Investigating dynamic terrain as a novel mechanic to solve puzzles in games within a real-time physics simulation context

UNIVERSITY OF
LINCOLN

Mr Ashley Cromack
CRO12450621

Project Supervisor: Dr. Grzegorz Cielniak

MComp Games Computing

University of Lincoln
School of Computer Science

ACKNOWLEDGEMENTS

I would like to thank the following, for without their help this project would not have been possible.

Firstly, I'd like to thank my project supervisor Dr Grzegorz Cielniak, for offering continual help and guidance along the course of this endeavour, helping to structure and

I'd also like to thank my family for their unending support throughout my university career, never once has their belief in my abilities faltered and that has helped assure my confidence and kept me going throughout the years.

Finally, I'd also like to extend my thanks to Miss Millie Williams. Her love and support helped to keep me focused on delivering work to the best of my abilities.

Contents

Abstract

Modern video games tend to rely heavily on the work of their physics engines, software that helps to compute components such as collision detection, particle systems and, more recently, terrain deformation. However, terrain deformation typically hasn't seen much application as a core component of commercial games, so this project aimed to investigate the use of this dynamic terrain modelling to solve puzzles in a video game. This project explores the development of this game using the *Unity* game engine, with its inbuilt implementation of the *PhysX* physics middleware, documenting how players responded to this novel application of terrain deformation as a puzzle solving mechanic. From a study involving 8 people, it was discovered that they enjoyed the deformable terrain mechanics involved in the game but felt that they could've been utilised in more challenging puzzles and that they would've preferred more freedom to play about with what the mechanics can do.

Background

As computational resources available continue to grow, games developers set their aims on implementing more facets of the physical world into their creations to heighten realism, this is where the idea of reproducing real-world physics into games came about. The usage of rudimentary physics simulation in games can even be dated all the way back to *Pong* with its simulation of a ball bouncing off walls and paddles, while this may seem incredibly simplistic by today's standards it laid the groundwork for the vastly more complex systems we have now.

From here, the demand for more elaborate physics simulations in games grew, this lead to the creation of dedicated physics middleware, engines devoted to the accurate and realistic simulation of physics in real-time. As of today, the most widely used and richly featured physics engines in the games industry are Microsoft's *Havok* and Nvidia's *PhysX*, *Havok* being the most popular choice among developers with over 400 games using it in some capacity (Havok, 2011). However, most modern physics engines all feature similar physics techniques that developers can then implement in their games, these can include:

- Rigid Bodies: A simulation of how solid objects are structured, which can then have Newtonian mechanics applied to them to model their motion.
- Collision Detection: A system of calculating whether two objects have collided (typically using a bounding volume) and determining an appropriate response.
- Ragdolls: A way of procedurally animating characters within a game to give them realistic movement and interact with the environment in a natural way.
- Particle Systems: A controllable set of individual physical particles that react to external forces and, in some instances, forces between each other. These are used to model things such as water, smoke, cloth and rain.
- Deformable Bodies: Giving objects the ability to either shatter, destruct or deform.

This concept of deformable bodies presents a great many possibilities in how a game world can be manipulated by the actions of the player as, while some methods to model bodies of this nature can be rather computationally expensive, it creates a much more accurate depiction of how objects behave in the real world.

The concept of presenting a puzzle to the player, which they then must solve with the mechanics available to them, has been a staple of video games for decades. Whether the puzzles define the core of the game, such as titles like *The Witness* (Thekla Inc., 2016), or they form small challenges along the way like in the *Uncharted* games (Naughty Dog, 2007), games developers are always searching for new and unique ways of testing a player's problem-solving abilities.

The original aims and objectives from the project proposal were as follows:

- Aim:
  To experiment with the concept of deformable terrain, using a real-time physics simulation context, as a novel mechanic to solve puzzles in a game. The impact of this mechanic in gameplay, how it affects a player's perception of puzzles and how they can solve these puzzles using the tools at their disposal.
- Objectives
  - To investigate and experiment with the current implementations of physics systems and middleware to find a suitable basis to build a real-time solution for terrain deformation, such as *PhysX* and *Bullet*. Additionally, to explore ways to implement this system into a game, potentially through existing games engines like *Unity* and *Unreal Engine*.
  - The game should go through an extensive design and prototyping process to ensure that it delivers an enjoyable and coherent experience for the players, with focus on conveying the mechanics and how they operate clearly.
  - To successfully develop the game into a functioning prototype, demonstrating the implementation of terrain deformation that responds in an appropriate real-time context to the player's interactions with it. The game will be designed with this player-controlled terrain deformation in mind as the primary tool for solving puzzles.
  - To evaluate the effect that using dynamic terrain as a game mechanic has on the gameplay experience and how it influences their approach to puzzle-solving. To do this the artefact will need to be presented to a group of play-testers to investigate what their experience with the concept was like, what components they enjoyed, and which features they gravitated towards, along with more quantitative data from logging their interactions with the game.

These aims & objectives provided a guideline for key areas that needed to be investigated for the outset of this project:

- Physics Simulation
    - As the focal point of this project, it is vital to explore the existing academic investigations into the subject of how to simulate physics in real-time, the various ways in which this can be achieved, the benefits and drawbacks of these implementations and then finding one that appropriately fits the scope of this project.

- Game Design
    - To justify the approaches made with regards to the design of the game produced for this project, existing literature that provides a framework of how to approach the design stage should be explored, whilst also being sure to consider the focus that should be placed upon the deformable terrain aspect.

- Puzzles in Games
    - As this project focuses not only on the implementation of deformable terrain but also how it can be applied to puzzles, it is key to research the techniques employed in designing these puzzles, presenting them to the player and how to place a focus on the deformable terrain mechanics the player will be using to solve them.

- Player Experience
    - It is important to consider the ways in which the artefact produced for this project would be evaluated regarding the aims and objectives laid out above. Therefore, appropriate research methods for evaluating the response of players to the various aspects of a game will need to be investigated, along with determining their overall player experience.

Literature Review

Following on from the aims and objectives laid out prior, and the subsequent areas highlighted for investigation, the following academic literature has been explored and evaluated in the relevant areas.

Physics Simulation

Due to the heavy focus that this project places on the area of real-time physics simulation, it was vital to investigate existing academic applications of physics systems, what they are used for, how they are implemented, and any other important academic discoveries made in this field.

An important first step in exploring the domain of physics simulations, especially their implementation in video games, is to gain an appreciation for the background on how these systems have changed over the years. In Árnason's exploration of the evolution of physics in gaming (Árnason, 2008) he found that, while physics have played some part in games since their inception, it is not until the advent of more realistic simulations that they have come to the forefront of the experience. Through this investigation it has become clear that these complex physics engines are now integral to modern video games, as they provide key functionality such as collision detection and several types of body modelling (rigid, soft and deformable). However, due to the rapid advancements in GPU technology, the concerns that this paper raises in terms of the computational cost of physics simulations are now somewhat alleviated. While the most accurate simulations of complex dynamics are still very difficult to simulate in real-time, the approximations that most video games now use are handled easily by the GPU (especially given GPU producer *Nvidia*'s assimilation of *Ageia* and the *PhysX* engine). This paper therefore grants a useful reminder of how far real-time simulations of physics have come in such a relatively short time, to the point where they are now integral to the functionality of many games, framing the work of this project as a continuation into the exploration of how physics in games can continue to evolve.

Many of the more prevalent studies, that not only investigate real-time physics simulations in general but place a focus on how to model deformable terrain, tend to take a serious approach in terms of their application. For instance, the modelling of soil deformation in real-time to use in virtual reality training simulators for tools such as bulldozers and excavators, where a sufficient level of accuracy is required to deliver a reliable training experience for the users (Holz et al., 2009). This study places the focus on implementing complex equations and calculations to deliver the most realistic behaviour of soil deformation, featuring both erosion and compaction. The issue with studies of this nature is that the onus is on this computationally expensive accuracy, which is not something that could be afforded in a project such as this where there are limited resources available, both in terms of hardware and time. These types of papers do still hold some value for this project however, as they present a good idea of how terrain deformation could be approached but with some simplification changes to the modelling to alleviate some of the computational expense.

It is also key to investigate potential methods of implementing deformable terrain in a real-time context, one academic paper explored this through the visualization of tire tracks on large scale dynamic terrain (Zhang et al., 2010). In this paper, the academics took the approach of using a GPU-based terrain deformation algorithm to compute the alterations caused to the terrain as a result of a vehicle in the scene. This implementation utilised a custom algorithm in conjunction with heightmaps, passing this to GLSL shaders and rendering the scene using OpenGL. Having such a specific implementation allowed them to achieve a very high frame-rate for the simulation, making it suitable for a game application, which could prove useful for consideration of what tools to use for the development portion of this project. However, it may be important to consider the other aspects of the game design that will be required in the artefact's development other than this purely physics simulation-based approach.

In *Alternative Trajectories* (McKeown, 2016) McKeown analyses the impact that the physics engine has on a first-person shooter videogame, in this case *Call of Duty: Ghosts*. With this paper he outlines how the game prioritises its realism through a fairly accurate approximation of Newtonian physics, shaping the players' actions through its implementation. The noted functionality of this game's physics simulation includes a set of complex physical effects, such as smoke, where the effects respond to the player and other objects in the scene. However, the key focus was on how the game's physics interprets gravity, especially regarding how players can adapt their gameplay techniques to take advantage of the systems available to them to create novel gameplay moments (such as throwing a knife into the air, anticipating its arc of movement due to gravity and aiming to eliminate an enemy player). When considering the goals of this project, the most important takeaway from this paper is what drives players to experiment with physics systems in this way and how we can foster this gameplay behaviour. It appears that the best way to achieve this is to make the behaviour of the game systems explicit to the player, ensuring they understand exactly how they behave and thereby allowing them to take matters into their own hands when playing around with them.

Another important consideration to be made with physics simulations is how they can be understood by the people playing it, as the simulation needs to make coherent sense with a player's personal understanding of how physics works, a subject that was explored by academics looking into cognitive science (Ullman et al., 2017). Here, the academics explored the parallels between the features of physics engines and how humans represent physics in their head, placing a focus on the intuitive physics of infants and how this develops over time. Throughout the course of their study, they outline the key features of game physics engines and compared this with how people interpret the physical world around them, noting that there is a convergence in the need to overcome resource challenges in both instances and finally offers new hypotheses as to how intuitive physics could be used in the development of artificial intelligence. This study provides useful considerations in terms of how best to structure and present the physics systems at play in the artefact in this project, ensuring that they make immediate sense to the people playing it.

One of the most valuable applications of physics systems is their ability to aid the learning of students, as it provides a safe and infinitely variable way of experimenting. These physics simulations, especially those that were originally developed for use in commercial video games, are typically richly featured and can be used to model typical physics experiments, which was the focus of a 2008 paper (Price, 2008). In this study, the physics engine of *Unreal Tournament 2004* was utilised to demonstrate various forms of physics experiments, such as the interference of waves and parabolic motion due to gravity, this was motivated by a desire to encourage younger students to learn and develop an interest in physics. Through extensive experimentation with both students and their teachers, they found that both parties found it straightforward to construct experiments using the game's physics engine, the qualitative experiments developed using the software had a definite impact on the students' learning experience and that the theory of 'concept maps' provided a valuable design methodology. This provides a valuable knowledge base for this project, as it demonstrates an intuitive understanding that users have when interacting with a virtual environment that successfully emulates real-world physics concepts, such as gravity. Knowing this ensures that, so long as the physics system and related mechanics within this project's artefact fit with generally understood concepts in physics, players should confident in experimenting with the tools and understanding what impact will be made on the game world.

Another way in which physics concepts can be simulated to assist teaching is through their application within a serious game setting, where the educational content is fed through the gameplay, elevating the software beyond pure entertainment. For academics, this means studying the impact of these serious games on a student's learning, with one such study looking at teaching high school physics (Stege et al., 2012). In this work, the academics placed their focus on the effect of serious games on teaching electrical engineering to high school students, comparing this to those taught via text. They evaluated this through a phase of testing where, after students had either played the game or read the text, they would be tasked with solving assignments, which was then followed by a survey. From their results, the researchers found that the use of a serious game had a positive impact on the learning of male students compared to female students, though they found that female students who played the game felt less motivated than those who read the text. The reasoning for this was left undetermined, but they concluded by noting that serious games have the potential to be a more effective education tool than textbooks, particularly for male students. While the game being produced in this project is not of a serious nature, the ability to successfully convey physics behaviour to the players is crucial for understanding the mechanics of the game. It may therefore be useful to note the ways that the serious game in this study attempts to relay knowledge of physics to the players, as this could prove beneficial for the game design portion of this project.

Game Design

Due to this project's focus on creating a puzzle game that utilises deformable terrain as a mechanic, it was imperative to investigate the theory of game design, looking at the core components of games and established frameworks for their design.

When considering how to approach designing a game, it can be valuable to make use of pre-existing frameworks produced and reviewed by academics, with one of the most ubiquitous being the MDA Framework (Hunicke et al., 2004). This framework places an emphasis on both how the player and the designer interact with the game, be it from alternative perspectives where the designer focuses primarily on the mechanics whereas the player focuses on the aesthetics. This framework provides a good basis for how to approach game design, as it serves to remind the designer that players will first be drawn in at the surface level to the game before they delve deeper into how the game works. By having these abstracted layers, the MDA framework helps to tackle iterative design, allowing the designer to tweak certain aspects of the game to obtain the desired player experience, this therefore could make MDA a useful tool for the designing process of this project.

Another important consideration in the design of games is how to link all the various software that makes it up and presenting it as a cohesive whole to the player, this process has even seen the development of a framework in this academic field (Catanese et al., 2011). In this paper the researchers combined multiple cutting-edge games middleware, including the *PhysX* physics engine, and integrated them together using *Direct3D* and *OpenGL* for graphics. The focus with this study was on how to produce a dynamic 3D environment that features a greater degree of realism through the use of these various middleware technologies, aiming to simulate how the objects in the world behave and how they can interact with each other. While the focus on delivering a wholly realistic and complex world may not in itself be vital for this project, the techniques through which they successfully allow these tools to work together may be worthwhile to consider, as it may help to ease development later in the implementation process.

Puzzles in Games

The primary component to the game produced for this project's artefact, other than the terrain deformation, is the production of puzzles that the players can use the game mechanics at their disposal to solve. It therefore seemed necessary to examine the existing material regarding the design and importance of puzzles in games.

Much like with traditional game design that aims to structure the entire game, there are also many design principles that go into the production of both puzzles in games and puzzle games themselves, this being something that applies to real-world puzzles and digital puzzles alike. One paper investigated these principles of puzzle design (Zhou and Wu, 2012), particularly looking at the integration of educational content in these games and how they can be used to aid learning. They broke the requirements of these puzzles down into aspects of content, manifestations, human-computer interactions and feedback, then outlining the principles that need to be followed in these different categories. While this paper takes an overtly serious approach due to its connection to aiding education, the design process they lay out still has a great relevance to the desired aims of the puzzles produced in this project, it may therefore prove useful to refer back to this study during the design process of the game's development to aid the structure of the puzzle design aspect.

One study looked at how puzzle games can be thought of as a metaphor for computational thinking (Law, 2016), where people go through a process of implementing sequence, selection and iteration. Throughout this paper, Law explores the links between puzzle-based learning and the concepts of computational thinking, making a case that puzzle games could prove to be a valuable tool for teaching programming, as they provide an environment for developing the players' problem-solving skills. From here he then gives examples of existing educational and commercial games that foster these problem-solving abilities, with the educational games delivering a more overt demonstration of computational processes whereas the commercial games have these processes occurring in the background. The commercial games chosen here, namely *Portal*, *The Talos Principle* and *World of Goo*, provide valuable examples of how both physics and puzzle solving can be united to creating interesting and valuable game experiences. This paper therefore provides useful reference material for existing implementations of puzzles and physics in games for consideration of how to design the artefact for this project, while also providing a framework of how to consider the design of puzzles in terms of computational problem solving.

Another paper looked at how typical puzzle design doesn't go far beyond visual cues, taking advantage of this to experiment with new puzzle challenges using other sensory information, such as through sound and vibrations (Carvalho, Duarte and Carriço, 2012). From the results they obtained, they discovered that players tended to solve the most prominent areas of the puzzle first and would then leave the more abstract regions to the end, this proved to be true regardless of the interaction modality used, be it audial or visual. This could prove to be a valuable consideration when designing puzzles, to provide the players with a more prominent problem to solve first before they can tackle the more challenging aspects, which could then be observed as people play the game to see if they tend towards this approach.

One of the key components to many puzzle games is that there are typically many different techniques that the player can employ to solve the puzzle, this can provide a fascinating research point as it can help us to understand how players learn skills in games and then apply this to solve problems. In a paper focused on these puzzle-solving tactics (Vahdat, 2016), the academics applied Learning Analytics in puzzle games to explore the players' approach to puzzles and how they applied skills that they learnt in the game. Through their research process, the academics presented a three-step analysis to retrieve the puzzle-solving tactics of players from data, allowing them to identify tactics not considered by the game designer. Their Learning Analytics approach proved successful as they discovered two main successful tactics, so this could be asset in evaluating the techniques that players employ in playing the artefact of this project, should sufficient data be collected, though it presents a good example of puzzle game study design regardless.

Player Experience

When it comes to the evaluation side of this project, it became swiftly clear that devising an entirely original system of questionnaires and interviews would prove to be somewhat of an insurmountable task given the time and resource constraints of this endeavour. However, there already exists a vast range of academically ratified games evaluation materials, therefore this required exploration to determine the best fit for this project.

Due to the proliferation of player experience research and the development of surveys to monitor this information, it is therefore paramount to find a way of determining which of these standardised questionnaires are most appropriate for the research being carried out. In an evaluation of the currently available methodologies for measuring player engagement (Nordin, et al., 2014), academics found that there is a certain degree of overlap between many of the existing questionnaires in what components of user engagement they are measuring. Knowing this, it is therefore important to ensure that the survey used in the evaluation of this artefact fulfils all the appropriate criteria and that no unnecessary overlap is created in the event of multiple different surveys being used concurrently.

One such evaluation strategy comes in the form of the *Game Engagement Questionnaire* (Brockmyer et al., 2009), a way in which players' engagement in a video game can be measured through the application of a survey. In the development of this survey, the academics highlighted the key components of presence, flow, absorption and dissociation, which define key states of being while a player is engaged in a game. Through their development and evaluation, the academics found that the Game Engagement Questionnaire provided a psychometrically strong measure of the player's engagement levels while playing a game, which they highlight as being invaluable for examining factors for the negative impact of games. While in this project we won't be investigating this impact, the measurement of player engagement that this survey affords could prove to be useful in evaluating this project's artefact.

Another example of a standardised post-game survey can be found in the *Game Experience Questionnaire* (IJsselsteijn et al., 2013), a survey that utilises a modular structure that can be adapted depending on the evaluation requirements of the study being conducted. This questionnaire presents a straightforward methodology for administering a set of questions designed to learn about certain aspects of the players' experience, such as immersion, challenge, flow and competence. The modular nature of this questionnaire makes very suitable for a project of this scale, as it allows the questions to be focused and speeds up the evaluation process.

It's also valuable to investigate how established research studies carried out their evaluation in practice, one such study investigated the effects that the game *Portal* had on players' physics intuitions and spatial cognition skills (Adams, Pilegard and Mayer, 2016), resulting from the Newtonian physics simulation in the game. In this study, participants were measured on their performance through tests on their retention of physics knowledge that they may have gathered from their experience in the games they were asked to play. While they found no concrete evidence that playing these games facilitated the participants' physics learning, this study still provides a good example of how to conduct a games research study in terms of structure and data collection.

Another important realm of study to research was that of how to analyse the data that would be collected in the evaluation phase of this project, especially given the elicitation of qualitative data in this stage. One such approach to analysing this type of data is 'Thematic Analysis', a form of qualitative data analysis that focusses on identifying, examining and denoting patterns within the data; this technique typically finds application in the fields of psychology, but it can be applied in a vast array of fields. The problem with thematic analysis is that it is typically poorly laid out, leading to confusion over how to properly apply it, this is where work in the field of psychology comes in (Braun and Clarke, 2006). In this paper, Braun and Clarke outlines the key aspects of thematic analysis, generating a table of different phases and a 15-point checklist of how to determine good analysis. This therefore will provide a vital tool for carrying out the analysis side of the research further in the project, as qualitative data is likely to be key in evaluating the success of this project in its aims and objectives.

Methodology

Project Management

The paramount factors when considering how to manage a project of this type is the limited resources available, both in terms of time and manpower, as it is entirely the work of an individual over the course of just a few months. Therefore, it was vital to break down the project into distinct and measurable components to more appropriately structure the process. As this project is comprised of both artefact development and evaluation through research, the key components highlighted were: Background Research, Designing, Development, Artefact Evaluation and Documentation.

Regarding how best to manage a project, a range of different management techniques were investigated to determine which of the existing methodologies would best suit the needs of this project. Chief among these strategies is that of PRINCE2, the UK government standard for managing information systems projects. Upon investigation of this methodology however, it soon became clear that it would be inappropriate for a project of this scale. The reasoning for this is that, while there claims to be a degree of scalability to the PRINCE2 methodology, the amount of paperwork it demands would eat up a considerable amount of time, which would raise an issue with the strict time limitations of this project. Despite this, this methodology still proved to be useful to investigate, especially with regards to context of how larger scale software development projects may operate in the UK technology sector.

As investigated in a recent paper comparing the various software development methodologies (Shaydulin and Sybrandt, 2017), many organisations engaging in software development are exploring the use of agile methods in their development. They deemed that these agile methodologies, while they do not tend to scale well for larger projects, for smaller scopes and teams they can be more readily employed. However, due to the inherent focus on customer needs and the ability to implement many different features, it was deemed that an entirely agile approach would not be applicable for this project as it is primarily focused on the delivery of a specific artefact that will not change and any feature-creep could harm the time management of the project. This paper does also explore a more traditional development method, that remains one of the more widely used, the waterfall approach. This methodology was chosen for the management of this project as the plan for it works in a sequential manner, with the requirements of the game being outlined at the beginning, then followed by the design stage, which is then implemented in code and tested to ensure that the implementation works as intended. This helps to keep the focus on completing a layout of tasks that are each dependent on the task before them being completed, an important consideration especially when it comes to the research component of this project as it will require a fully functional artefact for the study to be carried out.

From the outset of the project, a range of tasks were discerned that would comprise the work required to carry out this project in terms of software development, research and report writing. Clearer versions of the below Gantt charts, along with the related list of tasks that comprise the Y axis and the dates denoted by the X axis, can be found in Appendix 1.

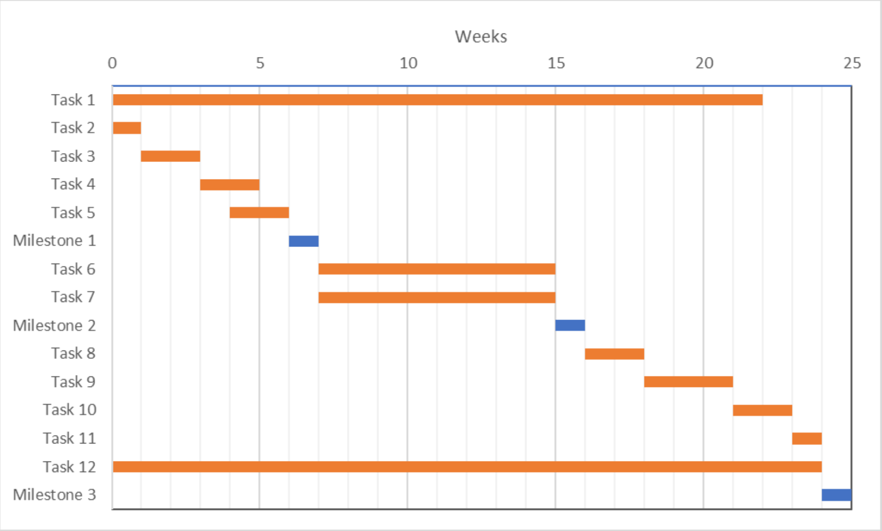


Figure 1: Original Gantt Chart

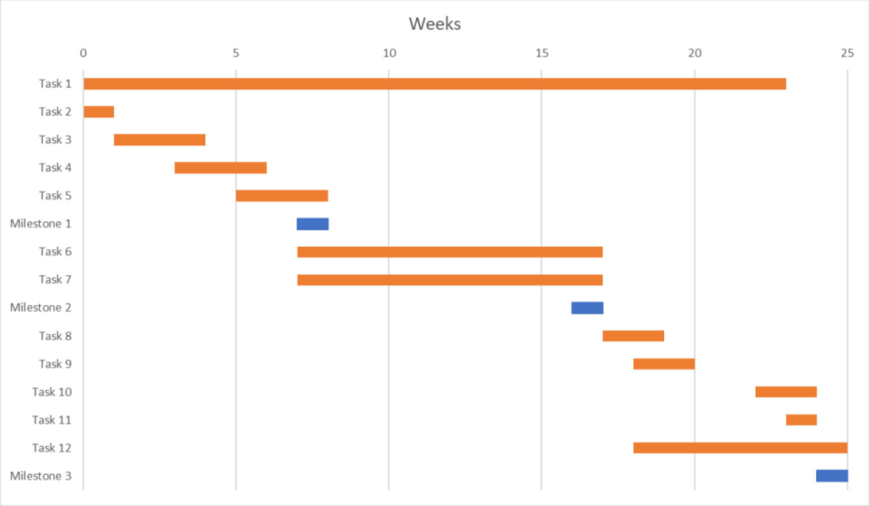


Figure 2: Final Gantt Chart

While the Gantt chart provided a good visual representation of how the project should pan out in the initial stages, it soon became clear that, in reality, there are unforeseen occurrences that disrupt the course of events and impact the timeline. For instance, the work of other university assignments demanded more time than anticipated which meant that certain parts of the project needed to be put on hold until the other assignments were completed. Additionally, some tasks laid out at the start ended up taking less time in practice than originally anticipated while others took longer, this therefore meant that the original Gantt chart lost its relevance soon into the project. To compensate for this, amendments were made to the initial Gantt chart along the course of the project to give a more up-to-date impression of how the project was shaping up, with some of the notable changes being the extension of the development phase due to illness prohibiting work on the project and the introduction of a gap between tasks 9 and 10, as the University Easter break meant that it was challenging to organise study sessions with the participants.

Toolsets and Machine Environments

For the project management component of this project, the key factors that influenced the selection of the tools used was the availability of the tools and their accessibility in a range of scenarios. For instance, *Google*'s *Calendar* tool was used extensively for setting goals, milestones and deadlines throughout the course of the project as it is available anywhere, so long as the user has some form of internet connection. Additionally, through connected functionality with a modern smartphone, notifications and alerts can be set to pop-up at certain times to remind the user of various events, which proved invaluable for setting deadlines on getting certain aspects of the implementation completed and ensuring that this personally set goal was not forgotten. Another tool utilised to aid the project management side of the project was *Microsoft Excel*, as this helped to structure the Gantt charts and their related information regarding how long each task should take, using this information to provide a visual representation of the project's time structure that could be adapted quickly and easily when required.

One tool that provided uses for both project management and software development, is the versioning tool *GitHub*. This not only provided a way to continually back up the code implementation of the project, meaning that every step along the development process was saved as a version so that, if any part of the code broke irreversibly, an earlier working version of the code could then be accessed.

With regards to the software development side of this project, there were some vital components that needed consideration. Chief among these was how to appropriately model the desired physics in the produced artefact, as there are many approaches that could be taken. One way in which this could be achieved is to produce the physics functionality independently, using a dedicated SDK of an existing physics engine middleware such as *PhysX* or *Bullet*, then using a graphics engine on top of this to present a visual component to the player, such as *OpenGL*. However, this would require a vast amount of work to build the system from the ground up as both the physics engine would need to be successfully implemented and the graphical side of the game would need to be produced entirely from scratch, making this approach unfeasible given the time and resource constraints with this project.

Another important consideration was in which physics engine to use, as there are a number of well documented products on the market, with the most prominent of which being the aforementioned *PhysX*, *Havok* and *Bullet*, though each of these engines has their advantages and disadvantages. *Havok* for instance, while it is by far and away the most popular dedicated implementation in the video game industry, it is also closed-source, requiring the developer to be accredited and to pay an expensive fee to license its usage. This therefore meant that the use of *Havok* would be off the table for this project, as it is a limited production with a relatively small amount of resources available. *Bullet* on the other hand is entirely open-source and freely available, while still maintaining a rich feature-set of physics techniques, such as rigid & soft body simulation, collision shapes and even deformable objects. However, the issue with *Bullet* arises when trying to build a complex game system around it, as many of the existing solutions using it are not as richly featured or conducive to a project of this nature. This is where *PhysX* comes in because, as of *Unity 5* (Unity Technologies, 2018) and *Unreal Engine 4* (Epic Games, 2017), *PhysX* has been built into industry standard game engines by default as the underlying physics engine running the simulations.

This approach of using an established game engine that already features an implementation of the required physics engine functionality was selected as this provides a well-documented development environment, with a large community of helpful developers, and handles the visual aspect as well, with competent high-fidelity prototypes being much easier to produce over coding the entire game system from scratch. It then became a question of which game engine to use, since both *Unity* and *Unreal Engine* feature the *PhysX* physics engine. The choice of *Unity* was finally made, as the *Unity* development environment is designed to be more conducive to small scale projects and prototypes, such as this, whereas *Unreal Engine* features a lot of additional overhead that makes it more suitable for larger scale, commercial products. *Unity* also features a very quick and simple build system which allows the game to be swiftly packaged up and deployed on any number of devices, with the primary focus of this project being on producing it for the *Windows* PC platform, which would aid the setup of the research study later in the project.

Research Methods

In terms of evaluating how successful this project has been in its aims and objectives, research was determined to be a key component of this project, as the emphasis is on the player experience of using deformable terrain to solve puzzles.

To help answer this research question, as well as drawing other conclusions about the execution of the project, it was determined that both quantitative and qualitative data would need to be collected through the study as this would give a good balance of information regarding how the play session went measurably and how the players deemed their experience. For the quantitative data, this should be collected through in-game metrics and logging in the code, giving an overall view of how players interact with the game on a technical basis. The qualitative data however, will be collected through post-game interviews, as this will allow the players to express their play experience in a more nuanced way and will therefore help to gauge how the players felt while playing the game, especially with regards to the terrain deformation mechanics and their application in the artefact to solve puzzles.

In terms of representing these results, much of the quantitative data will be presented in the form of graphs and tables, to give a visual representation of how the players interacted with game in terms of the time they spent on each level, how many times they had to restart the level and how much they played around with the terrain deformation mechanics. As the qualitative data takes the form of more longform, crafted answers, these will be transcribed and typed up, with the key themes analysed and answers of key relevance to the research question being highlighted and investigated.

Design, Development and Evaluation

As the aims and objectives laid out for this project demanded both the development of an artefact, as well as the research and evaluation of said artefact, it was imperative to approach these components in a careful and thorough manner.

Software Development

For the software development component of this project, there were key aspects of the development process that could be separated out into the following distinct tasks.

Requirement Elicitation

The foremost step in the software development section of this project was to determine what the requirements of the final artefact would be, as there is no client to speak of, these requirements were defined by research question we are looking to answer. In this case, our primary area of interest was the implementation of real-time dynamic terrain and using this to solve some sort of puzzle in a game. Therefore, the basic requirements for this artefact were to produce a game that presents the user with a selection of puzzles to solve, with the primary mechanic available to them being the ability to deform the terrain in the level, so there needed to be some degree of real-time physics simulation to model this.

Additionally, it was important to outline a target demographic and hardware for the game, as this would help to frame some of the design decisions and features of the game. As the testing for this game would take place at the University of Lincoln campus, the primary target group would be University students and would need to run optimally on the University computer hardware. Following on from this, a user scenario was crafted to gain an understanding of the game's players, what they would expect from a game of this type and how they may want to interact with it. The user scenario set out in this case was a young university student who consumes video games as a large part of their media diet and enjoys encountering new challenges, hence this target user was a key consideration when entering the design phase of the game's development.

Design

The next step in the process of development was to design the game itself, as this would then form the basis of the entire game it was important that the concept was a strong one. Firstly, a range of different game concepts were produced to cover a range of potential mechanics, these were then evaluated and whittled down to one final concept. One of the discarded game ideas followed a similar set of mechanics to *Super Monkey Ball* (Amusement Vision, 2001), where the player's main aim was to guide a rolling ball to a goal at the end of the level, past a range of obstacles that the player would be required to deform the terrain upwards or downwards to navigate the ball around. However, upon consideration, it was determined that this game idea would discourage experimentation and appreciation of the deformation mechanics as the players would be aiming to navigate through the course as quickly as possible. The base idea that was finally selected was a puzzle-platformer, where the player needs to reach and hit switches to materialise platforms that form a path to the level's exit. These switches should be in hard to reach locations that require the player to either raise or lower the terrain, so they can hit them. Though the player has limited usage of their terrain modification, meaning the player needs to be thoughtful about how and where they use this deformation.

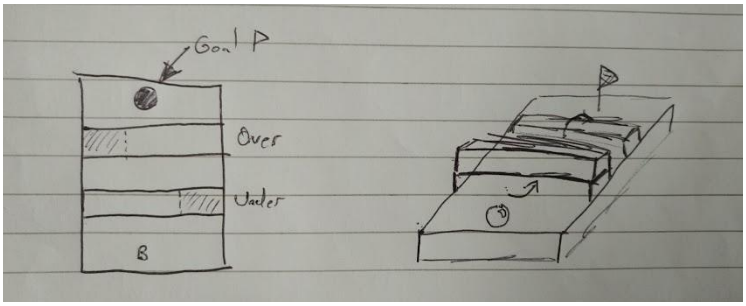


Figure 3: Sketch of original 'ball' idea

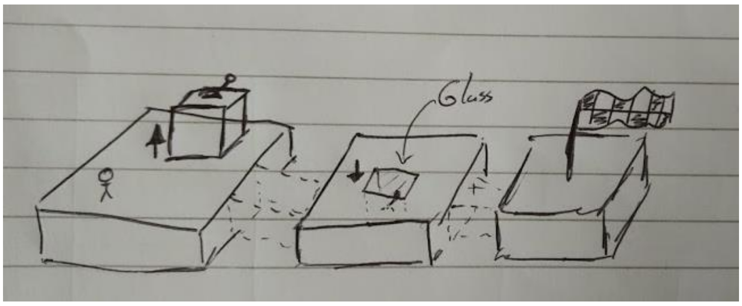


Figure 4: Sketch of original 'puzzle-platformer' idea

To establish this original idea as a firm design outline, a game design document was produced [Appendix 2]. This document sets out the core gameplay experience that the player is intended to have, the flow of the game, the mechanics that should be at the players' disposal and the intended solutions to the puzzles in the game. The high-level concept of the game outlined by this document determined that the game would take the form a puzzle-platformer where the players would control a character around small diorama levels, akin to the structure of levels in the puzzle game *Captain Toad Treasure Tracker* (Nintendo, 2014), their goal is to reach a progression platform by using their ability to deform the terrain both up and down to reach switches. These switches then materialise other platforms in the level, helping to form a pathway that will allow them to reach the goal in some way, be it directly (straight to the goal) or indirectly (to help reach another switch). Additionally, some levels feature another degree of challenge as there are some areas of the level where the player is prohibited from using their deformation abilities, thereby forcing them to consider alternative approaches to the problem. The player can also rotate the camera around the entire level to give them a different perspective on how to approach the puzzle, with some puzzles made simpler by taking this approach.

The game starts out by introducing the core mechanics to the players in three separate demo levels, giving them a chance to experiment with the mechanics in a relaxed environment. The first of these demo levels presents the player with a relatively empty space with the only notable feature being a raised platform with a switch on it, the colour of this switch relating to the transparent apparition of the level exit. Here the player can use the empty space to play around with the controls and deform the ground as they wish, with the intended solution simply being for them to deform the ground by the platform upwards so they can reach the switch on top of it. The following level is presented in a similar way, except the switch is recessed in the ground underneath a translucent block, encouraging the player to use the downwards deformation to progress. The third and final demo level introduces regions of the map that are highlighted in red and, if the player attempts to use their deformation mechanics in these zones, they will find they cannot move the terrain. This allows the 'blocked zones' mechanic to be introduced to the player in a way that should not overwhelm them, as it is the primary obstacle in this level. The following 3 levels in the game combine these various gameplay elements together to create more complex challenges, typically involving a step-by-step process where the player needs to interact with one switch to be able to reach another switch which will then help them to reach the goal. Each of these puzzles were produced considering an approach to puzzle design proposed by a contributor to game development site *Gamasutra*, entitled 'Four-step puzzle design' (Asher Einhorn, 2015). This framework for puzzle design sets out four key steps that should occur when a player encounters a puzzle:

1. The player understands the objective
2. The player discovers the puzzle
3. The player works out a solution in their head
4. The player implements the solution and solves the puzzle

This framework helped to structure the puzzles in the game, as it provides a clear understanding of how most players will approach any puzzle presented to them. This influenced the colour connections between the switches and the platforms they create, especially with regards to the level exit platform, as this forms a clear link between these two objects in the game. This approach also resulted in the dropping of a visible timer in the level, as it was felt that this could discourage players from thinking about the puzzle mentally and trying to come up with solutions that they could implement, instead encouraging a brute-force approach to solving the problem.

Coding

Due to the usage of Unity in this project, the choice needed to be made between which of the engine's supported languages to use, C# or JavaScript. However, Unity doesn't support full JavaScript, instead using an alteration of the language that supports static objects but rescinds many of the dynamic features. The engine does have full C# support though, with some advanced features included, which means that greater coding assistance and documentation can be found outside of the Unity development community, unlike its altered implementation of JavaScript. Additionally, Unity themselves noted that approximately 85% of all projects are entirely C# and are therefore dropping support for their 'UnityScript' version of JavaScript (Richard Fine, 2017), so it made for good industry practice to stick to C# for this project.

For the first step in constructing of the deformable terrain mechanics, it was imperative to explore *Unity*'s implementation of *PhysX* and to find a way of creating heightmaps which can then be modified in real-time by the player. This investigation led to the discovery of *Unity*'s 'Terrain' game object, which creates a large plane which the developer can then sculpt using sliders and brushes to build a varied landscape. This object therefore makes use of heightmaps to determine the height of each point on the terrain, taking the form of an array which is then visualised in *Unity* by a 'SplatAlpha' map. From here an issue arose, while *Unity* has the functionality to alter the heightmap of a terrain through inbuilt functions, there is little-to-no documentation regarding this functionality online. Therefore, a degree of experimentation was required, in conjunction with some limited anecdotal responses on the subject found in the *Unity* development community, to find a way of allowing the player to alter the terrain in real-time. The final implementation of this mechanic utilises an image file for the creation of the craters/hills, using the alpha values of the image to produce a smooth dip or rise in the terrain. Upon the player calling the deformation function, the script first retrieves the values of the heightmap at the current location, then it calculates the new value for the height of the terrain where the crater or hill is made. Whenever this function is called, it also decreases the available usage of the mechanic for the player, which is represented in-game by a slider that draws from this variable.

Much of the key functionality of the game, such as the controlling of the player character and the interactions they can have with the objects in the game, was separated out into different scripts which were then applied to the relevant actors within the scenes. This allowed for a degree of object orientation in the implementation, as each object in the game has its corresponding variables and functions connected, with the public components being visible in the editor.

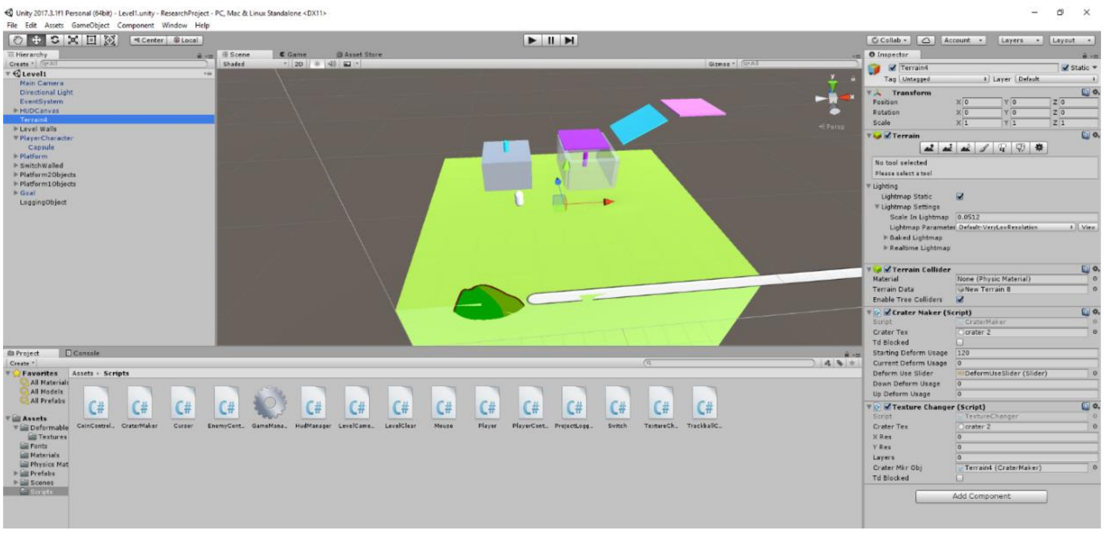


Figure 5: Unity Editor

To facilitate both the experimentation of features in Unity and the creation of multiple levels for the game, use was made of *Unity*'s scene management, which allowed each of the levels to be separated out and constructed individually. This meant that testing and prototyping of the different mechanics could be performed in a closed environment that would not negatively impact any of the other levels in the game. This also provided a way to implement progression in the game, as the scripts attached to the level exits could then load the next level of the game as soon as the player enters its trigger collider.

To improve the user friendliness of the game, all the controls were changed towards the end of development to draw from *Unity* inputs, which can be mapped onto a gamepad. This made the controls of the game much easier to communicate as they could be labelled on a diagram of the controller being used [Appendix 3], which would aid the research component of this assignment.

Testing

It was vital, upon each implementation of a mechanic or level, to personally test the gameplay impacts of the additions made. As the development of the artefact was carried out in the Unity game engine, much of the testing was performed using the editor's inbuilt player which allows the user to playtest the current build of the game. An additional functionality of the *Unity* editor is a bespoke console that can be used to log and debug any problems that arose therefore, whenever an issue would arise in testing, the corresponding section of code would be adapted to log out the relevant variables and data to see where the issues were stemming from.

As the game was designed for the Windows platform, the game could be easily compiled and built into a complete executable for the x86_64 architecture. This was done to ensure that all the functionality of the game worked when built and could be deployed on multiple different systems, which would be paramount of importance when carrying out the research phase of this project. However, thanks to the parallels between the *Unity* editor's player and running the game fully built on the PC, there were next to no discrepancies between the two builds of the game and meant that the editor's play mode could be relied on for most of the testing process.

Operations and Maintenance

When it came to performing maintenance and operations on the artefact, the relevant sections of the code that were being tested would be altered to output information to *Unity*'s console or, in the event of checking the metrics being tracked for testing purposes, through the external logging text file. This proved to be valuable for working on the dynamic terrain implementation, as some issues arose during the development of this mechanic that were difficult to discern from the limited documentation. For instance, at points the deformation appeared to stop working but, through this console logging, it was discovered that the code was indeed working as intended but the settings of the terrain object itself were preventing the terrain from deforming downwards, therefore this setting simply needed to be changed. This logging, combined with the straightforward on-the-fly testing through the *Unity* player, allowed for quick maintenance whenever a new feature was implemented and performing maintenance on any issues that popped up as a result.

Research

Once the development of the artefact was completed, the next step in the process was to begin the research component and gathering data from play-testers to evaluate how successful the artefact was in meeting the aims and objectives of this project.

The first step here was to try to recruit participants for the study, which required a determination of how many participants would be required and how best to elicit them. As the study would take place on the University of Lincoln campus, participants were recruited from around the University area, through fielding messages on online forums and from asking for participants in the computing labs. These participants were asked for an email address to communicate with them and determine when they would be able to participate in the study, which was the extent of the usage of this information.

As with any study involving human participants, it was vital that ethical procedures were followed in the collection of data from these play-testers. To help facilitate this, a user testing consent form was produced [Appendix 4] to inform the research participants as to what the study is about, what they will be doing during the study session, what information is being collected from the session and how it will be used. In addition to this, all research participant data is completely anonymised, with participants told that they could pull out of the study at any time if they wanted.

A study design was then produced to help structure the research process in terms of what research question was being investigated, what data would be required to answer this question, how this data would be collected and the procedure of the study. The first stage in producing this study was to outline the research question to be answered which, given the focus on producing a puzzle game that utilises deformable terrain in the solutions, the research question was outline as:

- Does the implementation of player deformable terrain as a tool to solve puzzle create a positive player experience?

To help answer this question, as well as draw further inferences and conclusions from the play-testing, both quantitative and qualitative data needed to be collected. Most of the quantitative data could be collected from metrics and logging in the games code, this included information such as: play time per level, amount of times reset per level and the player's usage of both the upwards and downwards deformation per level. In addition, a form of quantitative data would also be collected through the administration of select questions from the *Game Experience Questionnaire*, which operates on a Likert-type scale [Appendix 5]. The qualitative data on the other hand, which relates more closely to the quality of the player experience at the heart of the research question, would be collected through interview questions delivered to the participants [Appendix 5]. Once this data was collected it would need to be analysed; due to the small study size and emphasis on qualitative information, the quantitative data would be simply analysed through descriptive statistics to generally investigate trends in the study and visualise it using charts or tables. The qualitative data however would be investigated using thematic analysis, using the deductive approach as the research question is already laid out and the analysis is carried out in-line with it, through identifying key themes and patterns that emerge from the collected data.

The general procedure that each play-tester experienced was structured as follows:

- The participant was introduced to the subject matter of the project
- They were told what they would be required to do (Play the game, answer some survey questions and then respond to some brief interview questions)
- The participant was then presented with the informed consent form, which reiterates what the focus of the project is, what they will be doing in the study and the information that will be collected
- The player was then given the control sheet for the game and given up to 10 minutes to play through the game
    - Should they encounter any issues, they were encouraged to ask the study administrator for explanation
- Upon either finishing the game, or reaching the end of the 10 minutes, the participant was then asked to fill in a questionnaire that was provided to them
- After this they were asked the interview questions, beforehand being asked if they were averse to the interview being recorded for transcription purposes (if so, the answers were written down as the interview progressed)
- The participant was then thanked for their time and the study session concluded

Through the course of the study, eight participants were successfully recruited and the data from their involvement in the study was collated [Appendix 6]. The quantitative data was collected from the log files of the play sessions, along with the responses to the questionnaire, and then transferred into an *Excel* spreadsheet where the data could be analysed. From this it was discovered that the mean amount of time that players spent in the game was around 402 seconds, with the mean amount times that players reset being 9. This tells us that perhaps some of the players were taking a brute-force approach to solving the puzzles, encountering issues with their solutions and then having to reset the level, however it was observed that some of these players were using the reset functionality to give them more time to play around with the terrain deformation.

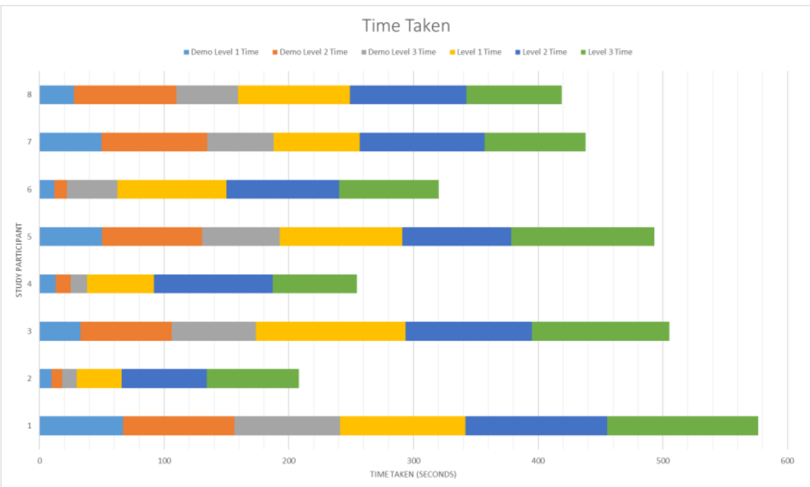


Figure 6: Stacked Bar Chart of Time Taken In-Game

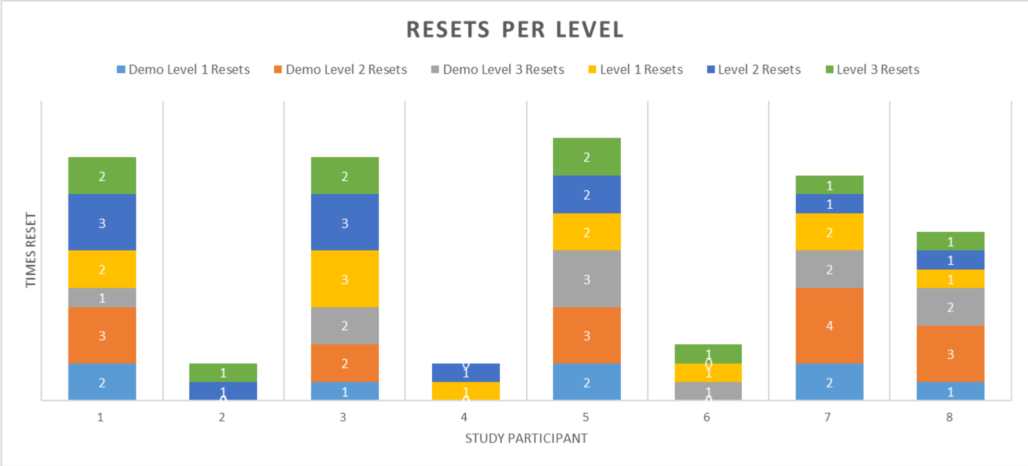


Figure 7: Stacked Column Chart of Level Resets

We can find from this stacked bar chart of the time that players took in the levels of the game (Figure 6), that many of the players took an elongated time playing the game. This shows that, when viewed in conjunction with the chart showing the number of times that players reset (Figure 7), that most of the players engaged in experimenting with different solutions and testing out the deformation mechanics instead of simply finding the quickest route through. This could show that players were using the structure of the game to simply play around with the deformation mechanics or perhaps they were trying to find alternate solutions to puzzles which did not pan out, to determine this however we will need to investigate the data extracted from the questionnaire and interview components.

From the questionnaire given to the participants, in accordance with the scoring guidelines of the *Game Experience Questionnaire*, the scores of the various components being investigated were computed and then represented in a bar chart containing all the participants.

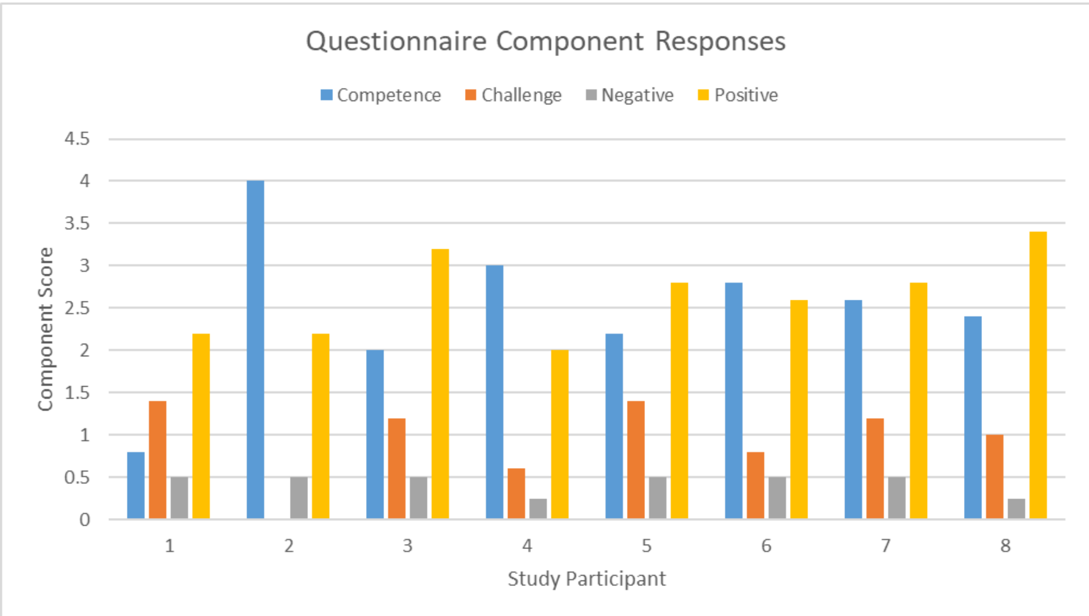


Figure 8: Questionnaire Response Graph

From this we can determine that, in general, players had a relatively high level of both competence and positivity. This bodes well for the concepts in the game as it means, not only did the players feel like they understood the tools and puzzles being presented to them, but they also felt like they had a positive experience playing the game, especially as no participant's negative experience score register above 0.5. However, what we do notice is that the scores relating to the challenge that players felt during the game are rather low, perhaps meaning that the puzzle design in the game wasn't challenging enough for the players, which is a theme noted in the qualitative data collected from the interviews.

From the interviews, there were several core themes that cropped up across the participants, one of these being the challenge of solving the puzzles. When asked a question regarding how they found the puzzles, there was a clear theme emerging that players found them rather simple and requiring relatively little effort to solve. Additionally, many players used this opportunity to air grievances with the switches that required deforming the terrain downwards to reach, with one participant mentioning that it proved difficult to solve these parts of the puzzle in practice due to the way the deformation works. This provides a vital consideration when producing a game such as this, where the puzzles should have a clear solution, but it should not be easily discovered so that the puzzle lacks challenge, as this proves somewhat detrimental to the player experience. Another theme encountered was the desire among the players for a greater freedom over their use of the terrain deformation, both in terms of how much they can use it and where they perform it, with players expressing a wish for the deformation to take place in a chosen direction in front of the player instead of directly underneath them. This links with the issues that players were having with the downwards deformation under walls, as they highlighted that this aspect of the mechanic made it difficult to create a dip to travel under the walls as intended. An additional pattern that emerged was an issue regarding the camera controls, which are designed to allow the player to view the level from an infinitely variable degree of angles and positions, however in practice players found that this made the player character much more challenging to control which comes as a result of the controls not taking into account where the camera is currently positioned around the level. While these themes focused around how the game could be improved, when asked about the terrain deformation mechanics themselves there was a prevailing pattern of enjoyment that arose among the participants, with respondents noting that they enjoyed the mechanic, had fun playing with it and found them an interesting concept.

Conclusion

Overall, this project found a level of success with applying deformable terrain as a game mechanic to solve puzzles, with a fully implemented prototype game that follows the conventions of a puzzle-platformer where the player is tasked to deform the terrain to solve the puzzles.

From the research carried out, we found that many players enjoying playing around with the terrain deformation, with some forgoing the primary driving force of solving the puzzle to instead experiment with the deformation and using it to find quirks in the game design. Furthermore, we found that some players managed to find unintended solutions to the puzzles set out in the game, which isn't necessarily a negative thing as it shows that the deformation mechanics implemented allow a degree of creative freedom in how the player approached the problem.

However, there were some noted areas for improvement that the play-testers touched upon with their experience in the game. For instance, some of the players expressed that would have preferred a greater amount of freedom with the terrain deformation tools as it would have allowed them to experiment a bit more and understand the different ways they could approach the puzzles. This could be facilitated by altering the design of the game, so that players aren't dealing with a finite amount of time they can use the deformation which requires them to reset the entire level to experiment more, perhaps instead operating on a regenerative basis where the terrain will revert to its original state over time. Additionally, a few of the players found that the puzzles in the game weren't especially challenging, with a couple of players completing all the levels relatively quickly compared to the others. So, it may be beneficial in future to introduce more complex puzzles that will challenge the players' understanding of the mechanics, forcing them to consider different potential solutions before settling on one that works. Aside from the terrain mechanics, players also had issue with the movement of the camera, as the movement of the player did not account for the current position of the camera, so it therefore made some parts of the game slightly disorientating for the players, negatively impacting their experience. This highlights that greater attention should have been paid to these ancillary components of the game as, while they are not the focal point of the project, they still feed into the players perceptions of the game as a whole.

From the work laid out in this project, there certainly feels as though there is a definite potential for further applications of deformable terrain in the puzzle game genre, as it not only served well as the only core game mechanic in this project's artefact but also elicited a positive experience from the players, who expressed an interest in what a more polished implementation could look like.

Reflective Analysis

In general, I found that this project ran relatively smoothly and didn't encounter any major stumbling blocks, but there are certainly areas where issues arose, and compromises had to be made.

When it came to the software development side of the project, the development of the artefact didn't encounter too many major issues, though the lack of any substantial documentation regarding the alteration of *Unity* terrain heightmaps meant that developing these aspects ended up taking longer than anticipated. However, thanks to help from questions asked in the *Unity* development community, many of the obstacles encountered in the development were overcame without too much issue.

The implementation component of the product overall went rather successfully, as the concept of having player controlled deformable terrain was achieved and this was used to approach puzzle design in an alternative fashion. However, as discovered in the research component of the project, there were aspects of the artefact that could have been improved. The first of these improvements outlined would be to alter the player and camera controls to try and eliminate any control confusion when viewing the level from different angles. Additionally, the idea of extruding the deformation away from the player instead of performing it directly beneath them will be explored, as this should help alleviate the issues that play-testers noted when trying to create trenches underneath walls.

With regards to the management of the project, the development of the original Gantt chart in the Project Proposal stage very soon became forgotten in the grand scheme of the project's undertaking. I feel this is because, while that Gantt chart provided a valuable rough idea of how I would've liked the development of the project to pan out, it ended up being unfeasible in the real world. This came as a result of pressures from other assignments and work meaning that, in some instances, more time needed to be devoted to these endeavours which then left the timeframes of that original Gantt chart out of balance. While some amendments were made to the Gantt chart along the way, it soon became clear that having a set time structure for an evolving project such as this would simply not work.

The field of physics simulation in games continues to be an intensely fascinating subject area, as more and more technologies are developed that change the way in which games are applying these physics systems. One example of this is AMD's TressFX Hair, a real-time physics system used for the simulation of characters' hair in games, gaining widespread attention in the reboot of *Tomb Raider* (Square Enix, 2013). This library models each strand of hair as an individual element, allowing them all to react individual to forces such as gravity, inertia and wind, allowing the hair to move in a much more realistic way. Competing GPU designer *Nvidia* produced their own implementation of this technology through their *Hairworks* library. The problem with these libraries, as with any physics simulation that introduces a vast amount of additional computations, is that it dramatically impacts performance when they are used in games. However, as technology advances both in terms of CPUs and GPUs, we could see that computational expense either optimised or trivialised.

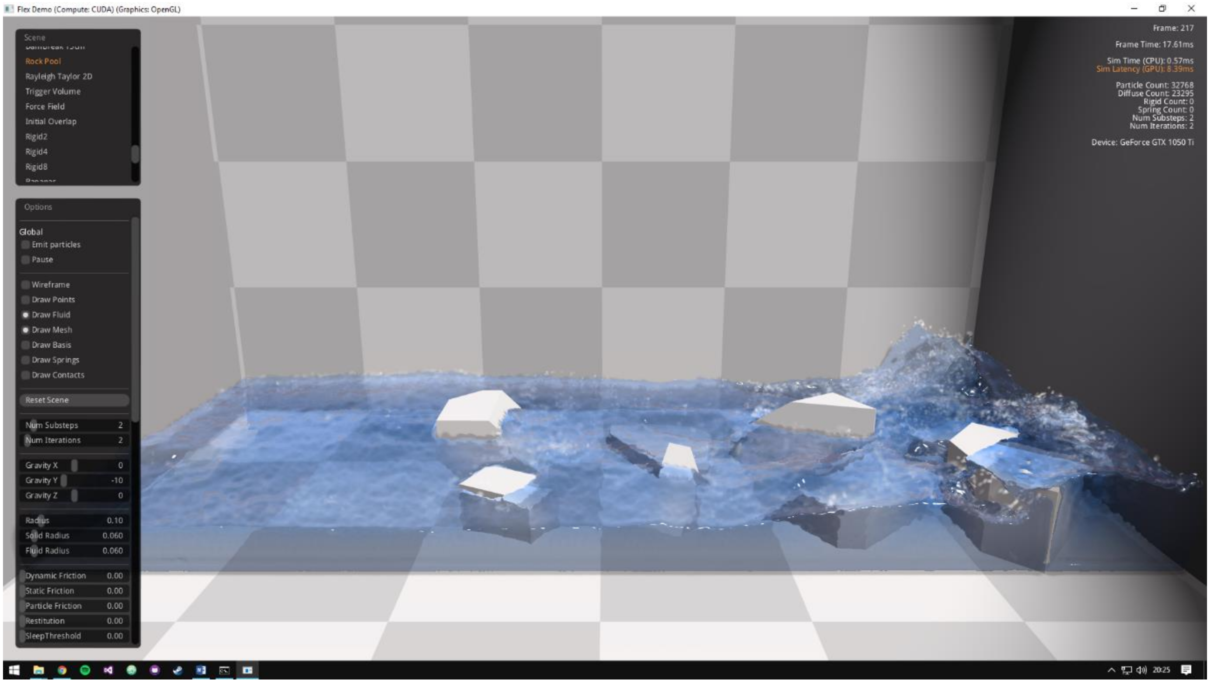


Figure 9: Nvidia's FleX Demonstration

Another fascinating development in physics simulation technology for games can be seen in *Nvidia's FleX*, a particle-based technique for real-time visual effects such as fluids, cloth, rope and gases. While it appears like it would be an incredibly intensive system to utilise in games, it already has a competent piece of demo software to show what can be achieved with *FleX*, and has even been implemented in a commercial game, *Killing Floor 2* (Tripwire Interactive, 2016). All these new technologies show promising glimpses of what could be common place in the future of gaming, perhaps even becoming as intrinsic as physics engines have become themselves, hopefully we could even see deformable terrain technology achieve this kind of prominence.

References

Adams, D.M., Pilegard, C. and Mayer, R.E. (2016) Evaluating the cognitive consequences of playing "portal" for a short duration. Journal of Educational Computing Research, 54(2) 173-195.

Amusement Vision (2001) *Super Monkey Ball* [game]. Sega.

Árnason, B. (2008) Evolution of Physics in Video Games. Available from http://www.olafurandri.com/nyti/papers2008/Evolution%20of%20Physics%20in%20Video%20Games.pdf [accessed 21 March 2018]

Asher Einhorn (2015) *Four-step puzzle design* [blog]. 28 May. Available from https://www.gamasutra.com/blogs/AsherEinhorn/20150528/244577/Fourstep_puzzle_design.php [accessed 16 January 2018]

Braun, V. and Clarke, V. (2006) Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2) 77-101.

Brockmyer, J., Fox, C., Curtiss, K., McBroom, E., Burkhart, K. and Pidruzny, J. (2009) The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45 624-634. Available from https://www.sciencedirect.com/science/article/pii/S0022103109000444 [accessed 2 March 2018]

Carvalho, J. Duarte, L. and Carriço, L. (2012) Puzzle Games: Player Strategies across Different Interaction Modalities. In: 4th International Conference on Fun and Games, Toulouse, France, 4-6 September. New York, USA: ACM, 64-72. Available from https://www.researchgate.net/publication/238601481_Puzzle_games_Player_strategies_across_different_interaction_modalities [accessed 28 March 2018]

Catanese, S.A., Ferrara, E., Fiumara, G. and Pagano, F. (2011) Rendering of 3D Dynamic Virtual Environments. In: 4th International ICST Conference on Simulation Tools and Techniques, Barcelona, Spain, 21-25 March. New York, USA: ACM, 351-358. Available from https://dl.acm.org/citation.cfm?id=2151116 [accessed 15 October 2017]

Crystal Dynamics (2013) *Tomb Raider* [game]. Square Enix. Available from https://store.steampowered.com/app/203160/Tomb_Raider/ [accessed 4 April 2018]

Epic Games (2017) *Unreal Engine* [software]. Cary, North Carolina: Epic Games. Available from https://www.unrealengine.com/en-US/what-is-unreal-engine-4 [accessed 8 November 2017]

Havok (2011) *Havok* [software]. Dublin: Havok. Available from https://www.havok.com/ [accessed 12 October 2017]

Holz, D., Beer, T. and Kuhlen, T. (2009) Soil Deformation Models for Real-Time Simulation: A Hybrid Approach. In: *Workshop on Virtual Reality Interaction and Physical Simulation*, Karlsruhe, Germany, 5-6 November. Goslar, Germany: Eurographics Association, 21-30. Available from https://www.researchgate.net/profile/Torsten_Kuhlen/publication/221622677_Soil_Deformation_Models_for_Real-Time_Simulation_A_Hybrid_Approach/links/0912f50c196bf67fbc000000/Soil-Deformation-Models-for-Real-Time-Simulation-A-Hybrid-Approach.pdf [accessed 2 November 2017].

Hunicke, R., LeBlanc, M. and Zubek, R. (2004). MDA: A formal approach to game design and game research. In: Proceedings of the AAAI Workshop on Challenges in Game AI, 4, 1-5.

IJsselsteijn, W.A., de Kort, Y.A.W, and Poels, K. (2013) The Game Experience Questionnaire [research]. Available from https://research.tue.nl/en/publications/the-game-experience-questionnaire [accessed 20 March 2018]

Law, B. (2016) Puzzle Games: A Metaphor for Computational Thinking. In: 10th European Conference on Games Based Learning: ECGBL 2016, Paisley, Scotland, 6-7 October. Reading, UK: Academic Conferences and Publishing International Limted, 344-353. Available from https://www.researchgate.net/publication/309611614_Puzzle_Games_A_Metaphor_for_Computational_Thinking [accessed 27 March 2018]

McKeown C. (2016) Alternative Trajectories: Structuring play through videogame physics engines. Performance Research, 21(4) 95-99. Available from https://www.tandfonline.com/doi/abs/10.1080/13528165.2016.1192879 [accessed 19 January 2018]

Naughty Dog (2007) *Uncharted: Drake's* [game]. Sony Computer Entertainment. Available from https://www.playstation.com/en-gb/games/uncharted-drakes-fortune-remastered-ps4/ [accessed 18 November 2017]

Nintendo (2014) *Captain Toad: Treasure Tracker* [game]. Nintendo. Available from https://www.nintendo.com/games/detail/captain-toad-treasure-tracker-wii-u [accessed 8 December 2017]

Nordin, A.I., Denisova, A. and Cairns, P. (2014) Too Many Questionnaires: Measuring Player Experience Whilst Playing Digital Games. In: *Proceedings of the Seventh York Doctoral Symposium on Computer Science and Electronics*, York, UK, 30 October. York, UK: Department of Computer Science & Department of Electronics, University of York, 69-74. Available from https://www.cs.york.ac.uk/ftpdir/reports/2014/YCS/494/YCS-2014-494.pdf [accessed 21 March 2018]

Price, C. (2007) The usability of a commercial game physics engine to develop physics educational materials: An investigation. Simulation & Gaming, 39(3) 319-337. Available from http://journals.sagepub.com/doi/abs/10.1177/1046878108319579 [accessed 19 November 2017]

Richard Fine (2017) *UnityScript's long ride off into the sunset* [blog]. 11 August. Available from https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/ [accessed 14 January 2017].

Shaydulin, R. and Sybrandt, J. (2017) To Agile, or not to Agile: A Comparison of Software Development Methodologies [pre-print]. Available from https://arxiv.org/abs/1704.07469 [accessed 4 December 2017]

Stege, L. Lankveld, G. and Spronck, P. (2012) Teaching High School Physics with a Serious Game. International Journal of Computer Science in Sport, 11(1) 123-134. Available from https://www.researchgate.net/publication/266488840_Teaching_High_School_Physics_with_a_Serious_Game [accessed 23 January 2018]

Thekla, Inc. (2016) *The Witness* [game]. Thekla, Inc. Available from http://store.steampowered.com/app/210970/The_Witness/ [accessed 20 October 2017]

Tripwire Interactive (2016) *Killing Floor 2* [game]. Tripwire Interactive. Available from https://store.steampowered.com/app/232090/Killing_Floor_2/ [accessed 7 April 2018]

Ullman, T. Spelke, E. Battaglia, P. and Tenenbaum, J. (2017) Mind Games: Game Engines as an Architecture for Intuitive Physics. Trends in Cognitive Sciences, 21(9) 649-665. Available from https://www.sciencedirect.com/science/article/pii/S1364661317301134 [accessed 25 March 2018]

Unity Technologies (2018) *Unity* [software]. San Francisco: Unity Technologies SF. Available from https://unity3d.com/ [accessed 7 November 2017]

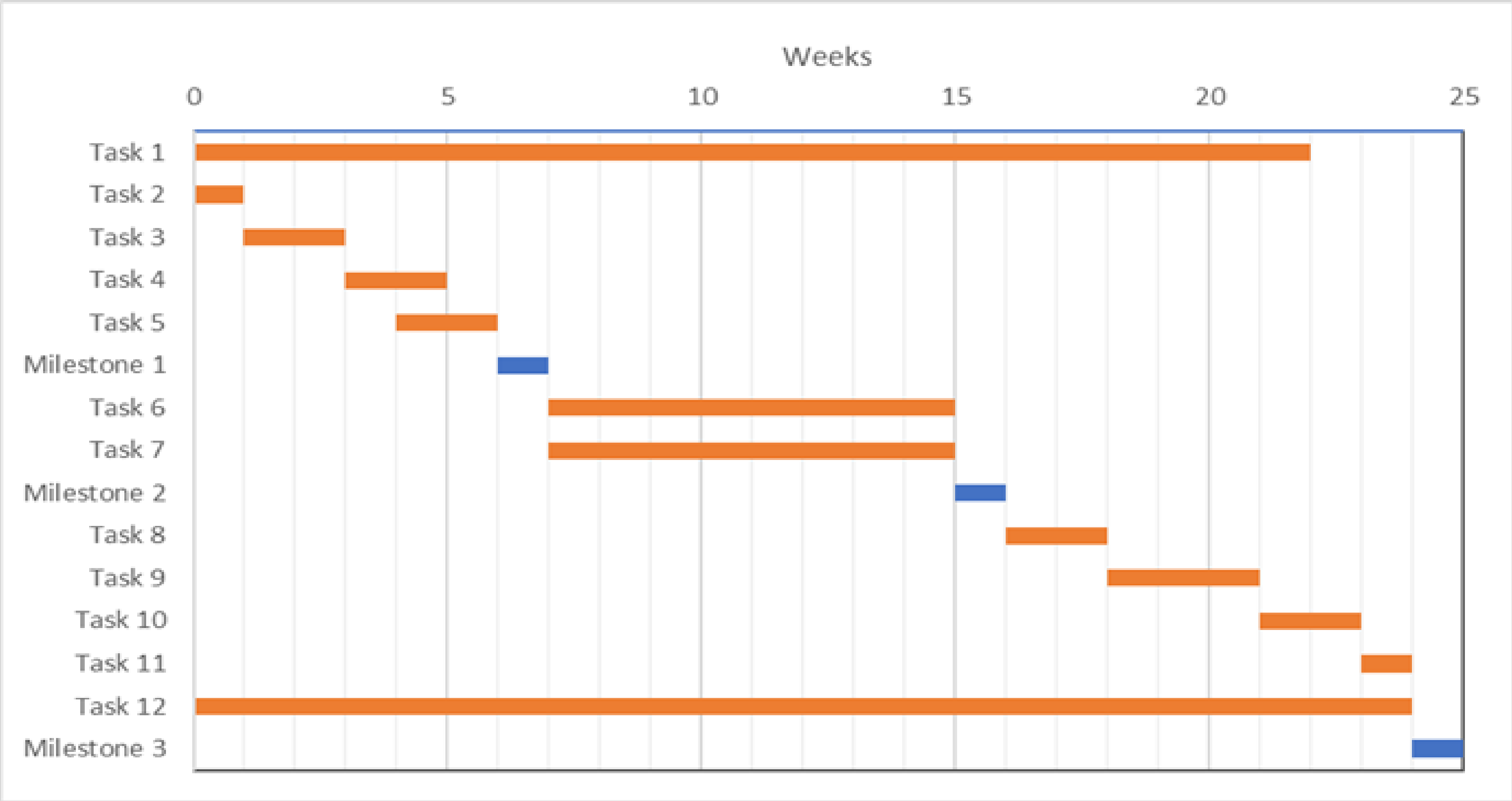
Vahdat, M. Carvalho, M. Funk, M. Rauterberg, M. Hu, J. and Anguita, D. (2016) Learning Analytics for a Puzzle Game to Discover the Puzzle-Solving Tactics of Players. In: Adaptive and Adaptable Learning. EC-TEL 2016, Lyon, France, 13-16 September. Cham, Switzerland: Springer, Cham, 673-677. Available from https://link.springer.com/chapter/10.1007%2F978-3-319-45153-4_89 [accessed 27 March 2018]

Zhang, Y., Wang, D., Yan, N. and Shang, Y. (2010) Real-Time Visualization of Tire Tracks in Large Scale Dynamic Terrain. In: *3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu, China, 9-11 July. New Jersey, USA: IEEE, 263-266. Available from http://ieeexplore.ieee.org/abstract/document/5565005/ [accessed 15 October 2017]

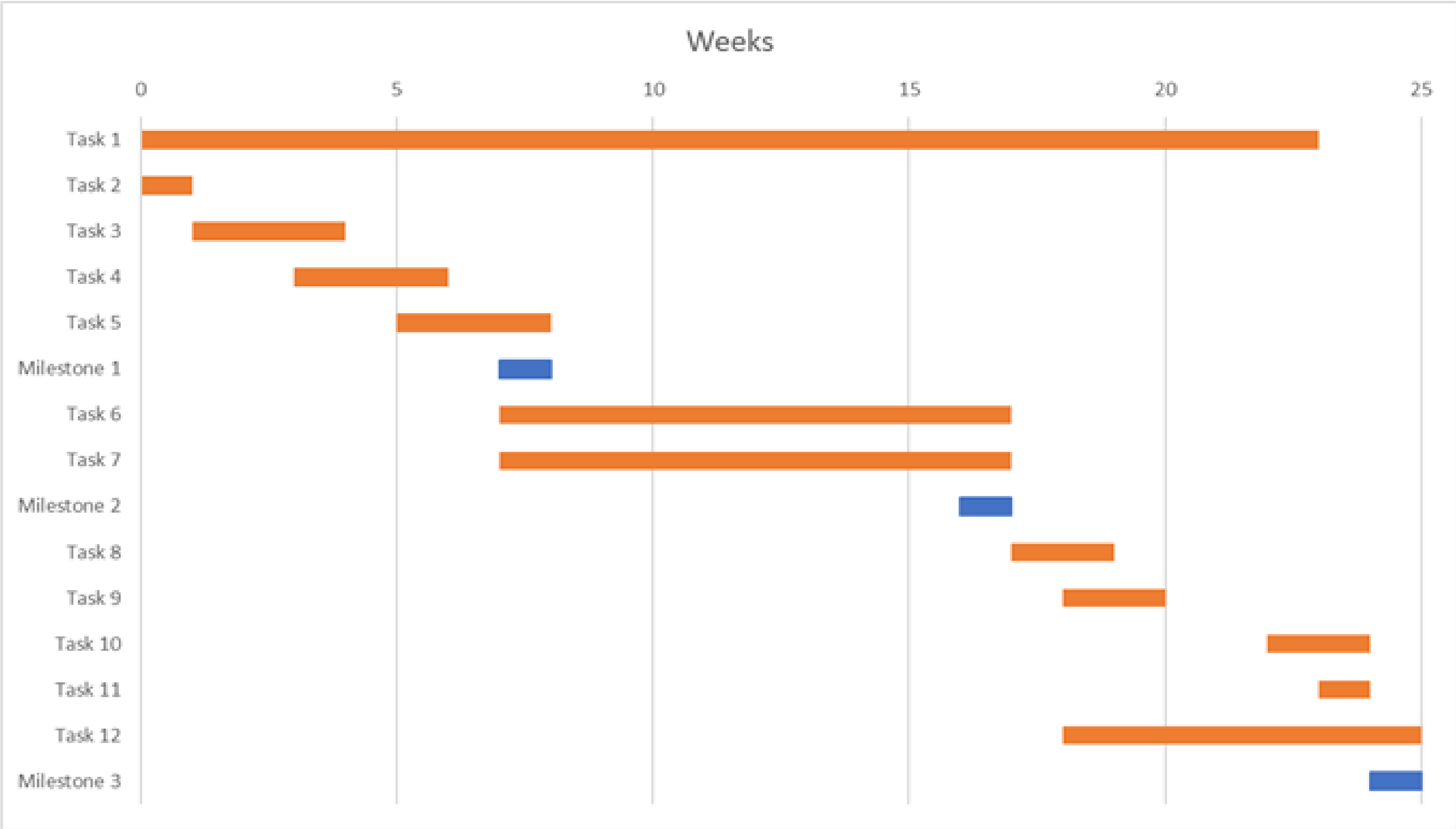
Zhou, Z. and Wu, L. (2012) The Study of Principles of Puzzle Game Design. In: 2012 International Symposium on Information Technology in Medicine and Education, Hokkaido, Japan, 3-5 August. IEEE, 1079-1083. Available from https://ieeexplore.ieee.org/document/6291487/ [accessed 25 February 2018]

Appendices

Appendix 1 – Gantt Charts and Tasks



Original Gantt Chart



Final Gantt Chart

Table of Project Tasks and Milestones Relating to Gantt Charts

Task Number	Task
1	Research existing literature in the subject domain, making notes of their relevance and consider how their findings could be used to shape this project. This will be an ongoing process throughout this project.
2	Investigate the existing physics and game engines available for use, making selections appropriate for this project.
3	To experiment with the chosen physics and game engines to see what their feature set is.
4	To draft designs and prototypes for how the game could play and function, consider how dynamic terrain could be used in a unique way to solve puzzles.
5	To refine these concepts to produce a definitive design document that will lay out the functionality the game should have, how it should play and how it should be presented.
Milestone 1: Have a completed design document for the game, highlighting how the game should look, play and behave. The physics and game engines to be used should also be selected.	
6	Develop the game, using the previously chosen physics and games engines, according to the specifications laid out in the game's design document.
7	Test out the state of the game and make any necessary tweaks along the way, keeping track of these and noting why they may have been required.
Milestone 2: The development of the game should be complete, ready to be presented to a selection of players to test it.	
8	To consider the information required from the user testing phase, write up a post-test questionnaire and a consent form for the play-testers.
9	Put a call out for people who would be willing to play-test the game from this project and answer a few questions about their experience with it.
10	Carry out the play testing phase, allotting time for all the previously sourced play-testers to try out the game, retrieve a log of their play sessions and ask them questions about their experience.
11	Compile the results of the user testing phase and evaluate their experience.
12	Write up the project report. The continual, overlapping nature of this task demonstrates that this will be an ongoing process throughout the project, with small additions and alterations being made throughout.
Milestone 3: The project should be fully completed by this point.	

On the Gantt charts, Week 0 relates to a week starting from 26th October 2017, with Week 26 relating to a week starting 26th April 2018.

Appendix 2 – Game Design Document

Game Name: Puzzling Terrain

Theme: Puzzle-Platformer

Target Platform: Windows PC

Concept:

The game takes the form of a third-person puzzle-platformer, where player must navigate through the level to reach the goal and progress. To achieve this, they must interact with switches throughout the level to make their corresponding platform materialise, these platforms helping to build the path to the end goal.

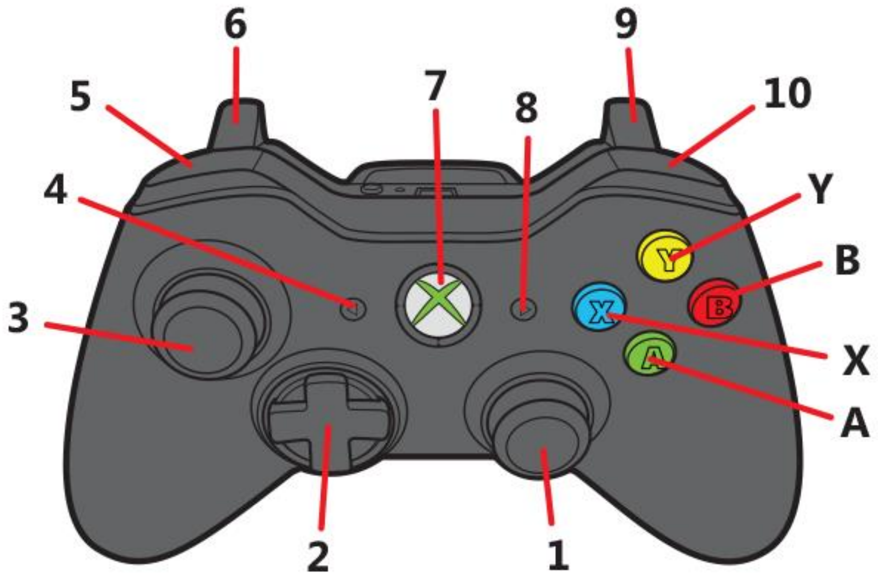
To reach these platforms and switches, the player has several mechanics available to them. They can move around, they can jump and, most importantly, they can deform the terrain beneath them either upwards or downwards. To reach switches on-top of tall platforms the player is required to deform the ground next to the platform upwards, then to reach platforms encased in glass boxes the player is intended to deform the ground downwards underneath the walls.

The game will be comprised of three demo levels and then three full levels, these levels are structured as:

- Demo Level 1
    - An introduction to the mechanics of deforming the terrain upwards. The level is largely empty, other than a lone tall platform with the level end switch on top of it, to guide the player towards using upwards deformation to reach it.
- Demo Level 2
    - An introduction to the mechanics of deforming the terrain downwards. The focal point of the level is the level end switch visibly encased underground, encouraging the player to deform downwards to reach it.
- Demo Level 3
    - An introduction to the additional level component of having regions where the player cannot use their terrain deformation. This level is largely similar to the first demo level, except the platform has a blocked zone around it, requiring the player to deform the terrain upwards outside of this zone.
- Level 1
    - A full level that combines the mechanics of the first two demo levels, with there being a tall platform next to an encased switch. The intended solution is to deform the terrain upwards between the two, allowing the player to reach the platform, and then deform underneath the encased switch elsewhere. This then creates a path to the level goal by using the original upwards deformation.
- Level 2
    - A full level that combines all the demo level components: a tall platform the player must deform upwards next to, a switch in an encased box and regions that block terrain deformation.
- Level 3
    - A full level that introduces a floating boxed switch, while also including the other mechanics of the game (tall platform attached to the ground and regions that prevent terrain deformation).

Appendix 3 – Controller Sheet

Game Controls



Button Number / Letter	Name	Functionality
1	**Right Analog Stick**	N/A
2	**Directional Pad**	N/A
3	**Left Analog Stick**	**Move the player character**
4	**Back Button**	**Resets the level**
5	**Left Bumper**	**Rotate camera left**
6	**Left Trigger**	**Deform terrain downwards**
7	**Xbox Button**	N/A
8	**Start Button**	N/A
9	**Right Trigger**	**Deform terrain downwards**
10	**Right Bumper**	**Rotate camera right**
A	**A Button (Green)**	**Jump**
B	**B Button (Red)**	**Zoom camera in**
X	**X Button (Blue)**	**Zoom camera out**
Y	**Y Button (Yellow)**	**Interact with switches**

Xbox Controller Diagram : https://support.xbox.com/en-US/xbox-360/accessories/controllers

Appendix 4 – Example User Consent Form

User Testing Consent Form

Study Administrator:

Participant Number:

This study is focused around the implementation of real-time deformable terrain as a core gameplay mechanic to solve puzzles. The intended audience for this game is casual video-game players who enjoy solving relatively straightforward looking puzzles using unconventional methods. The goal of the game is to reach and interact with switches that will create platforms in the game, with the pink switch creating the platform that leads to the level exit. Upon reaching this pink platform you will progress to the next level.

For the study period you will be given up to 10 minutes to play through the game, which will be running on a Windows PC with an Xbox 360 Controller hooked up to control it. You are under no requirement to complete all the levels, so play at your own speed. A sheet will be provided to you that will explain the controls of the game and label them respective to the controller provided. The first three levels of the game are for mechanic demonstration and getting comfortable with the controls, then the following three levels involve more complex puzzles for you to solve.

After you have finished the play session, you will be given a short questionnaire regarding your experience playing the game, after which you will be asked a brief selection of interview questions relating to the game.

The information that will be collected in the testing period will be logged data from your time playing the game, from the questionnaire administered afterwards and from writing up your responses to the interview questions.

Statement of Informed Consent

Upon signing this document, I indicate that I have read the description of the study and am aware of my rights as a participant. I voluntarily agree to participate in the study.

Print Name:

Signature:

Date:

Appendix 5 – Questionnaire & Interview Questions

Research Study Questionnaire

Regarding the game you've just played, please tick how much you feel the following statements applied to you.

I felt content					
I felt skilful					
I thought it was fun					
I was fully occupied with the game					
I felt happy					
It gave me a bad mood					
I thought about other things					
I found it tiresome					
I felt competent					
I thought it was hard					
I felt good					
I was good at it					
I felt bored					
I felt successful					
I enjoyed it					
I was fast at reaching the game's targets					
I felt pressured					
I felt challenged					
I felt time pressure					
I had to put a lot of effort in					

Research Interview Questions

(For Study Administrator)

Delivered post-game. Make notes on the answers given. Ask for permission to record for transcription purposes.

- Did you find the puzzles in the game engaging?
    - If not, why?
- Did you enjoy the mechanics of altering the terrain?
    - If not, why?
- Were there any parts of the game that detracted from the experience?
    - Why was this?
- In a more fleshed out version of the game, what changes would you like to see?
- Were there any other prevailing thoughts you had whilst playing the game?

Appendix 6 – Collated Data

Game Metrics and Logging Data

Participant Number	1	2	3	4	5	6	7	8	
Demo Level 1 Time	67023	9771	32932	13049	50139	11837	49582	27658	
Demo Level 2 Time	89201	8523	73284	12043	80345	10234	85039	82304	
Demo Level 3 Time	84953	11454	67495	13046	62031	40324	53323	49522	
Level 1 Time	100321	36110	120034	53504	98432	87496	68759	89475	
Level 2 Time	113874	68177	101238	95403	87496	90355	100231	93403	
Level 3 Time	121038	74023	110243	67548	114927	80124	81294	76485	
Total Time (ms)	576410	208058	505226	254593	493370	320370	438228	418847	
Total Time (s)	576.41	208.06	505.23	254.59	493.37	320.37	438.23	418.85	401.88775
Demo Level 1 Resets	2	0	1	0	2	0	2	1	
Demo Level 2 Resets	3	0	2	0	3	0	4	3	
Demo Level 3 Resets	1	0	2	0	3	1	2	2	
Level 1 Resets	2	0	3	1	2	1	2	1	
Level 2 Resets	3	1	3	1	2	0	1	1	
Level 3 Resets	2	1	2	0	2	1	1	1	
Total Resets	13	2	13	2	14	3	12	9	8.5
Demo Level 1 Up	192	84	167	100	203	82	165	142	
Demo Level 2 Up	116	0	63	0	102	0	152	33	
Demo Level 3 Up	178	71	204	89	297	96	83	21	
Level 1 Up	264	56	291	98	107	89	112	74	
Level 2 Up	271	62	231	136	188	59	102	106	
Level 3 Up	270	99	276	79	234	172	193	102	
Total Upwards Deform	1291	372	1232	502	1131	498	807	478	6311
Demo Level 1 Down	64	0	18	0	75	18	103	32	
Demo Level 2 Down	284	57	203	92	275	87	274	305	
Demo Level 3 Down	0	0	62	0	103	0	207	279	
Level 1 Down	106	64	118	142	241	151	193	166	
Level 2 Down	102	68	263	103	192	71	132	154	
Level 3 Down	125	71	134	51	186	93	68	78	
Total Down Deform	681	260	798	388	1072	420	977	1014	5610

Post-Game Questionnaire Responses

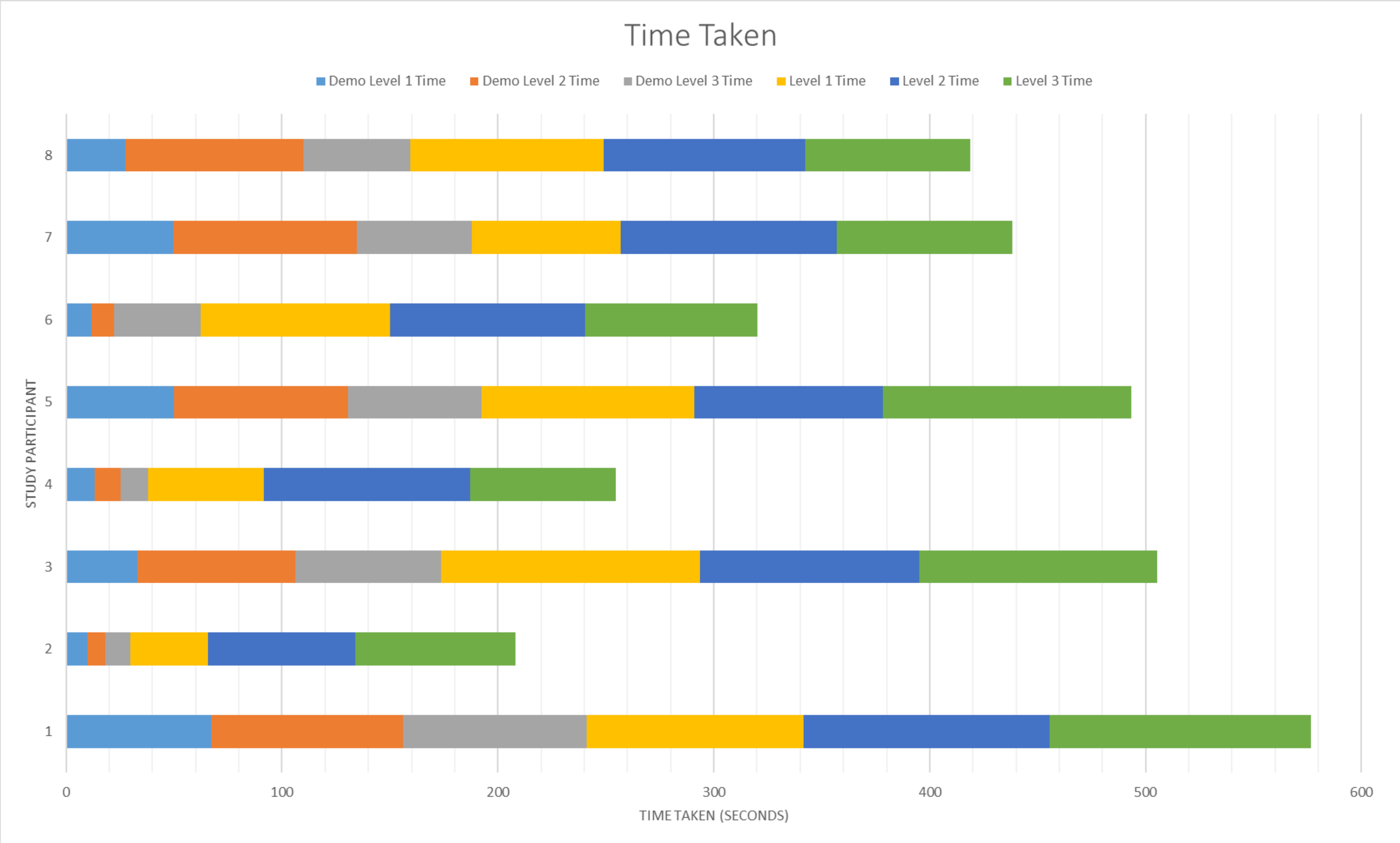
	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7	Participant 8
Question 1	2	1	3	2	2	3	3	4
Question 2	1	4	2	3	2	3	3	2
Question 3	3	3	4	2	3	2	3	2
Question 4	3	3	2	2	3	2	4	2
Question 5	2	2	3	2	4	3	2	3
Question 6	1	0	0	0	0	0	1	0
Question 7	0	0	1	1	2	1	0	1
Question 8	1	1	0	0	0	1	0	0
Question 9	1	4	2	3	3	4	3	3
Question 10	3	0	2	1	2	1	2	1
Question 11	2	3	3	2	2	3	3	4
Question 12	1	4	2	3	2	2	3	2
Question 13	0	1	1	0	0	0	1	0
Question 14	1	4	2	3	3	3	2	3
Question 15	2	2	3	2	3	2	3	4
Question 16	0	4	2	3	1	2	2	2
Question 17	0	0	1	0	0	1	1	0
Question 18	2	0	2	1	2	1	1	1
Question 19	1	0	0	1	2	1	1	2
Question 20	1	0	1	0	1	0	1	1
Competence	0.8	4	2	3	2.2	2.8	2.6	2.4
Challenge	1.4	0	1.2	0.6	1.4	0.8	1.2	1
Negative	0.5	0.5	0.5	0.25	0.5	0.5	0.5	0.25
Positive	2.2	2.2	3.2	2	2.8	2.6	2.8	3.4

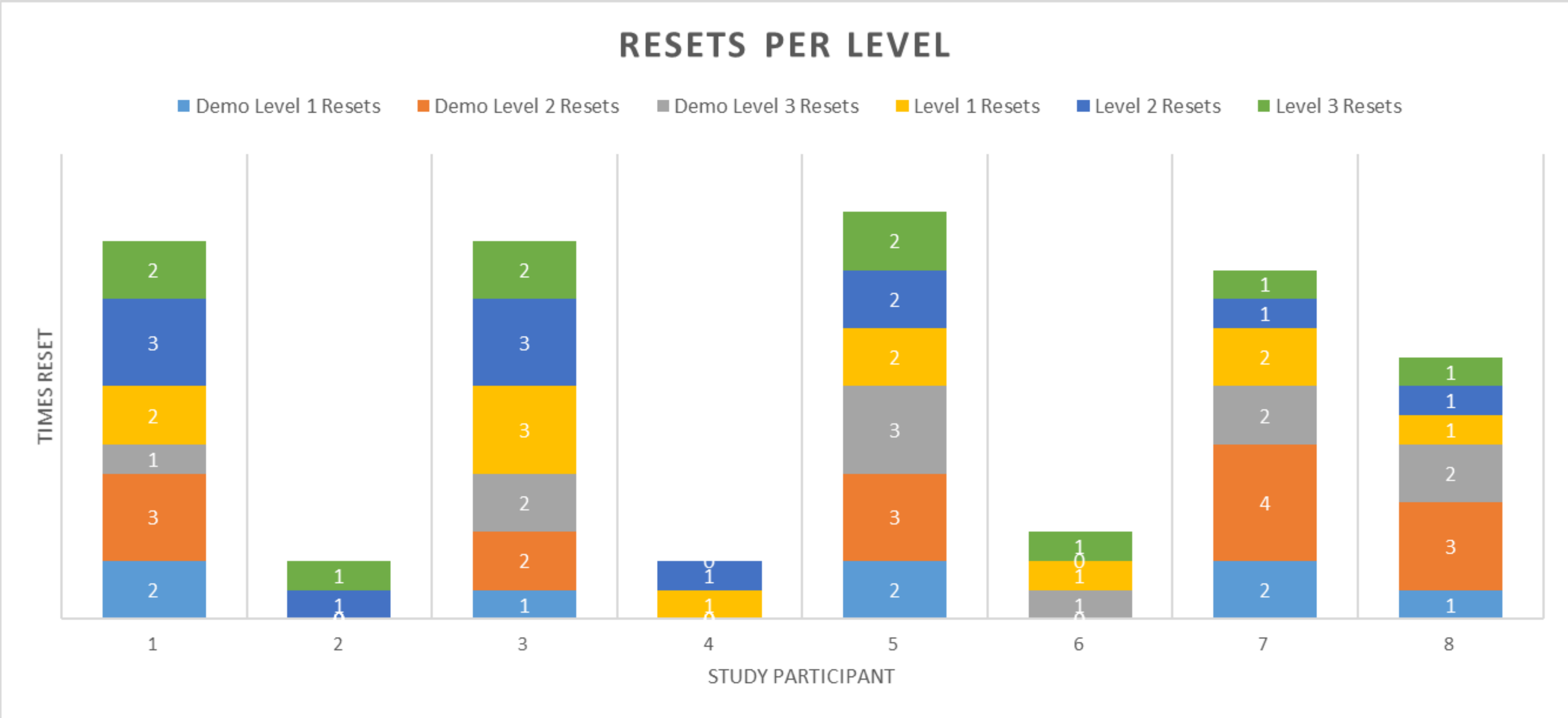
Post-Game Interview Responses

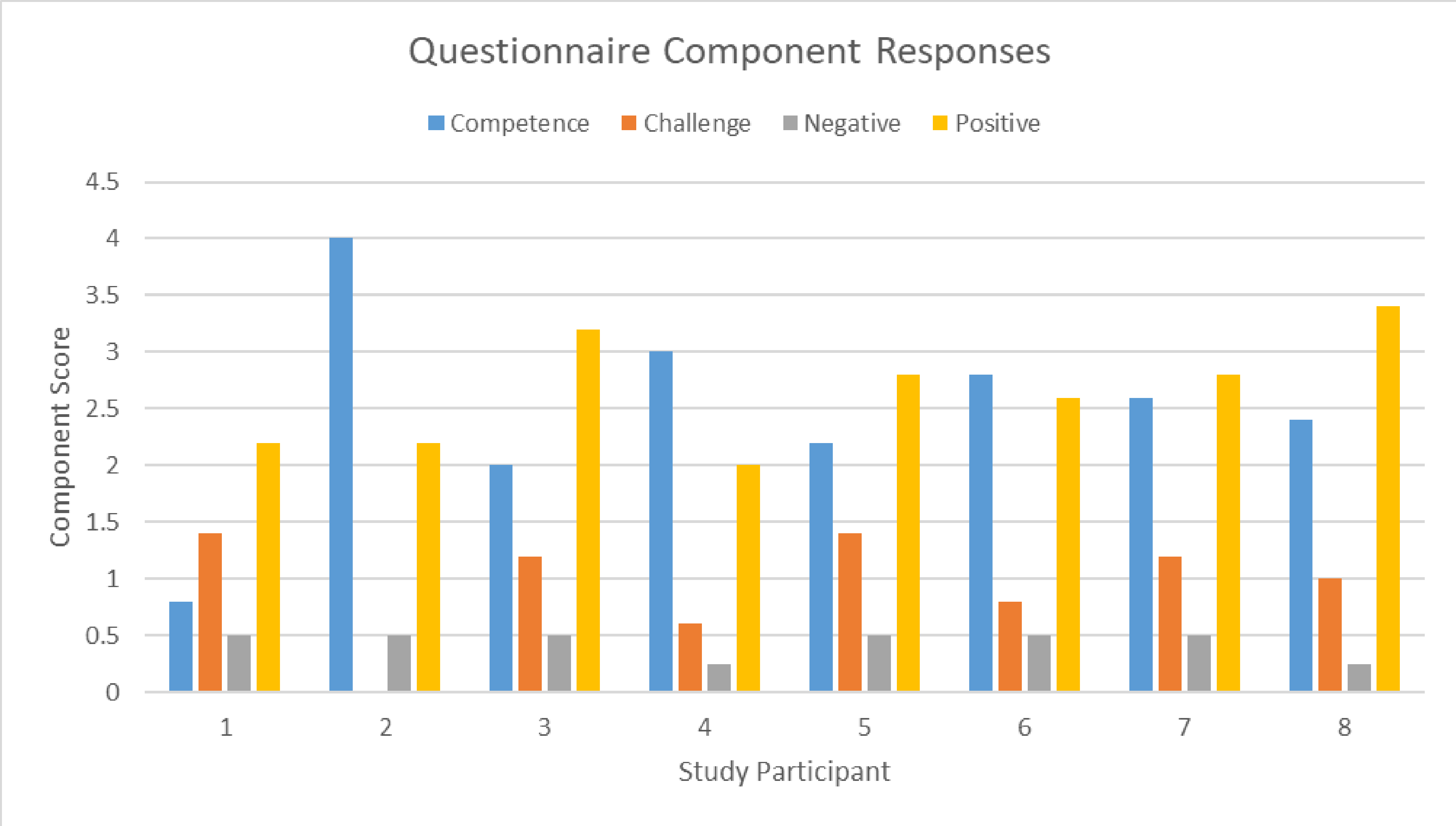
	Participant 1	Participant 2	Participant 3	Participant 4
Did you find the puzzles in the game engaging? If not, why?	I found the puzzles to be rather entertaining, though there were some slight design issues where I couldn't reach the switches sometimes.	I thought the puzzles were rather simple and didn't require too much effort to solve.	I enjoyed the puzzles in the game, though I would have preferred there to be more.	The puzzles were not particularly difficult, they could've had more challenging aspects.
Did you enjoy the mechanics of altering the terrain? If not, why?	I enjoyed the movement of the terrain, but I felt that it should have kept more of a flat surface, so the player can stand on it easier.	I liked the terrain alteration, though I felt it could have been better if it was done in front of the player instead of underneath.	The deformation was an interesting mechanic, I feel like it could lead to more complex puzzles.	The deformation was fairly enjoyable, though I feel it could have been used to solve more interesting puzzles.
Were there any parts of the game that detracted from the overall experience? Why was this?	Whenever you move the camera around the level, it makes it hard to keep track of how to move the player, which was rather confusing.	The camera controls, while they helped to give a view of the entire level, made it difficult to control the character.	The way the game is designed makes it a bit cumbersome to play around with the deformation, as you have limited usage of it.	It was rather difficult to smoothly get under some of the walls to reach certain switches because of the deformation happening under the player.
In a more fleshed out version of the game, what changes would you like to see?	I'd perhaps like to see larger levels where you can play around a bit more with the deformation, as it feels a bit limiting currently.	I would like to see some more challenging puzzles that use the terrain deformation in different ways.	I'd like to have more freedom in how to play around with the terrain deformation.	There should perhaps be some more complex challenges to make the puzzles more interesting.
Were there any other prevailing thoughts you had whilst playing the game?	It has a rather simplistic look to it, though the colours are quite pleasing.	I think there should have been more of a chance to play around with the deformation.	The game world could be more interesting to interact with and look at.	I would have preferred a different control scheme for the player, for when you move the camera.

	Participant 5	Participant 6	Participant 7	Participant 8
Did you find the puzzles in the game engaging? If not, why?	The puzzles started relatively simple but soon ramped up and were quite fun to work through.	The puzzles weren't especially complex, but they were still enjoyable to solve.	I found that some of the puzzles, in particular those where you had to go down, were rather difficult to solve in practice due to the way the deformation works.	Most of the puzzles were rather enjoyable, I liked the twist of having areas where you can't move the ground, but the walls you had to go under were a bit irritating.
Did you enjoy the mechanics of altering the terrain? If not, why?	I thought the deformation was fun to play around with and it made for a fresh and interesting way to think of a game level.	I had good fun with the deformation mechanics, as it made the level feel dynamic, though I would've preferred a bit more freedom to play around with them.	I think the idea of the terrain alteration was good, though I think it was rather simple in what it did, there could have been more ways for it to solve the puzzles.	I found the mechanics quite fun and interesting to play around with, though I think it would be better if you could choose where to do the deformation around your character.
Were there any parts of the game that detracted from the overall experience? Why was this?	It was quite difficult to control the character after moving the camera around the level, which made things a bit disorientating.	I think that it would have been better if the deformation wasn't so limited, so that you can play around with it a bit more to find different ways to solve the puzzles.	I found the walls you had to deform under a bit frustrating as you couldn't choose to place your trenches under the wall, just under your character.	The puzzles where you had to go under a wall to reach the switch were a bit of a pain as you couldn't easily make a dip under the wall.
In a more fleshed out version of the game, what changes would you like to see?	I think there could be a greater range in what you can do in the levels, perhaps having alternative side puzzles that the player can choose to do instead.	There should be a change to the player controls so that it takes into account when you've moved the camera around.	There could perhaps be a change to the deformation so that it occurs in front of the player instead of underneath them, so you can choose were to place it a bit easier.	I think there should be the ability to move where you perform the deformation in relation to your character, so you don't do it underneath yourself every time.
Were there any other prevailing thoughts you had whilst playing the game?	There could have been in-game prompts for the controls so that you don't have to refer back to the control sheet.	The player character was quite simplistic, I think it would've been more engaging to have an interesting character to play as.	There could have been a thematic reasoning for the deformation, such as the player having a spade or something.	I wish I would have had more freedom to play around with the ground deformation in the game.

Graphs and Charts of Quantitative Data




RESETS PER LEVEL
Demo Level 1 Resets
Demo Level 2 Resets
Demo Level 3 Resets
Level 1 Resets
Level 2 Resets
Level 3 Resets
TIMES RESET
STUDY PARTICIPANT
1
2
3
4
5
6
7
8


Questionnaire Component Responses
Competence
Challenge
Negative
Positive
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Component Score
1
2
3
4
5
6
7
8
Study Participant