the narrative game-based learning system have been accounted for. Students from the CSC1010H, CSC1011H and CSC1016S Computer Science courses at the University of Cape Town were contacted. No compensation was offered, all students undertook the exercise through willing participation. Some students that had taken one of these courses but were not majoring in Computer Science were also allowed to participate.

Students were asked to engage with the game-based learning system for around 45 minutes and then undertook a short survey that was designed in accordance with the User Engagement Scale[30], as explained above.

Furthermore, all students were informed of their precise role within the research and were provided with formal informed prior consent forms that needed to be signed before they a student was to be involved. Their confidentiality was also maintained as all data obtained was referenced without identifiers. For data gathering purposes, each student was identified by the order in which they signed up for a play-testing time slot. For example, the first student to play-test the game was assigned the ID; 'Student 1', and so on.

Taking student safety and the COVID-19 pandemic into account, there was absolutely no face-to-face interaction and the entire process of recruiting, sending the game-based learning system to the student and conducting the survey was done online.

The system is entirely comprised of either open source software or software and assets that have been licensed or bought.

Unity Personal Edition was used to make the narrative game-based learning system. Unity Personal Edition is free to use and share creations on if the individual using it makes less than USD 100'000 a year. We also made use of the Unity Asset store to purchase additional assets which operate under the Standard Unity Asset Store EULA. Any additional assets were created using Blender, a free and open source 3D creation software.

5 RESULTS & ANALYSIS

A total of seven research participants play-tested the game. Four of them played the narrative version and the remaining three played the non-narrative version of the game-based learning system. However, the data acquired from Student 5 - who did a narrative play-test was discarded as the student did not fill out the survey immediately upon completion of the play-test, send a qualitative opinion email or attempt to solve the coding challenges in earnest. This leaves three research participants who play-tested the narrative version of the game, and three who play-tested the non-narrative version of the game.

5.1 Non-Narrative Results

Focused Attention	3.667
Aesthetic Appeal	4.417
Perceived Usability	3.333
Reward	4.222
Overall Engagement	**3.910**

These results have a minimum value of 1 and a maximum value of 5. They are scores relating to the characteristics associated with user engagement described in section *4.1*.

It is clear to see from these results that the research participants felt the game-based learning system without narrative elements was a worthwhile experience. This can be gauged by the fact that the 'Reward' score was high considering the range of possible scores. This sentiment is echoed by the qualitative responses from research participants. Student 2 finished their qualitative review saying *" I really enjoyed it*[the game] *and the game environment!"*. Student 3 wrote *"I found playing this game to be a positive experience and I find that it did help reinforce my understanding from my Computer Science course"*. These responses, alongside the Reward score, indicate that even without narrative elements, students found the non-narrative version of the game to be a rewarding experience.

The game-based learning system seemed to contain a few blind spots when it came to the user's perceived feelings of control over the system and the amount of effort required to understand the task at hand. This is indicated by the relatively low Perceived Usability score of 3.333. Again, this sentiment echoes through into the qualitative responses, with Student 1 writing *" I found the game enticing, however I was a little confused in the beginning as to what I needed to do"* and Student 3 expressing that they felt the player movements and cursor panning was *"a little bit tough with the arrowkeys and the cursor"*.

Emotional Investment	3.333
Perceived Effectiveness as a Pedagogical Tool	4.667

Note, the above results are not associated with user engagement according to the User Engagement Scale, but they were deemed to be valuable questions to ask students in order to obtain a measurement of the extent to which they felt emotionally attached to the outcome of the story and if they felt the game-based learning system would be an effective educational tool to leverage in an introductory programming course. The emotional investment score being comparatively low indicates that not having explicit narrative elements present in the game-based learning system contributed to a lower Emotional Investment in the outcome of the story. This could lead to lower motivation to complete coding challenges.

5.2 Narrative Results

Focused Attention	4.333
Aesthetic Appeal	4.889
Perceived Usability	4.222
Reward	4.889
Overall Engagement	**4.583**

These results indicate that in every measure-able subfactor associated with user engagement, the narrative version of the game performed better than the non-narrative. While this is encouraging, it is also indicative that there are potentially hidden variables present in the narrative mode of the game that influence user engagement. This is covered in more detail in Discussion, *5.3*.

Emotional Investment	4.000
Perceived Effectiveness as a Pedagogical Tool	4.836

In line with the research hypothesis, a player's sense of Emotional Investment and Focused Attention both increased with the introduction of narrative elements to the game-based learning system. The higher Emotional Investment score indicates that narrative elements did in fact help to make the student more emotionally invested in the outcome of the game-based learning system. Likewise, the higher Focused Attention score indicates that the student was transported into the game-world to a greater extent in the narrative version of the game.

The narrative elements were a distraction.	1
The narrative elements made me more motivated to solve coding challenges.	4.667
The game's narrative elements increased my engagement.	4

These scores were specifically gathered from people that played the narrative version of the game. No comparative analysis can be conducted between the non-narrative group for these scores. However, the lowest possible value, 1, was obtained for the question related to how distracting students found the narrative elements. This indicates that narrative contributed little to a user's disengagement from the game-based learning system. High scores associated with narrative elements increasing engagement and motivation potentially indicate that students felt the narrative elements helped to increase both their motivation towards and engagement with the game-based learning system.

5.3 Discussion

The narrative version of the game saw an increased user engagement score for every possible sub-factor. Alongside this, the metrics for Emotional Investment and Perceived Effectiveness as a Pedagogical Tool both increased in the narrative version of the game. While these results bode well for the effect that narrative elements have on student engagement, they also indicate that there may be hidden factors that influence a students sense of engagement in the narrative mode of the game that are not narrative elements. This is most easily demonstrable with the Perceived Usability score. This score should not increase massively between versions of the game. Ideally, both the narrative and non-narrative versions of the game provide the user with the same degree of Perceived Usability. In this case, the narrative version of the game provided students with a clear next objective and a logical order in which to complete tasks that was not clearly described in the non-narrative version of the game. This, therefore, implies that it was not in fact the narrative elements of the game that increased the Perceived Usability but instead that fact that narrative elements provided pointers to the students as to what needed to be done next which the non-narrative mode failed to do.

Furthermore, the fact that Aesthetic Appeal score increased in the narrative version of the game could indicate two potential weak-spots of this research. Firstly, the introduction of narrative elements also included physical objects in the game world that a student could interact with. These elements were simply invisible in the non-narrative version of the game. These objects may have contributed to the aesthetic appeal of the game overall, leading to user's scoring the Aesthetic Appeal higher in the narrative version, even though the visual aspect of these objects was an unintended side effect of playing in narrative mode and is not related to the narrative of the game. Secondly, the User Engagement Scale may not contain a sub-factor that effectively accounts for a student's increased sense of engagement within the narrative version of the game, and as a result, they may have felt inclined to rate other aspects of the game higher than they would have had there been questions specifically related to the hidden sub-factor that contributed towards their increased sense of engagement. However, this result justifies the decision that was detailed in section *3.3* to keep visual and feedback elements in the non-narrative version of the game to mitigate the amount of increased perceived engagement in the narrative version of the GBL system that were not caused by the introduction of narrative elements alone.

Where the research was successful was in capturing the increased score of Emotional Investment and Focused Attention in the narrative version of the game. This was echoed by the qualitative responses of student who play-tested the narrative version of the GBL system. Student 4 said, of the narrative elements of the game, *" I feel like the narrative elements really helped engross me into it*[the game]". Student 6 felt that the *"Introductory story helped set the scene and stimulated emotional engagement"*. The literature indicates that narrative elements are effective tools to transport and draw emotion out of a player. To see that these research results are in accordance with the literature indicates that the narrative elements have been effective on student engagement in the same manner that they have been observed to be effective in the literature. This type of engagement causes intrinsic motivation to interact with a system which has been proven, in the context of Computer Science education, to result in a higher degree of knowledge acquisition than extrinsic motivation.

6 CONCLUSIONS AND FUTURE WORK

This research set out to take meaningful steps towards solving the problem of student engagement in introductory programming courses by testing the effect that narrative elements in game-based learning systems for teaching introductory programming have have on student engagement. This was tested by creating a game-based learning system which could be played with and without narrative elements, and running comparative experiments on both of the game versions.

It was found that narrative elements in the game-based system increased every user engagement sub-factor identified by the User Engagement Scale. Furthermore, the presence of narrative elements made students feel more emotionally invested in the outcome of the game and more likely to feel the game-based system would be an effective pedagogical tool. Finally, no students found the narrative elements to distract them away from the coding challenges or substance of the GBL system. Instead, they found narrative elements to make them feel more engaged and more motivated to interact with the system.

These results are entirely positive, however, it is impossible with the amount of information available to conclude definitively whether narrative elements in game-based learning systems have a positive effect on student engagement.

This research could be substantially furthered by conducting more considerable analysis into the sub-factors that influence student engagement and how best to measure them. Furthermore, this research project would have benefitted from closer analysis of the effect that second-rate factors introduced by the narrative version of the game would have and, where possible, introducing measures to nullify this effect. Finally, a larger scale experimental phase would have allowed for greatly improved statistical analysis of the effectiveness of narrative elements on student engagement. Had the sample of research participants been larger, more comprehensive non-parametric statistical tests such as the Mann-Whitney U Test to compare differences or the Chi-Squared test for independence could have been utilized to potentially produce statistically significant results indicating that narrative elements in game-based learning systems have a positive effect on student engagement.

This research project has revealed that narrative elements of game-based learning systems can potentially be leveraged as effective pedagogical tools to stimulate student engagement in introductory programming courses. Furthermore, this research indicates that a more detailed analysis of the role that narrative elements have in game-based learning systems is a worthy line of research to undertake and may produce meaningful results.

7 ACKNOWLEDGEMENTS

Thanks to my supervisor, Gary Stewart for his assistance with this research project. Thanks to my group partner, Cain Rademan for your continued support. Thanks to the student participants who helped to make this research a reality.

References

[1] Rula Al-Azawi, Fatma Al-Faliti, and Mazin Al-Blushi. Educational gamification vs. game based learning: Comparative study. *International Journal of Innovation, Management and Technology*, 7(4):132–136, 2016.

[2] Maxine Alterio and Janice McDrury. *Learning through storytelling in higher education: Using reflection and experience to improve learning*. Routledge, 2003.

[3] Jessica D Bayliss. Using games in introductory courses: tips from the trenches. In *Proceedings of the 40th ACM technical symposium on Computer science education*, pages 337–341, 2009.

[4] Geertje Bekebrede, HJG Warmelink, and IS Mayer. Reviewing the need for gaming in education to accommodate the net generation. *Computers & Education*, 57(2):1521–1529, 2011.

[5] Jens Bennedsen and Michael E Caspersen. Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2):32–36, 2007.

[6] Frank Biocca. The evolution of interactive media. *Narrative impact. Social and cognitive foundations*, pages 97–130, 2002.

[7] Ivo Blohm and Jan Marco Leimeister. Gamification. *Business & information systems engineering*, 5(4):275–278, 2013.

[8] Daniel Bormann and Tobias Greitemeyer. Immersed in virtual worlds and minds: effects of in-game storytelling on immersion, need satisfaction, and affective theory of mind. *Social Psychological and Personality Science*, 6(6):646–652, 2015.

[9] Elizabeth A Boyle, Thomas Hainey, Thomas M Connolly, Grant Gray, Jeffrey Earp, Michela Ott, Theodore Lim, Manuel Ninaus, Claudia Ribeiro, and João Pereira. An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers and education*, 94:178–192, 2016.

[10] Rick Busselle and Helena Bilandzic. Measuring narrative engagement. *Media Psychology*, 12(4):321–347, 2009.

[11] Antonella Carbonaro. Good practices to influence engagement and learning outcomes on a traditional introductory programming course. *Interactive Learning Environments*, 27(7):919–926, 2019.

[12] Chi-Cheng Chang, Chaoyun Liang, Pao-Nan Chou, and Guan-You Lin. Is game-based learning better in flow experience and various types of cognitive load than non-game-based learning? perspective from multimedia and media richness. *Computers in human behavior*, 71:218–227, 2017.

[13] Jonathan Cohen. Defining identification: A theoretical look at the identification of audiences with media characters. *Mass communication & society*, 4(3):245–264, 2001.

[14] Brianno D Coller and David J Shernoff. Video game-based education in mechanical engineering: A look at student engagement. *International Journal of Engineering Education*, 25(2):308, 2009.

[15] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15, 2011.

[16] Mohd. Elmagzoub Eltahir, Najeh Rajeh Alsalhi, Sami Al-Qatawneh, Hatem Ahmad AlQudah, and Mazan Jaradat. The impact of game-based learning (gbl) on students' motivation, engagement and academic performance on an arabic language grammar course in higher education. *Education and information technologies*, 26(3):3251–3278, 2021.

[17] .NET Foundation. Ironpython.

[18] Anabela Gomes and Antonio Mendes. A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, pages 1–8. IEEE, 2014.

[19] Anabela Gomes and António José Mendes. An environment to improve programming education. In *Proceedings of the 2007 international conference on Computer systems and technologies*, pages 1–6, 2007.

[20] Melanie C Green and Timothy C Brock. The role of transportation in the persuasiveness of public narratives. *Journal of personality and*

social psychology, 79(5):701, 2000.
[21] Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *2014 47th Hawaii international conference on system sciences*, pages 3025–3034. Ieee, 2014.
[22] Jeff Howard. *Quests: Design, theory, and history in games and narratives.* CRC Press, 2008.
[23] Maria-Blanca Ibanez, Angela Di-Serio, and Carlos Delgado-Kloos. Gamification for engaging computer science students in learning activities: A case study. *IEEE Transactions on learning technologies*, 7(3):291–301, 2014.
[24] Tony Jenkins. The motivation of students of programming. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education*, pages 53–56, 2001.
[25] Tony Jenkins. On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, volume 4, pages 53–58. Citeseer, 2002.
[26] Andrzej Marczewski. *Gamification: a simple introduction.* Andrzej Marczewski, 2013.
[27] Katie Larsen McClarty, Aline Orr, Peter M Frey, Robert P Dolan, Victoria Vassileva, and Aaron McVay. A literature review of gaming in education. *Gaming in education*, pages 1–35, 2012.
[28] Chris McKillop. Storytelling grows up: using storytelling as a reflective tool in higher education. In *Scottish Educational Research Association Conference, Perth, Scotland, 24–26 November*, 2005.
[29] Michael Miljanovic and Jeremy Bradbury. A review of serious games for programming. 11 2018.
[30] Heather L O'Brien, Paul Cairns, and Mark Hall. A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. *International Journal of Human-Computer Studies*, 112:28–39, 2018.
[31] Stamatios Papadakis. Evaluating a game-development approach to teach introductory programming concepts in secondary education. *International Journal of Technology Enhanced Learning*, 12(2):127–145, 2020.
[32] Trena M Paulus, Brian Horvitz, and Min Shi. 'isn't it just like our situation?'engagement and learning in an online story-based environment. *Educational Technology Research and Development*, 54(4):355–385, 2006.
[33] Arnold Pears, Stephen Seidman, Lauri Malmi, Linda Mannila, Elizabeth Adams, Jens Bennedsen, Marie Devlin, and James Paterson. A survey of literature on the teaching of introductory programming. *Working group reports on ITiCSE on Innovation and technology in computer science education*, pages 204–223, 2007.
[34] Nikolaos Pellas. Exploring interrelationships among high school students' engagement factors in introductory programming courses via a 3d multi-user serious game created in open sim. *J. UCS*, 20(12):1608–1628, 2014.
[35] Jan L Plass, Bruce D Homer, and Charles K Kinzer. Foundations of game-based learning. *Educational Psychologist*, 50(4):258–283, 2015.
[36] Marc Prensky. Digital game-based learning. *Computers in Entertainment (CIE)*, 1(1):21–21, 2003.
[37] Andrew K Przybylski, C Scott Rigby, and Richard M Ryan. A motivational model of video game engagement. *Review of general psychology*, 14(2):154–166, 2010.
[38] Meihua Qian and Karen R Clark. Game-based learning and 21st century skills: A review of recent research. *Computers in human behavior*, 63:50–58, 2016.
[39] Hua Qin, Pei-Luen Patrick Rau, and Gavriel Salvendy. Measuring player immersion in the computer game narrative. *Intl. Journal of Human–Computer Interaction*, 25(2):107–133, 2009.
[40] Daniel C Richardson, Nicole K Griffin, Lara Zaki, Auburn Stephenson, Jiachen Yan, Thomas Curry, Richard Noble, John Hogan, Jeremy I Skipper, and Joseph T Devlin. Measuring narrative engagement: The heart tells the story. *BioRxiv*, page 351148, 2018.
[41] Katie Salen, Katie Salen Tekinbaş, and Eric Zimmerman. *Rules of play: Game design fundamentals.* MIT press, 2004.
[42] Edward F Schneider, Annie Lang, Mija Shin, and Samuel D Bradley. Death with a story: How story impacts emotional, motivational, and physiological responses to first-person shooter video games. *Human communication research*, 30(3):361–375, 2004.
[43] Katie Seaborn and Deborah I Fels. Gamification in theory and action: A survey. *International Journal of human-computer studies*, 74:14–31, 2015.
[44] Judy Sheard and Dianne Hagan. Our failing students: a study of a repeat group. In *Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, pages 223–227, 1998.
[45] Gabriela Silva-Maceda, P David Arjona-Villicana, and F Edgar Castillo-Barrera. More time or better tools? a large-scale retrospective comparison of pedagogical approaches to teach programming. *ieee Transactions on Education*, 59(4):274–281, 2016.
[46] Robert H Sloan and Patrick Troy. Cs 0.5: a better approach to introductory computer science for majors. *ACM SIGCSE Bulletin*, 40(1):271–275, 2008.
[47] The Association for Information Systems (AIS) The Joint Task Force: Association for Computing Machinery (ACM) and IEEE Computer Society. Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science, December 2013.
[48] Anastasios Theodoropoulos, Angeliki Antoniou, and George Lepouras. How do different cognitive styles affect learning programming? insights from a game-based approach in greek schools. *ACM Transactions on Computing Education (TOCE)*, 17(1):1–25, 2016.
[49] Lynda Thomas, Mark Ratcliffe, John Woodbury, and Emma Jarman. Learning styles and performance in the introductory programming sequence. *ACM SIGCSE Bulletin*, 34(1):33–37, 2002.
[50] Christopher Watson and Frederick WB Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 39–44, 2014.
[51] Susan Wiedenbeck, Deborah Labelle, and Vennila NR Kain. Factors affecting course outcomes in introductory programming. In *PPIG*, page 11, 2004.
[52] Mirac Yallihep and Birgul Kutlu. Mobile serious games: Effects on students' understanding of programming concepts and attitudes towards information technology. *Education and information technologies*, 25(2):1237–1254, 2020.
[53] Michael F Young, Stephen Slota, Andrew B Cutter, Gerard Jalette, Greg Mullin, Benedict Lai, Zeus Simeoni, Matthew Tran, and Mariya Yukhymenko. Our princess is in another castle: A review of trends in serious gaming for education. *Review of educational research*, 82(1):61–89, 2012.
[54] Dan Zhao, Cristina Hava Muntean, Adriana E Chis, and Gabriel-Miro Muntean. Learner attitude, educational background, and gender influence on knowledge gain in a serious games-enhanced programming course. *IEEE transactions on education*, pages 1–9, 2021.
[55] Rolf A Zwaan, Mark C Langston, and Arthur C Graesser. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological science*, 6(5):292–297, 1995.

A APPENDIX A - Survey questions and Sub-factors

I would like to receive feedback about the results of the study. If you say yes, you will be sent a copy of the published research paper, and a summary of the main results/conclusions.	N/A
I lost myself in the game.	Focused Attention
This would be a valuable experience for first year Computer Science students.	N/A
This game aided my understanding of introductory Computer Science concepts.	N/A
I felt emotionally invested in the game.	Emotional Investment
The time I spent playing the game just slipped away.	Focused Attention
I was absorbed in this experience.	Focused Attention
I felt frustrated while playing the game.	Perceived Usability
I found this educational game confusing to use.	Perceived Usability
Using this educational game was taxing.	Perceived Usability
This educational game was attractive.	Aesthetic Appeal
This educational game was aesthetically pleasing.	Aesthetic Appeal
This educational game appealed to my senses.	Aesthetic Appeal
Playing the game was worthwhile.	Reward
My experience was rewarding.	Reward
I felt interested in this experience.	Reward
The narrative elements made me more motivated to solve coding challenges.	Only asked in Narrative Version
The game's narrative elements increased my engagement.	Only asked in Narrative Version
The narrative elements were a distraction.	Only asked in Narrative Version

B APPENDIX B - Game Screenshots

Figure 1. Main menu scene.

Figure 2. An example image from the first cut-scene.

Figure 3. Ship scene - tutorial window.

Figure 4. Ship scene - in non-narrative mode. No narrative interactable objects and no subtitles.

Figure 5. Ship scene - in narrative mode. Subtitles at the bottom, interactable objects highlighted yellow.

Figure 6. Ship scene - in narrative mode. Interacting with the barrel.

Figure 7. Electricity scene - a tutorial window.

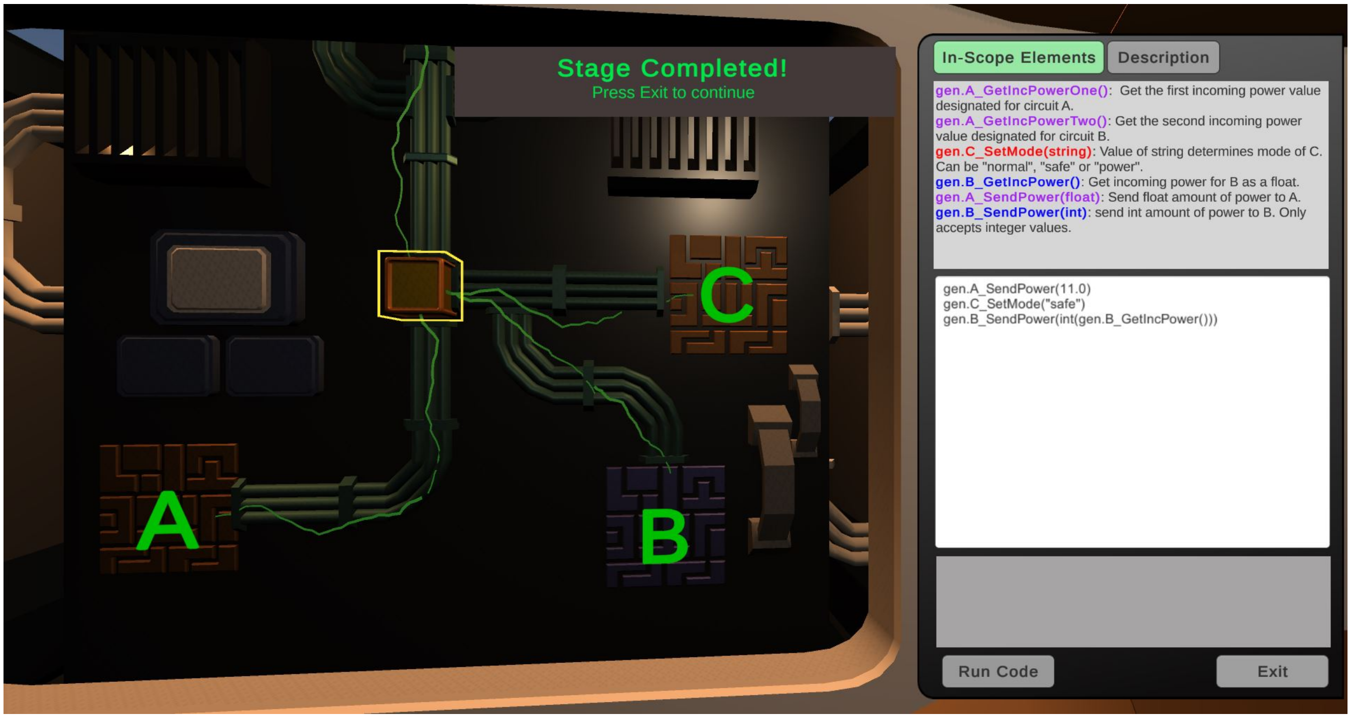

Figure 8. Electricity scene - The completed scene. Note the code in the right-hand panel.

Figure 9. Water scene - in narrative mode. Subtitles help to get the student to emotionally engage with the task at hand.

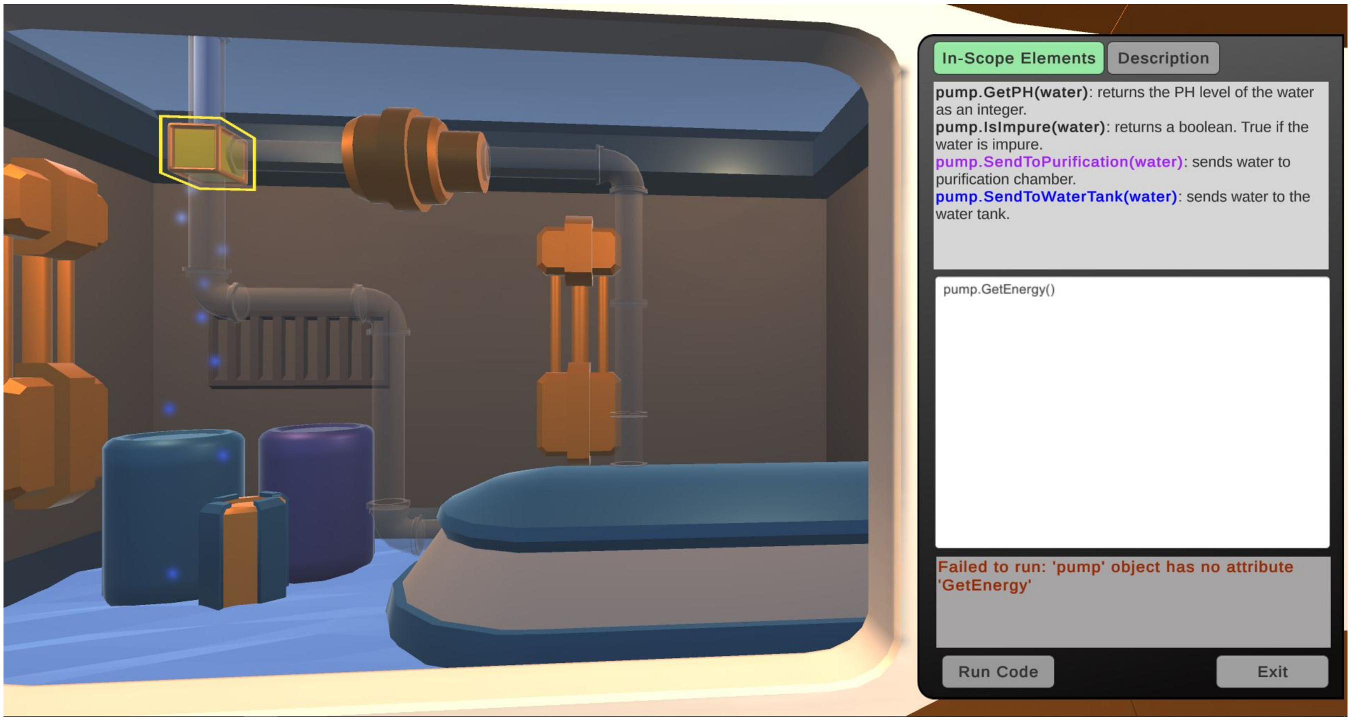

Figure 10. Water scene - An example of error feedback being shown to the player.

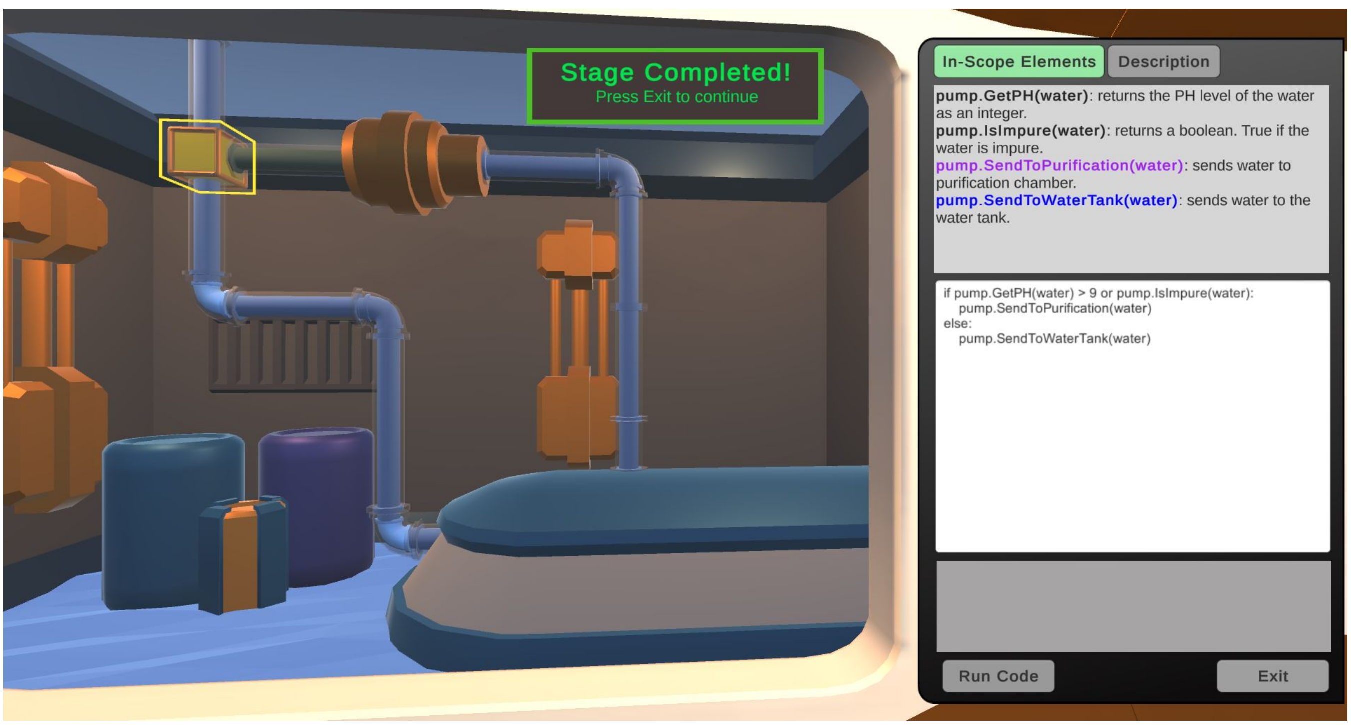

Figure 11. Water scene - A working solution to the water scene.

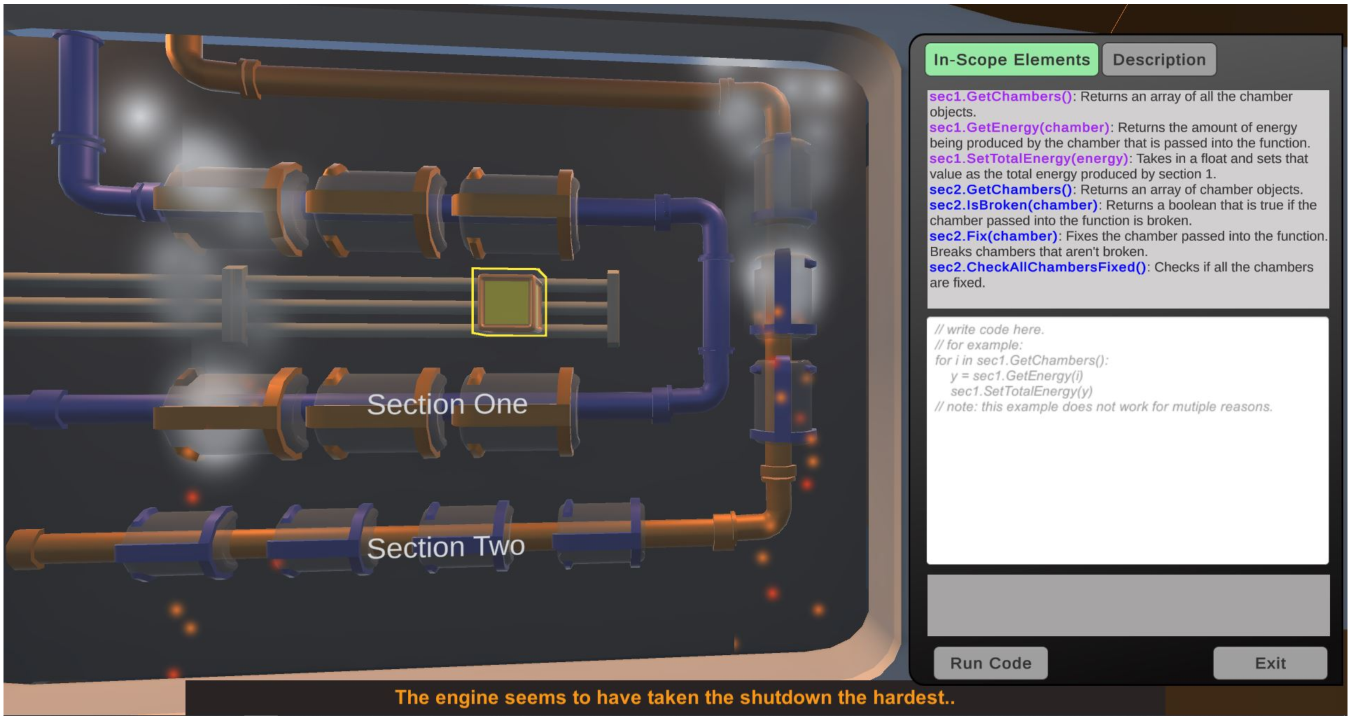

Figure 12. Engine scene - In narrative mode.

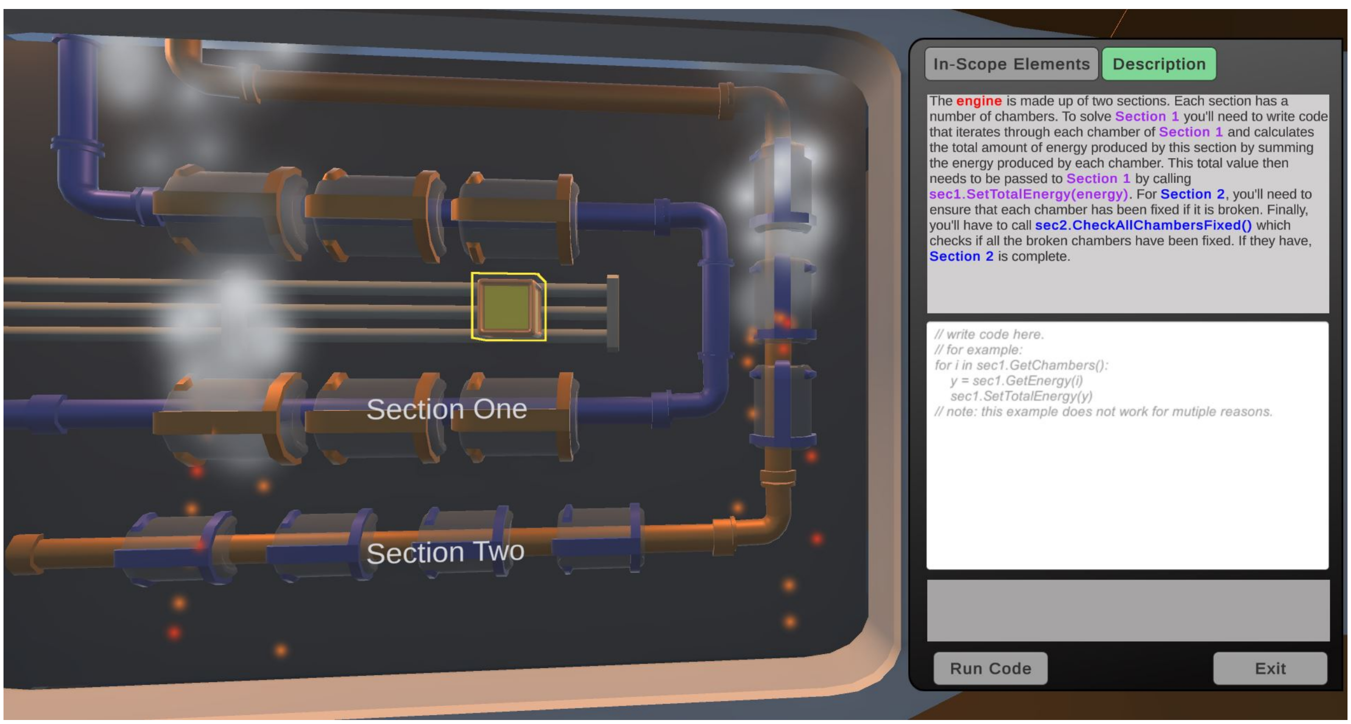

Figure 13. Engine scene - Showing the Description. This details what needs to be done to complete the scene.

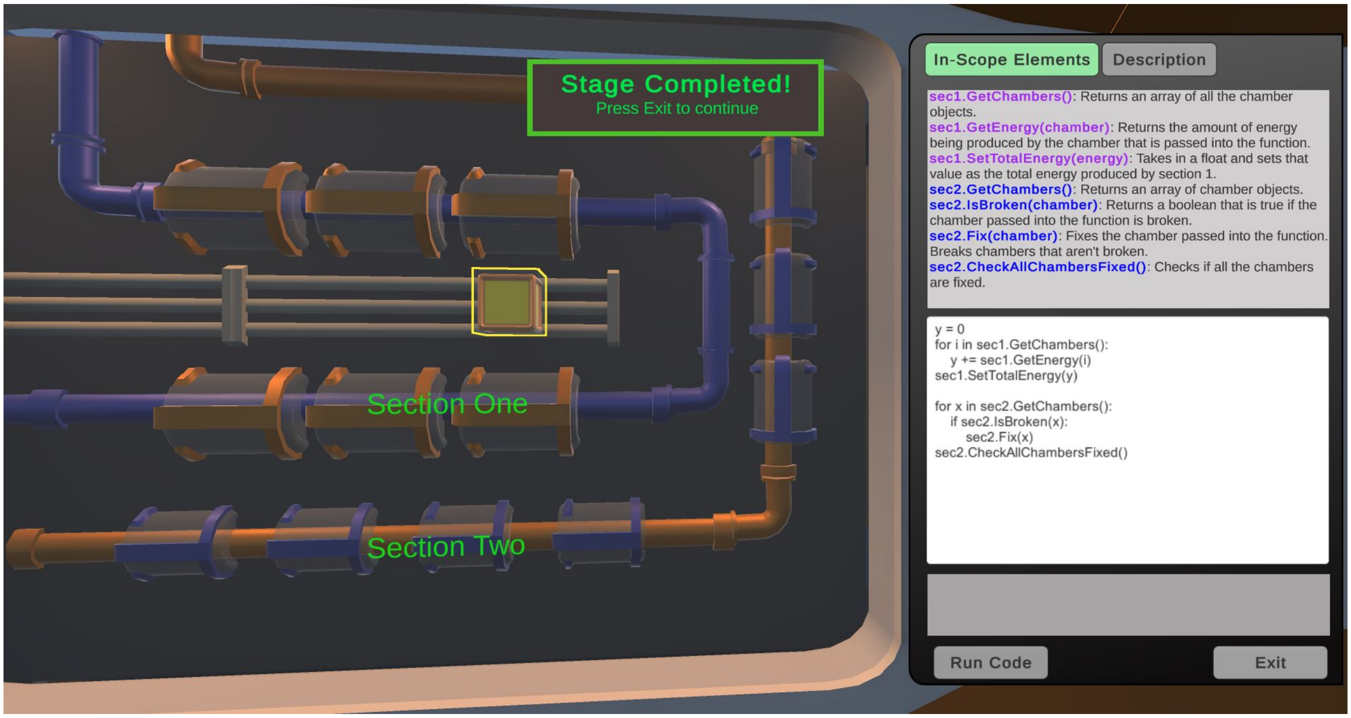

Figure 14. Engine scene - A working solution.

Figure 15. Final cut-scene - In narrative mode.