

DESIGNING PATHOS

GAM6001 – Major Project
Individual Report

Green, Rob – 1300835
25/04/2016



ABSTRACT

This report recounts the work that the author produced in a team working to develop a game for a final year Major Project module. The report talks through the background, process and justification of the author's work and presents the author's reflection near the end of the project for both the individual and team process. The end of the report concludes with the author's journey to the Major Project module and how that preparation was essential to producing the final deliverable.

INDEX

Abstract.....	2
Index.....	3
Table of Figures.....	5
1.0 Introduction	6
1.0.1 Report Structure	6
1.0.2 Literature Review Synopsis	6
1.1 Background	6
1.1.1 The Project	6
1.1.2 Team Composition	7
1.1.3 The Game	7
2.0 Testing & Research	8
2.1 Process of Testing	8
2.1.1 Preparation	8
2.1.2 Data Collection.....	9
2.1.3 Demographic.....	9
2.1.4 Testing Process.....	9
2.1.5 Gameplay Process.....	11
2.2 Results of Testing	11
2.2.1 What was Found	11
2.2.2 What was Modified.....	12
2.3 Test Evaluation.....	14
3.0 Project Development	14
3.1 Designing with Concepts.....	14
3.2 Puzzle Design	15
3.2.1 Rules of Pathos.....	15
3.2.2 Puzzle Theory	15
3.2.3 Puzzle Mechanics	16
3.2.4 Level Design	17
3.3 System Programming.....	18
3.3.1 Outset Issues	18
3.3.2 Pathfinding.....	18
3.3.3 Design in Mind	19
3.3.4 Working with Mobile	19
3.4 Game Construction	19
3.4.1 Level Building in Engine	19

3.4.2	Branching Out	20
3.5	Project Management	20
3.5.1	Planning.....	20
3.5.2	Workflow & Pipelines.....	21
3.5.3	Version Control	21
4.0	Reflection	22
4.1	Team Reflection	22
4.2	Individual Reflection	22
5.0	Conclusion.....	22
5.1	Major Project	22
5.2	Continuing Over Summer.....	22
6.0	Appendices.....	23
6.1	Appendix A: Literature Review	23
6.2	Appendix B: Post-Test Questionnaire	29
	Bibliography	32

TABLE OF FIGURES

Figure 1 - Pathos Before and After Major Changes	7
Figure 2 - Test Area in the FixedUpdate Office	10
Figure 3 - Explicit State.....	10
Figure 4 - Intuitive State.....	10
Figure 5 - Process of Gameplay.....	11
Figure 6 – Comparison of Each States Gameplay Data	11
Figure 7 - Comparison of Questionnaire Data	12
Figure 8 - Lara Croft Go Push Mechanic in Action (TombRaiders, 2015).....	13
Figure 9 - Initial Work on Early Mechanics	16
Figure 10 – Chapter 2: From Design to Art	17
Figure 11 - Hand-Written A* Pathfinding Code	18
Figure 12 - Drag & Drop Prefabs with Custom Inspector in Unity 3D.....	19
Figure 13 - Representative User Interface.....	20
Figure 14 - Methods of Project Planning	20

Literature Review:

Figure 15 - Monument Valley (TabTimes, 2014).....	24
Figure 16 - Classification by design (Tulleken, 2011)	24
Figure 17 - suteF always showing the player's goal (Riggall, 2011)	25
Figure 18 - Loom (Lucasfilm Games, 1990).....	26
Figure 19 - Examples of different difficulty profiles for a sequence of 12 puzzles (Tulleken, 2011)	26
Figure 20 - Non-Linear structure of Fallout 3 (Shepard, 2014).....	27

1.0 INTRODUCTION

1.0.1 REPORT STRUCTURE

In producing this report, a lot of attention to the structure and flow of each topic has been paid to allow all information to be provided consistently and logically. The project background, author research and player testing precede the development process and issues faced to allow a bank of knowledge to be built up and keep the later, more in-depth, topics relevant.

As the report moves into the design process of core mechanics and systems, relation back to the author's research compiled in the Research Methods module will be apparent; such as how the author's research can be justified by the methods tried and tested, and how the application of the research has shaped the course of development.

The main focus of the author's individual work includes these topics:

- ❖ **4.2 - Puzzle Design** (*directly focusing on "4.2.2" and "4.2.3"*)
- ❖ **4.3 - System Programming**

1.0.2 LITERATURE REVIEW SYNOPSIS

In the Research Methods module, the author produced in-depth research from the area of puzzles, specifically focusing on their affiliation with video games. The aim of the research was to find a broad method or process to use when building puzzles from scratch. It was found that even defining what a puzzle is or means is highly debated, and that having a personal understanding of this area creates unique methods of challenging players. The research has shown the key points that should be considered for a puzzle in a video game. The player must be able to progress with the challenge, and this links with the other key point that the designer must allow the player to learn for themselves and apply their own logic. (*For more information on puzzle design in video games, see Appendix A*)

1.1 BACKGROUND

1.1.1 THE PROJECT

The team (FixedUpdate) and the project were unconventional in terms of this papers module. The journey started in early 2015 and followed on through the Dare to be Digital (Abertay University, 2000) competition later in the year. After winning two awards, the team received funding to develop the mobile puzzle game Pathos and that in turn allowed its use in the dissertation project for each of the individuals.

While development spans over the entire year of 2016, the aim for the Research Methods and Major Project modules was to provide the initial drive for Pathos in terms of teaching the player. Following the research in the author's literature review, it was clear that the player must feel like they have learned how to play the game in their own way, so the goal for the team was to create an implicit way to teach the player how to use basic mechanics. Achieving this goal would let the team focus on the main mechanic in the game (a flipping of worlds and perspective) and how it could be used without the consequence of confusion.

Over the course of the Research Methods module, a prototype was built for testers to play in a test session based around the team's research question. The content of the prototype would not be used in the final game, but the outcome of the test would determine the direction that the mechanics would be taking when teaching the player – do they need explicit text prompts or can they be intuitive enough to be understood through game play?

1.1.2 TEAM COMPOSITION

Originally a team of five (previous to the Research Methods module), FixedUpdate reduced in size to three individuals - one designer and two artists. With such a small team working on a big project, multiple roles would have to be taken by the individuals.

The role of the author was to be the designer on the project, working on the mechanics, puzzle logic and overall player experience. As a designer with scripting knowledge, the programmer role had to also be taken.

1.1.3 THE GAME

The prototype was built using three levels of progressing difficulty, each gradually teaching the player new mechanics as they played. Movement, interaction and a push/pull mechanic, which the author finally cut from the game, were available to the player, providing them with the original core mechanics. It was purposely built to be very quick to play so that each player could finish it quickly and provide relevant feedback while the logic was still fresh in their minds.

During the Research Methods module, Pathos was still in a concept stage and building the prototype allowed the team to see what worked for the overall project. The prototype used mechanics that aren't used in the final deliverable, but it was crucial to test them because player feedback told the author how irrelevant they were.

At the start of the Major Project module, Pathos underwent a major redesign using the research gathered in the author's literature review, and the feedback received from players, to produce a new concept that was to stick with the final deliverable and beyond. As the game would be released long after the Major Project module had finished, the team started full development on finalised assets while tailoring what was produced for the module prototype. The aim was to produce two of six chapters (thirteen of thirty-six levels) for the final deliverable.



Figure 1 - Pathos Before and After Major Changes

2.0 TESTING & RESEARCH

As part of FixedUpdates journey previous to the Research Methods module was unconventional, the team was afforded one other previous test session on a large scale. During the Protoplay event, the core of the Dare to be Digital competition, Pathos' first prototype was played by many hundreds of people over the course of four days. This afforded the team some prior experience when dealing with testers and what to do with player feedback, as well as finding useful ways of gathering that data.

With a basic prototype in place, the team decided to focus on an area that the art and design of the game was striving to achieve, and that became the subject for research in the project:

- ❖ *How much guidance do players need?*

With a plan to collect quantitative data, a hypothesis was put forward by the team:

- ❖ *People who play the tutorial version should have the same average data values as people who play the non-tutorial version.*

The team's research topic gave a strong foundation for the author's literature review, where the application of puzzle design in video games is greatly supplemented by how the player understands their objectives, and the tools to achieve them. This also works in the opposite direction – understanding fundamental methods of building logic and mechanics supplements the process of teaching the player how to use them to the fullest potential.

2.1 PROCESS OF TESTING

2.1.1 PREPARATION

For the team to test their research, a prototype with two different states had to be built. Both states were to be near identical, with the difference being how the player was introduced to mechanics – one state with explicit text tutorials and the other relying on intuitive artwork (3D models and textures) and level design. As the team was already working on concepts and white boxing level prototypes, it was a case of choosing the levels that best flowed together to form a basic understanding of the game. Once the team decided on three levels, artwork and polish was added to the white boxed scenes as well as systems for gathering data from the gameplay session.

Unfortunately for the team, Pathos is a mobile game and there was only one test device was available when test sessions began. But the device was capable of providing what was needed for testers; audio, smooth performance and a big screen to see text easily.

Considering ethical issues of the test, a process of gathering testers was produced to provide the safest possible environment for anyone playing the game. A verbal introduction would be given to share details of the test itself and to ask the testers permission to use their help for research. A post-test questionnaire follows the gameplay test informing the tester that their data would only be used in the team's research, if this was unacceptable they were advised to tell a team member and their data would be deleted. (*See Appendix C*)

2.1.2 DATA COLLECTION

A test session on a big scale provided the author with an opportunity to cover other areas of the game that weren't necessarily part of the research topic. As the game was in development for eventual selling on mobile devices, the team wanted to provide the best experience possible. With this in mind the author added data collection not only for the research topic but for prototype mechanics and how the tester interacts with the environments.

Two methods of data collection were provided for the test session:

- ❖ **Hidden Data** – Inside the game files, code was added to record what interactions the tester was using while playing, and the number of times they were used. Null touches (finger taps where nothing happens), number of mechanic uses and move to object events were some of the quantitative data gathered inside the game. Each of the three scenes was separated to determine what and where interactions were used, offering the team the ability to be direct in their analysis. This was then output to a text document in the test devices file explorer.
- ❖ **Post-Test Data (Appendix B)** – After the gameplay session was over, the tester was introduced to a post-test questionnaire. Its main focus was to cover qualitative data gathering so the team could determine what players liked or didn't like, but it offered the opportunity for more quantitative data to be collected. Tester's opinions could be offered through questions regarding a specific mechanic or how they felt about the experience as a whole. The quantitative data could be provided through agree/disagree questions with an answer of 1 (disagree) to 10 (agree).

Each instance of the methods was uniquely numbered so once the text documents were taken from the test device; both methods could be matched to a single test session.

2.1.3 DEMOGRAPHIC

Using university students, specifically games students, as testers proved to be more beneficial to the game as the first main test session since the Protoplay event. The team targeted the session at forty games art, design and programming students because of their exposure to games – if Pathos is not understandable to veteran game players, the casual market of players won't be able to understand it. The team needed twenty students per prototype state because once one state was played; the player then already knew how to interact with the game properly, skewing test results if they played again.

2.1.4 TESTING PROCESS

With only one test device the team had to stretch the whole process across three sessions, two sessions of ten students and a final session of twenty students, totalling forty testers over three days.

While the team's device issue made the process slower, being in a comfortable position with an office solely for the team to use balanced out the overall time it took to produce forty tests. Plenty of space was given to the testers - once a gameplay session had finished, another tester could sit and play while the previous could answer the questionnaire.



Figure 2 - Test Area in the FixedUpdate Office

The first stage of testing was to gather testers, two at a time. After verbally agreeing to take part, the team brought the testers to the office and sat them together in the area shown above. The author chose to lay out the area like this as to make the testers feel comfortable together, but not a distraction to each other.

Stage two was for the first tester to play through the gameplay demo, unaware of what they were being tested on (if the tester requested it, at the end of the session the team would share details on their data and how it was gathered). As the first player was busy, the second would be waiting in the questionnaire seat, however magazines and conversation were provided to pass the initial waiting time.



Figure 4 - Intuitive State



Figure 3 - Explicit State

Stage three was to change the two testers around, moving the first game player over to the questionnaire seat. While the second tester was now playing the game, the first would be answering the questionnaire, free of the teams input as not to skew results. Once the gameplay demo was over, the tester in the questionnaire seat would be just finishing allowing the other to take over and answer their own.

The process worked well over the three sessions, allowing testers to quickly come and go while providing a lot of detailed feedback.

2.1.5 GAMEPLAY PROCESS

In the prototype, a tester had to go through a process of gameplay to reach the end goal. As the team built the prototype states very similar to one another, it was simple to keep track of the overall process over three levels.

- ❖ **Level 1** – Introduction to movement and interaction mechanics, offers insight into the most basic core of the game.
- ❖ **Level 2** – Introduction to the push/pull mechanic, one of the experimental mechanics being side-tested by the author.
- ❖ **Level 3** – Alternate use for push/pull mechanic, introducing multiple roles of core mechanics. Another experiment being tested.



Figure 5 - Process of Gameplay

The difference between the two prototype states was that the opening of each level contained a text overlay telling the player how to use each mechanic in the explicit version.

2.2 RESULTS OF TESTING

2.2.1 WHAT WAS FOUND

After receiving the analysed data, it was the designer's job to look at the bigger picture and determine where the key problem areas were and what changes would need to be made.

Tutorial Version					Non-Tutorial Version				
	Level 1	Level 2	Level 3	Total	Level 1	Level 2	Level 3	Total	
Time	17.95	19.10	44.30	81.35	21.50	23.65	33.90	85.05	
Non-Grid Touch	1.45	7.10	3.50	12.05	3.75	12.25	4.25	20.25	
Player Movement	5.80	3.30	9.65	18.75	8.70	5.10	10.80	24.60	

Figure 6 – Comparison of Each States Gameplay Data

The data seen above is the main focus of the team's hypothesis, comparing gameplay result averages between the two prototype states - time is calculated in seconds and the others are the related number of screen touches. In another team member's analysis of the data, changes were highlighted through different tests to show a major difference between values (and thus going against the team's hypothesis). These can be seen in green.

The major difference the team noticed was the screen touch values in the first two levels, where values were higher in the intuitive state. It shows that players struggle more at the start of the demo when they have nothing to tell them how to use simple controls/mechanics. However, the learning curve balances out by the end of the game, as shown by the red averages in the above image. Relating with the similar time values, overall each player finishes the demo at the same skill level and in roughly the same amount of time.

	Yes		No	
	Tut	!Tut	Tut	!Tut
Did you reach the end of the prototype?	20	20	0	0
Did you ask for a supervisors help at any point?	0	1	20	19
Were the controls clear when you entered the game?	19	10	1	10
Do you feel like there should me more information at the start?	4	14	16	6
Do you think you could now use these mechanics in a puzzle?	20	20	0	0
Are the GUI arrow controls for this mechanic confusing?	2	3	18	17
Does it feel suitable for a mobile puzzle game?	19	20	1	0

Figure 7 - Comparison of Questionnaire Data

Again the questionnaire data was analysed through different test methods by another member of the team. With this set of data, however, the designer's job was to determine how more qualitative data could affect or improve the mechanics in the game.

The questions were designed so that opinionated answers would produce quantitative data that could be analysed, but stay true to the tester's thoughts. Some housekeeping questions were also given so that the team could see if people were completing the demo or not.

The data shows that every tester was able to complete the demo and that even in the intuitive state (without tutorials) only one tester had to ask a team member for assistance. Also the experimental mechanics were liked by the testers and were overall welcome in this mobile puzzle game. But the main focus of our research was the section marked as green, asking the player if they understood how to interact with the game. Although all testers made it through the entire demo, finishing with roughly the same time and touch values, the intuitive state players shown that they would have preferred more information to start with.

2.2.2 WHAT WAS MODIFIED

With the analysis complete, the team had to take what was learned and apply it to the gameplay. Luckily the analysis finished in the period of massive change to the games long-term design, offering the team tested ideations and, more importantly, an understanding of the limits of the player when pushing them to work things out. The results of the test were the defining pieces of information that helped the team decide to modify aspects of the game that were in limbo due to decisions relying on the teams opinions.

The first decision was based on the experimental mechanics and how they would affect Pathos in the long-term. All of the experiments were tied to the push/pull mechanic, a remnant piece of the original prototype from Protoplay, which in turn was a remnant project that was originally built specifically for the PlayStation Vita (Sony Computer Entertainment, 2011) mobile device. A big issue relating to this mechanic was that while the game moved towards touch-based gameplay, relying on pathfinding for movement, push/pull was only suited to analogue controls. While testers agreed that it fit into the game, and passive research into Lara Croft Go (Square Enix Montreal, 2015) shown it has a place even in mainstream mobile puzzle games, it was decided by the team that the mechanic didn't belong in the new Pathos. It was crucial to the game for research into player limits, but ultimately it was cut from the new design.



Figure 8 - Lara Croft Go Push Mechanic in Action (TombRaiders, 2015)

The second decision pushed the team as a whole to start creating more readable designs of mechanics and artwork to let the player focus on puzzle logic. The team understood that the direction that the game moved in was working, but the player was feeling frustrated because they had to take the extra step of working out what a basic mechanic actually was. The design shifted so that this step is either skipped or is much quicker to pass through, with artwork and slight bread crumbing to force the player to interact with something to understand it.

Finally, the last decision focused on creating more universal assets to be used across many levels. Building the prototype was relatively quick, but the team would need to increase the speed of development to achieve the level of output needed to release the game on. Art worked on producing platforms, interaction objects and paths to keep the same visual cues across the entire game, meaning the player would know exactly what something would do just by looking at it. Design focused on building modular assets and tools, using the art provided and adding code to allow a drag and drop system that allows rapid prototyping in both puzzle design (using a circuit building style) and in-engine white boxing. Building a modular system also allowed simple changes to be applied to all levels at once.

2.3 TEST EVALUATION

At the end of the analysis of the test results, the team had a secure direction to take the game so that both developer and player were happy with the result. The ability to get a large number of people to test what was learned after the huge Protoplay event was incredibly useful; it was easy for the team to assume that after tackling a competition that difficult, they were at the top level of development. The reality was that the first idea was not the best idea, where the experimental mechanics and remnants of past concepts were cut from the game, but it was much clearer to take feedback and use it to its highest potential.

Considering the research topic, the test results, even with the big differences at the beginning of the demo, concluded that the hypothesis was correct. The reason behind the research was to achieve the same result of an explicit tutorial without forcing the player to pull themselves out of their immersion to read lines of text. The hypothesis stated that the average results from each prototype state should match one another, and while the values weren't exactly the same, the analysis tests (applied by another member of the team) shown they were well within the tolerances of being defined as "similar". The team could then comfortably accept that explicit tutorials were not needed in the game.

The way that the team structured the research could have been different, however. The hypothesis and null hypothesis for the module were, conventionally, the wrong way around and it was highlighted in a discussion with module tutors. Trying to understand what was to be found from the research would have been much simpler and meaningful if the team were to prove something wrong rather than hope for something to be right. But the process of testing people and providing the gameplay in such a fashion that brought the most out of the test sessions was an achievement especially for the author, and gave the team experience in providing a quality game to many people which would be crucial knowledge for preparing the games release.

3.0 PROJECT DEVELOPMENT

3.1 DESIGNING WITH CONCEPTS

From the outset of securing the games original concept, it has been the task of the designer to build meaningful mechanics from the conceptual narrative of the game. The core concept was always to use the theme to boost the gameplay quality; Jesse Schell talks about this subject and offers two steps to achieve this goal (Schell, 2015, p. 59):

- ❖ **Step 1** – Figure out what your theme is.
- ❖ **Step 2** – Use every means possible to reinforce that theme.

Before moving onto his own experiences, he mentions "Most game themes are experience based; that is, the goal of the design is to deliver an essential experience to the player." (Schell, 2015, p. 59). The connection between theme and game relies on recreating the experience of the story you're trying to tell – and to return to Schell again, he once worked on recreating the experience of *being* a pirate and used peripherals and mechanics to do it.

Working with this in mind, the author worked on the level of depth that the game needed to achieve to recreate this experience for the player. But with the fragile nature of the narrative (following the rough journey of a child growing up in the foster care system), the experience had to be altered so that the player would be more of a guide rather than taking on the stress and fears of the character.

The main aspect that the team wanted to recreate was the feeling of challenge, and a popular genre of games that offer this exact thing is puzzles. Once this decision was made, a lot of research was put into the puzzle area of the project before any meaningful work was produced.

3.2 PUZZLE DESIGN

3.2.1 RULES OF PATHOS

Although designing and building puzzles was a new concept to work with for the author, the team knew what design goals they wanted to achieve at the very start of the project. To create a game that meant something and had something different to offer to players, a set of rules, or pillars, were created to hold the foundation of the project:

- ❖ **Accessibility** – Simple controls, quick access to gameplay, casual difficulty
- ❖ **Simplicity** – Focus on logic not mechanics, distractions kept to a minimum
- ❖ **Pressure-Free** – No fail states, move at own pace, learn to play over time
- ❖ **Balanced Theme** – High-level narrative is not forced, gameplay matches theme

Everything built for Pathos had to be scrutinised by these pillars before being accepted into the game, and the team built up an understanding of what the game was. However, this proved to make bringing in new members, or explaining the concept of the game in a simple manner, difficult.

3.2.2 PUZZLE THEORY

Previous to the Research Methods module, the author conducted extensive research into puzzles in video games. Looking at developer interviews and video game reviewers to understand not only the process of what it was like to create puzzle games, but also how it feels to play them. This knowledge built a foundation on which to start developing early puzzles and mechanics before moving onto more academic research.

Some of the summations found have been areas that the design has tried to challenge:

- ❖ Puzzles always have a dominant strategy
- ❖ Puzzles cannot be replayed
- ❖ Players and designers love unexpected solutions
- ❖ Puzzles shouldn't have misleading paths

Each point was built from the author's opinionated view of multiple sources found around the internet; however, the important articles are located in this papers bibliography.

Following this personal research, the module requirements of a literature review let the author expand on key areas of interest and apply academic research into them (see Appendix A).

When the research began, the author chose to pursue the vague definition of what a puzzle is, as this topic alone offers much to debate. Schell, using his own research, found that most debates ended in agreement that puzzles are some form of game with a correct answer or dominant strategy (Schell, 2015, p. 241). But interestingly the author found that looking in the opposite direction was equally important – are games themselves just puzzles? (Salen & Zimmerman, 2004, p. 81)

Following this discovery, the research branched off into classification of puzzles and puzzle games in particular, and also into the process of creating them. This was crucial to developing a better core concept for the game because there was now a process that worked for the author. Finally, extra research was conducted into areas such as:

- ❖ Teaching the player
- ❖ Letting the player feel smart
- ❖ Progression of difficulty
- ❖ Linearity in gameplay

These topics covered other areas of interest for the author, and helped build a knowledge base for future efforts on the project.

3.2.3 PUZZLE MECHANICS

After working on the theory for a while, the next step was to apply it to the process of puzzle game creation and work on the mechanic concepts made earlier in the project. The way the designer builds the games chapters (a collection of puzzles in one area of the game world) relies on an ongoing process of designing new mechanics.

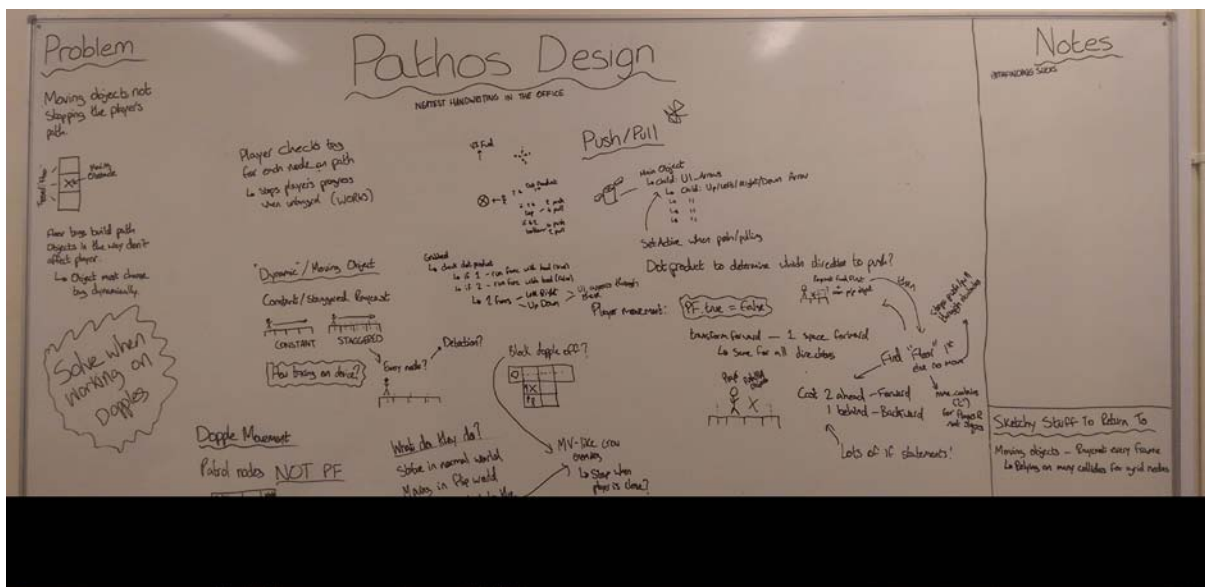


Figure 9 - Initial Work on Early Mechanics

With earlier mechanics concepts, the prototypes the team had created in the past had shown them to be completely separate from each other, not allowing the weaving of logic between them. The first task was to rebuild a new set of mechanics and use old prototypes to understand where the team went wrong but also to stay true to the original concept.

The author knew that the game had its main mechanic of flipping both player and character perspective through a rotation of mobile devices. However, it never had a place as being the only core on which every other mechanic relied on. To create this better core, the design had to look at the narrative again to build something that really suited the game, and thus the aim of Pathos became clear – the player would always be creating a path for the character. It fit with the theme of the game by recreating the experience as the team wanted to from the beginning.

From this many ideas branched out to offer overlapping gameplay features and built a set of broad avenues to use when adding anything new:

- ❖ **Interaction** – Levers, cranks, pressure plates etc.
- ❖ **Inventory** – Keys, machine parts, rune stones etc.
- ❖ **“Enemies”** – Non-violent characters and denizens
- ❖ **Animals** – Friendly characters with their own unique mechanics

Ultimately these four core mechanics allow the design to branch out in many different directions, owing to the information found during the authors research phase.

3.2.4 LEVEL DESIGN

Level design in the previous prototypes was more challenging because both mechanic generation and level building were done at the same time. But with the new concept, the tools were built previous to the level design so that a drag and drop system could be used in both the designs and building in-engine.

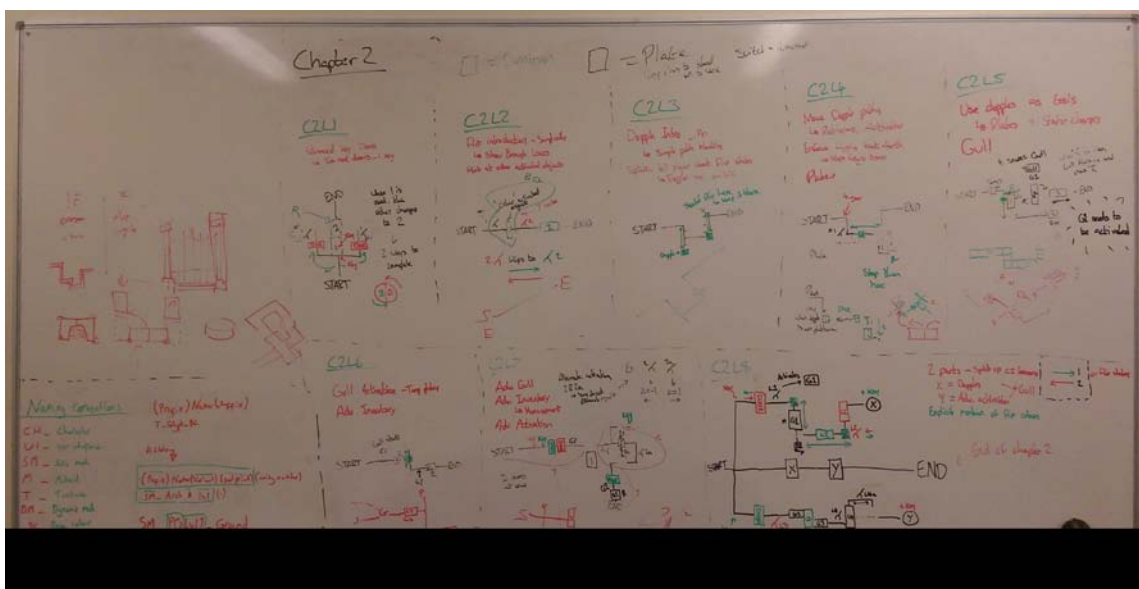


Figure 10 – Chapter 2: From Design to Art

The outset of producing designs for puzzles proved to be difficult as there is no easy way of communicating puzzle logic to other members of a team. The team managed to get through this by working on a circuit style system (seen in figure 10) which has proven useful in electrical engineering for many years. It works with the designs system of drag and drop because a key is used to show what each symbol means and what it does. Connecting lines show the flow of logic and when each piece of the puzzle should be solved.

From these diagrams, the environment artist could then build layouts of the levels and frame them accordingly to allow a suitable art style to be applied. As each level has a fixed camera, each level pushed through the design>layout>art pipeline needed to be carefully crafted to work in the frame of a mobile device screen.

3.3 SYSTEM PROGRAMMING

3.3.1 OUTSET ISSUES

Through both the Research Methods and Major Project modules, the team did not have a dedicated programmer. It was the role of the designer to take up two major disciplines of game development to build the game as they were the only member of the team capable of scripting in a game engine. The team could have actively searched for a programmer; however, previous attempts before taking part in the Dare to be Digital competition had proven that it was much more difficult than imagined. The process of introducing a new member to a project of this size was more trouble than it was worth in the end.

3.3.2 PATHFINDING

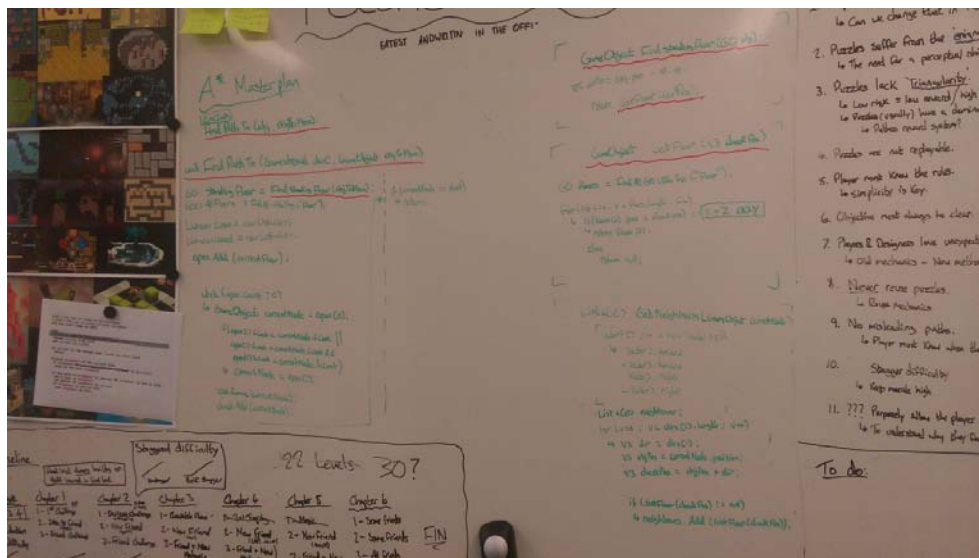


Figure 11 - Hand-Written A* Pathfinding Code

A major problem plaguing the project from the first mobile prototype was the way the player would move in the environments. After using a placeholder pathfinding script for the testing demo, the author had sufficient knowledge of scripting to begin development on a custom pathfinding system that would allow the team to do whatever was needed in the future. This is one of the biggest achievements for the team, in terms of development, as the entire experience of gameplay relies completely on this system.

3.3.3 DESIGN IN MIND

To focus on building level designs with tools, each script had to work in a modular fashion. Hard-coding would be kept to a minimum to allow objects to easily speak to each other and exchange values. Each major component of the game should work independently of everything else. With Unity (Unity Technologies, 2005), the game engine used by the team, prefabs (prefabricated game objects) allowed work to be focused on one object and any changes made would apply to every instance of that object. This is used for every object in the game with the exception of unique pieces within levels.

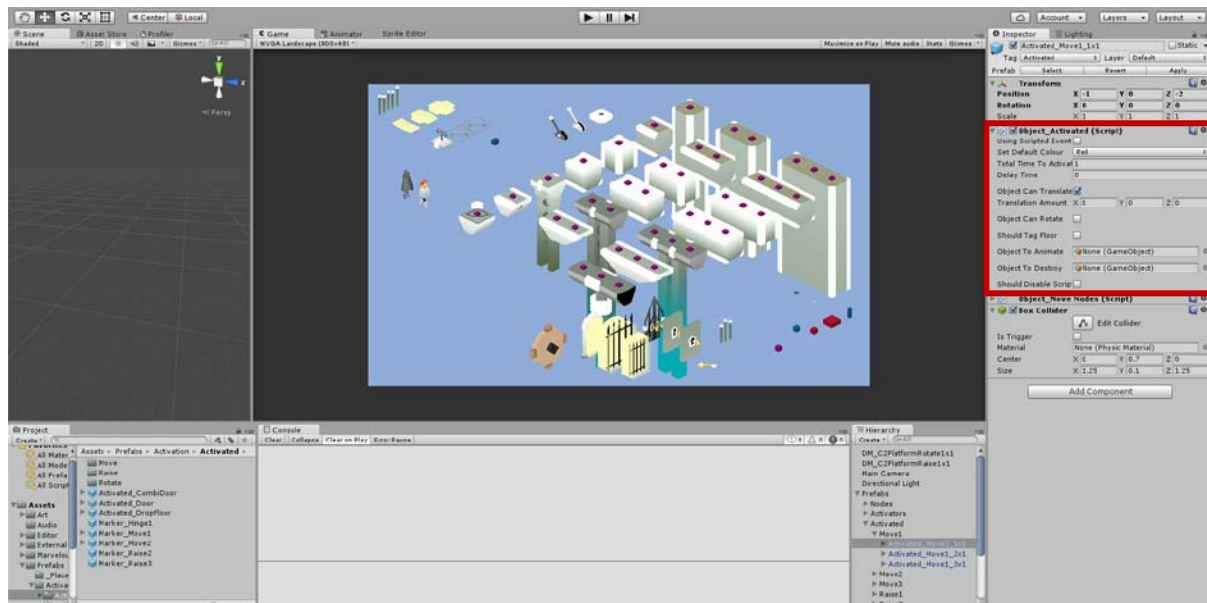


Figure 12 - Drag & Drop Prefabs with Custom Inspector in Unity 3D

With prefabs such as activated objects, too many variables were exposed at one time. To tidy the inspector and improve work efficiency, a custom inspector script was created to only show information that was requested.

3.3.4 WORKING WITH MOBILE

To work with mobile platforms, the previous methods used in the project to script gameplay behaviours had to be completely changed. With much less processing power than a PC or console platform, logic could only be called upon when it was needed. The process of learning programming principles for mobile (in particular with C#) was long, but worthwhile as the information would work for efficient coding on any platform.

3.4 GAME CONSTRUCTION

3.4.1 LEVEL BUILDING IN ENGINE

When finally building levels in the game engine, the collaborative effort of work is funnelled through the designer and is assembled as final assets. Art, animation, design and code form together to create assets such as levers, moving platforms and animated characters. Prefabs are created with 3D meshes, animations and animator controllers are added to the meshes to prepare for the code to make it all function.

3.4.2 BRANCHING OUT

The task of the designer/programmer continues into the design of user interface (UI) and audio systems. To create a prototype to represent Pathos until the games release, every aspect of the game needed some form of representation. Because of this a UI that covered the main menu, level select and gameplay UI (GUI) was created, although the final plans for the official content had not been drawn up. However, the GUI layout design had been finalised as research and work was produced to create an effective interface that would not impede gameplay by getting in the way.



Figure 13 - Representative User Interface

Audio is also a big part of the puzzle experience, offering ambiance, feedback and relaxing music to build atmosphere and show the player the effect of their actions. A music manager exists in the game, fading out currently playing music as one scene ends and fading in new music as another begins. Some UI elements also have audio feedback and more sound effects would be used in the future of the games development.

3.5 PROJECT MANAGEMENT

3.5.1 PLANNING

Daily scrums kept morale and creativity at their peaks, with the ability to get and give feedback on the previous day's work to push quality. Because the plan covered a long period of time, the team decided to keep it broad with the ability to change at any time. Coloured notes let the team pick and choose where they wanted key dates to be and Trello (Fog Creek Software, 2011) allowed each member to apply and see updates from the whole team.

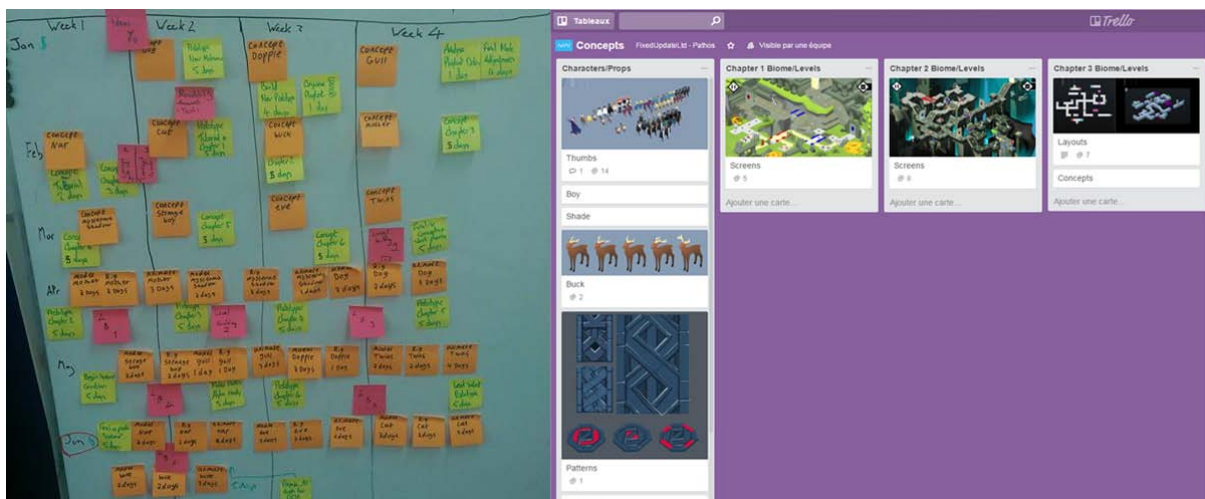


Figure 14 - Methods of Project Planning

The structure of the team had been tried and tested over a year going through many different issues and solutions; the process of planning is kept small because the team works better when using a dynamic system. Processing and finalising a whole chapter of levels in the game has proven that a specific plan could not be created. The result of this is the direction the team has taken for planning in general. The team also occasionally received deadlines from an external publisher, keeping the bigger picture relevant in the time period of the Major Project module.

3.5.2 WORKFLOW & PIPELINES

Working in a small team is supplemented by the ability to work independently (not including input on each other's progress) until the collection of all assets at the end of a milestone. This means that only one team member only needs to be working one piece of work at any time, without the need of merging regularly.

Work begins at the design stage; the author creates puzzle logic in the form of circuits. The circuits are then explained and handed off to the environment artist who then creates a layout based on safety zones (judged by GUI and resolution scaling) and concept art. While at this stage, mechanics are prototyped and coded into prefabs. The layout is then handed back to the designer and both team members work separately on their subject area. After building the white boxes and prototype mechanics, the scene is handed off to the environment artist again to place background art assets and create sufficient lighting. At milestones determined by the project plan, the character artist hands off character models and animations to the designer. The animations are then implemented and the process of creation is complete.

This worked well for the team as individuals were never waiting with nothing to do; this allowed a lot of polish to be put into the game early in the development process.

3.5.3 VERSION CONTROL

During the beginning of the Major Project module, the team attempted to use version control for the project. A chain of events occurred that made using version control useless for the game.

Perforce (Perforce Software, 1995) is a popular source control platform used by many development companies of different industries. For the team, however, it created issues with Unity project files that altered important folders and created access problems when reopening projects in the engine. Alongside the issue of access permissions in the teams university office, a lot of time was wasted trying to use a system that didn't work well from the beginning.

The pipeline used for the game gave the team no reason to rely on version control as it was much quicker and efficient to use manual methods. Google Drive (Google, 2012), HipChat (Atlassian, 2010) and hard drive transfers were more efficient and file backups were always made before assets were placed in project files.

4.0 REFLECTION

4.1 TEAM REFLECTION

As FixedUpdate has been a team for over a year, each individual knows how one another works and can play towards strengths and weaknesses. But there are issues that create problems in the team dynamic; such as an imbalance of responsibilities and a drop in effort over long periods. On a lengthy project with a small team, morale plays a big part in the quality of the work and because of these issues the game can't be as good as it should be.

Over the period of the Major Project module, the team has finally learned where they want to take Pathos. Since the testing phase, each member knows exactly what they should be doing and what will be coming next.

4.2 INDIVIDUAL REFLECTION

During the Major Project module, the authors own mindset for the game changed. Before testing, being in a position where a large portion of the responsibility for the game was rested made the author overwork, and eventually drop the quality of both design and programming disciplines. But after working with testers and seeing what people thought of the game, the author calmed down and produced work at a slower, but higher quality, pace. Working on a full game project and tying its development into the Major Project module meant basic assets and systems were not in place for the final deliverable, such as full audio and save/load systems. But ultimately the author was happy with the deliverable as it represented a true vertical slice of the game.

5.0 CONCLUSION

5.1 MAJOR PROJECT

Work on the project began long before the Major Project module, but it wasn't until the team began working to both a publisher and a module that the game started to take shape. The biggest bonus of the module was that it required research to be conducted in a subject relating to the project, and that helped build an important knowledge base to draw upon when not only creating Pathos, but any game in the future. Testing in the module was incredibly useful as the game had undergone many changes since the feedback was gained from Protoplay. While working on a full game and tying it to modules proved difficult in some cases, the game benefitted from the experience and so did the team.

5.2 CONTINUING OVER SUMMER

The journey of Pathos does not end with the Major Project module. Working towards a November 2016 release, the team has to create four more chapters of levels and the systems to collect them all into a mobile game environment (level select, saving, loading etc.). The summer following the Major Project consists of more work seen in the module, but with the added benefit of tried and tested methods of creating content that players will like.

6.0 APPENDICES

6.1 APPENDIX A: LITERATURE REVIEW

2.0 Introduction

For this module my team and I have been working on the introduction to our mobile puzzle game, which itself is a continuation of a successful Dare to be Digital 2015 project. Being able to use our game in this module allows us to understand how to effectively teach the player how to use core mechanics, but in a way that doesn't make us spoon-feed information to them. This also doubles as a test to see if these mechanics actually have a place in the game or not.

Working with two artists, it is my job as a designer to create the mechanics to use in a puzzle environment, while keeping in line with the game's theme and narrative design. With such a small team we have to fill multiple roles, therefore it is also my job to create all of the scripting/programming involved with the game's systems and mechanics. This allows me to work directly in-engine with things I have designed on paper, which in turn shows me what sort of limitations I can work with.

While there will not be any puzzles in the first deliverable, the feedback we will receive from it will help introduce puzzles in the final deliverable, and good design is needed to provide everything a puzzle needs to entertain them. With this in mind, this report will focus on puzzle design specifically in video games – identifying how puzzles are used in this medium, what makes up a puzzle in a video game and what the player's experience involves. Writing the report this way will give me the most amount of research into areas that will help me build puzzles for use in the final game after this module has concluded.

2.1 Research Methods

While researching puzzle theory and how game developers have done it in the past, I've found that game design books have been the most helpful. As puzzle design has no certain way to do things, older text is just as useful as newer text - puzzles are a timeless form of entertainment and video games have used them since the beginning. I've also made use of a small amount of web pages that include interviews with game developers who have recently worked on successful games involving puzzles.

3.0 Literature Review

Jonathan Blow lays out a good starting point when it comes to creating puzzles in video games (Davies, 2015):

"There are definitely techniques involved, I'm not completely sure I could lay out for you what those are, because it's an art, not a science."

3.1 Puzzles to Video Games

As with most information about puzzles, there is a blurred line to cross when defining what a puzzle actually is. Multiple game developers have debated this topic, with a focus on their field of expertise, leading to a difference in opinions. What is agreed on is that all puzzles are some form of a game with a correct answer or dominant strategy (Schell, 2015, p. 241), and that it is also useful to reverse the thinking – are games themselves just puzzles? (Salen & Zimmerman, 2004, p. 81)

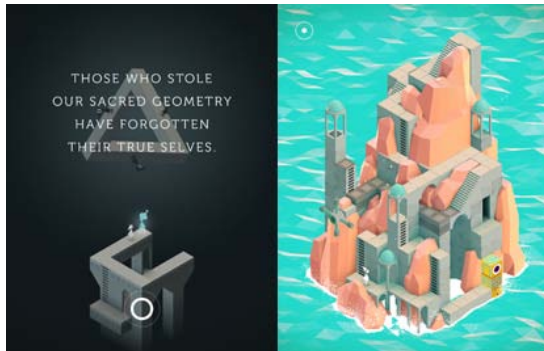


Figure 15 - Monument Valley (TabTimes, 2014)

With that sort of definition to work from, it is easy to see a connection with games as they tend to be one and the same. This helps when adding video game elements to puzzles, or vice versa. An example of this would be storytelling, which was recently used in the successful mobile game 'Monument Valley' (Ustwo, 2014). In the game the player solves puzzles to find piece of geometry, as they find more pieces more story is given to them. Following on from this, developer of many text-adventure games, Steve Meretzky,

was once asked if he would ever “want to forget about the puzzles” to focus on story, in which his reply was “My desire. . . is to do more story and less puzzle, but puzzle is necessary to keep that thirty-to forty-hour playtime goal.” (Rouse III, 2004, p. 198).

3.2 Fundamental Design

3.2.1 Classifying the Game

According to Dev.Mag, a magazine created by the South African game development community, states that puzzle games can be categorised in 3 different ways (Tulleken, 2011):

- ❖ **Procedural** – Computer generated puzzles with core mechanics.
- ❖ **Combinatorial** – Core mechanics are reused in new ways.
- ❖ **Heterogeneous** – New mechanics per puzzle.

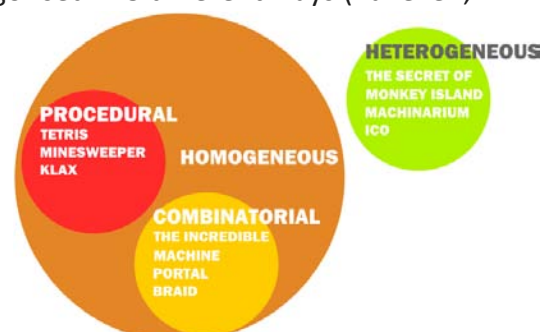


Figure 16 - Classification by design (Tulleken, 2011)

Once broadly classified, the type of puzzle that can be created has its own classification based on what the designer wants to achieve through gameplay (Saltzman, 2002, p. 169).

- ❖ **Environmental** - Puzzles involve analysing and altering the player’s surroundings, an example would be ‘Legend of Zelda: Ocarina of Time’ (Nintendo EAD, 1998) where the player would have to light torches to open the exit door.
- ❖ **Inventory** - Puzzles involve objects in the player’s possession, and is often used alongside environmental puzzles. Pieces of a puzzle would be picked up and placed

by the player, which leads to more possibilities of interaction. An example of this would be 'Day of the Tentacle' (LucasArts, 1993), by placing a bottle of wine into a time machine the player can pick up vinegar in the future.

- ❖ **Conversation** – Puzzles entirely contained within a dialogue system, by saying the right phrase or sequence of answers, the player can progress. An example of this would be 'John Saul's Blackstone Chronicles' (Legend Entertainment, 1998), where the player would talk to ghosts to find their kidnapped son.
- ❖ **Irrelevant** – Puzzles used in games that do not relate to its core mechanics in any way, making them completely irrelevant to the task at hand. An example of this would be 'The Neverhood' (The Neverhood, Inc., 1996), where the player would solve a sliding tile puzzle in one instance, and later play a game of "Concentration" in another.

3.2.2 Process of Creation

Game developers have their own processes they will follow to implement puzzles, not all use the same methods but ultimately there are overlaps between each process, below is a broad representation of those overlaps.

The first step is to think about game mechanics, how will the player interact with your game? Ted Lauterbach, creator of 'suteF' (Lauterbach, 2010), states that his puzzle design is "always dictated by the mechanics" (Dev.Mag, 2011), making it first on his list when creating a puzzle. With puzzles relying on mechanics, it is useful to make them flexible, as Teddy Lee says in his interview "A rule of thumb I follow is if I can think up five or more puzzles on paper with a single mechanic, then there's a good chance I can come up with plenty more later on." (Dev.Mag, 2011).

What is the player actually supposed to achieve?

A good puzzle must show this goal clearly to the player so they can easily understand the rules, this will help them get started as stated by 'Q.U.B.E' (Toxic Games, 2011) developer Dave Hall (Louw, 2011). Schell continues on from this and explains that if a player cannot understand their goal, ultimately they won't know what they are supposed to be doing, leading to a loss of interest. Figuring out what the goal is can be



Figure 17 - suteF always showing the player's goal (Riggan, 2011)

part of the puzzle and actually be fun for the player, but those types of puzzles are risky as generally only "die-hard" puzzle fans will find them fun (Schell, 2015, p. 243) – though George Broussard, a man responsible for early Duke Nukem games, mentions his example of a "good puzzle" in which the goal wasn't immediately obvious to him until he finally found it "placed literally" at his feet (Saltzman, 2002, p. 172).

Returning to 'suteF', Lauterbach talks about the next stage of the process - "Once they (goal and start point) are in place, I'll put what I feel is an appropriate amount of obstacles between them and the goal." What an appropriate amount is depends on the designer's

preference and what level of difficulty they want to create, as he mentions with a final point, “Mostly it’s a feeling I go off. . . . It’s awfully hard to explain it” (Dev.Mag, 2011).

3.2.3 Building Mechanics as Tools

As defined by Lorne Lanning, creator of the Oddworld series, a mechanic is “anything that can be done in the game” by the player. He goes on to talk about how a combination of these basic tools create chemistry, and that combining these elements should be fun for the player to be involved with. In creating a fun chemistry, building puzzles then becomes a simple case of laying out the environment of play. He also goes on to mention that once players understand how to use the basics, you can give those more to work with. If this chemistry is created well, there is an experience that won’t overwhelm players and teaches them how to apply these mechanics to solve puzzles (Saltzman, 2002, p. 170).

3.3 Player Experience

3.3.1 Teaching the Player

As with any tool for any application in any industry, the user must first learn how to use it, but without a way to actually learn the user is effectively being punished for not knowing. In an analysis of ‘Loom’ (Lucasfilm Games, 1990), Rouse III mentions that traditional adventure games would make players play many times before they achieved the ultimate goal of receiving a reward, but ‘Loom’ was “made from the start to be an easy game”. He goes on to say that the

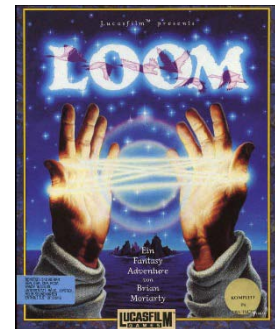


Figure 18 - Loom (Lucasfilm Games, 1990)

game actually wants players to succeed so that they can see the end of the story (Rouse III, 2004, p. 233).

Returning to Broussard, his way of building “good puzzles” rely on the fact that they don’t require big leaps in terms of logic to solve them. The puzzle must make sense in the world it is in, which relate back to the mechanics that are taught to the player – how will a puzzle make sense if the player doesn’t know how to interact with it? Finally a puzzle must allow experimentation by the player, giving those tools and teaching them how to use them will allow focus to be directed at playing with them, instead of moving backwards by trying to understand them thus taking the attention away from the puzzle (Saltzman, 2002, p. 171).

3.3.2 Progression of Difficulty

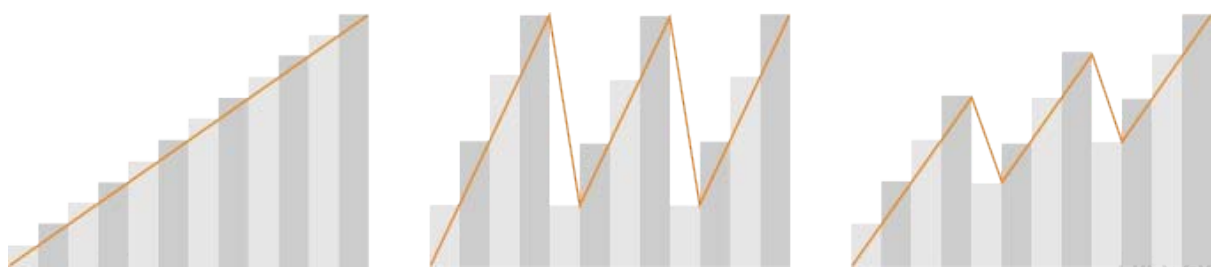


Figure 19 - Examples of different difficulty profiles for a sequence of 12 puzzles (Tulleken, 2011)

Guy Lima, a developer who worked on the popular student puzzle game 'Continuity' (Ragtime Games, 2010), talks about how his team kept the players motivated while playing through their game – “if you presented another level that looked harder after. . . . a difficult level, he or she would just lose interest in continuing” (Tulleken, 2011).

An example of an alternate difficulty profile is used in 'Q.U.B.E' - new mechanics are introduced on their own to provide a quick and simple way of teaching the player without the need to understand many things at once, relating to section 3.3.1, and then moves back into complex puzzles. This staggered approach brings the pace down for the player allowing them to rest between difficult puzzles by learning more mechanics for the inevitable harder puzzle coming next (Louw, 2011).

3.3.3 Linearity in Gameplay

The problem with puzzles in general is that they cease to be engaging or fun once the dominant strategy has been found (Schell, 2015, p. 241). To get around or soften the blow of this, developers can use different elements of gameplay to provide more choice for the player (Rouse III, 2004, p. 120):

- ❖ **Multiple Solutions** – Well designed challenges will allow the player to come up with their own solutions. Not every player will use the same process, multiple ways of completing the challenge will allow almost anyone to overcome it, if the solutions are reasonable.
- ❖ **Order** – Designing puzzles in a way that allows the player to choose the order of the challenges they face will significantly take away linearity. A lot of adventure games froze the player's progression if they were stuck at a single challenge, due to the nature of their linear design.
- ❖ **Selection** – Not to be confused with “Order”, selection allows the player to pick what challenges they face. Instead of presenting them with a set of challenges that they must overcome to progress, the player is in control of their own progression.

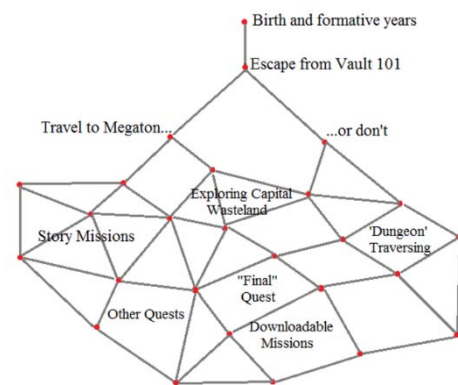


Figure 20 - Non-Linear structure of Fallout 3 (Shepard, 2014)

3.3.4 Letting the Player Feel Smart

A short final point by continuing from giving the player a choice, designing puzzles in a way that let the player feel smart by solving them. Anticipating the player's actions in the design stage is beneficial not only to the polish of the technical side, but it is also a nod to the player so that they know they're on the right track (Rouse III, 2004, p. 116).

One thing that crossword puzzles take advantage of is redundancy – a method of leaving out a piece of information that can be understood using the context around it. This kind of method skirts the edge of emergent gameplay, fitting in the middle of highly rigid structures

and player freedom. In doing this the player gets to use their own logic to fill in the blanks and apply it to the challenge in front of them (Salen & Zimmerman, 2004, pp. 198, 199).

4.0 Conclusion

The aim of writing this paper was to produce a broad process to run through when thinking about designing a puzzle for a video game. This was achieved by focusing on key areas within this process, such as puzzle classification and player experience, and taking data from respected game developers, many considered industry veterans. These developers share the same thought process that will hopefully be used in the final deliverable.

To reiterate a point made at the beginning of the review, Jonathan Blow gave the best explanation based on the information that has been gathered here, “it’s an art, not a science” (Davies, 2015). There is no sure way to do things when creating puzzles, only principles that have worked for game developers in the past. The most important things to take away from this is that for a puzzle to be considered “good” or “well designed”, it must challenge the player without the need to explain everything and it must not allow them to get stuck. Progression, even giving the player the ability to pick and choose it themselves, must always be considered in every puzzle or challenge.

With a lot of theory gathered, future research will be aimed at case studies – what games actually follow this theory, if not why, and is it going in a similar direction to the final deliverable? Mainly aimed at mobile puzzle games, it will be useful to know how far mechanics can be taken and what kind of level of experimentation is involved with the player’s knowledge of these mechanics. Once the player can experiment with the tools at their disposal, complex puzzle systems can be built to an appropriate level of difficulty.

Push/Pull Mechanic:

Are the GUI arrow controls for this mechanic confusing? Yes | No

Does it feel suitable for a mobile puzzle game? Yes | No

Do you have any other feedback on the “Push/Pull” mechanic?

Lastly, do you have any other feedback about the game in general?

Please fill in the next page and hand this back to your supervisor.

Fill in the circle(s) with '1' being totally disagree and '10' being totally agree:

I felt confused playing this prototype:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

I understood what my overall objective was:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

I understood what the mechanics (what I interacted with) did:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

The mechanics fit a mobile puzzle game:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

I would enjoy using the mechanics to solve a puzzle:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

I would tell my family/friends to be interested in this game:

Totally Disagree 1 2 3 4 5 6 7 8 9 10 Totally Agree

BIBLIOGRAPHY

- Abertay University, 2000. *Dare To Be Digital*. [Online]
Available at: <http://www.daretobedigital.com/>
[Accessed 27 April 2016].
- Atlassian, 2010. *HipChat*, Sydney: s.n.
- Davies, M., 2015. *A Good Puzzle Game Is Hard To Build*. [Online]
Available at: <http://www.rockpapershotgun.com/2015/01/22/how-to-make-a-puzzle-game/>
[Accessed 1 January 2016].
- Dev.Mag, 2011. *How are puzzle games designed? Ted Lauterbach*. [Online]
Available at: <http://devmag.org.za/2011/05/09/how-are-puzzle-games-designed-ted-lauterbach/>
[Accessed 1 January 2016].
- Dev.Mag, 2011. *How are puzzle games designed? Teddy Lee*. [Online]
Available at: <http://devmag.org.za/2011/05/23/how-are-puzzle-games-designed-teddy-lee/>
[Accessed 1 January 2016].
- Fog Creek Software, 2011. *Trello*. [Online]
Available at: <https://trello.com/>
[Accessed 27 April 2016].
- Google, 2012. *Google Drive*, Menlo Park: s.n.
- Lauterbach, T., 2010. *suteF*, Madison: Ted Lauterbach.
- Legend Entertainment, 1998. *John Saul's Blackstone Chronicles*, Chantilly: Mindscape.
- Louw, V. W., 2011. *How are puzzle games designed? Dave Hall*. [Online]
Available at: <http://devmag.org.za/2011/05/16/how-are-puzzle-games-designed-dave-hall/>
[Accessed 1 January 2016].
- LucasArts, 1993. *Day of the Tentacle*, San Francisco: LucasArts.
- Lucasfilm Games, 1990. *Loom*, San Francisco: Lucasfilm Games.
- Nintendo EAD, 1998. *Legend of Zelda: Ocarina of Time*, Kyoto: Nintendo.
- Perforce Software, 1995. *Perforce*, Eden Prairie: s.n.
- Ragtime Games, 2010. *Continuity*, Gothenburg: Ragtime Games.
- Riggall, J., 2011. *A challenging and atmospheric platform game*. [Online]
Available at: <http://screenshots.en.sftcdn.net/en/scrn/305000/305372/sutef-3.jpg>
[Accessed 1 January 2016].
- Rouse III, R., 2004. *Game Design: Theory & Practice*. 2nd ed. Plano: Wordware Publishing.
- Salen, K. & Zimmerman, E., 2004. *Rules of Play: Game Design Fundamentals*. Cambridge: MIT Press.
- Saltzman, M., 2002. *Game Design: Secrets of the Sages*. 3rd ed. Indianapolis: Pearson Education.
- Schell, J., 2015. *The Art of Games Design: A Book of Lenses*. 2nd ed. Boca Raton: CRC Press.
- Shepard, M., 2014. *Interactive Storytelling - Narrative Techniques and Methods in Video Games*. [Online]

Available at: <http://scalar.usc.edu/works/interactive-storytelling-narrative-techniques-and-methods-in-video-games/linear-and-non-linear?path=narrative-styles>

[Accessed 1 January 2016].

Sony Computer Entertainment, 2011. *PlayStation Vita*, Tokyo: Sony.

Square Enix Montreal, 2015. *Lara Croft Go*, Montreal: Square Enix.

TabTimes, 2014. '*Monument Valley*' Review: *Commemorating Magnificence*. [Online]

Available at: <http://tabtimes.com/wp-content/uploads/ckfinder/userfiles/images/monumentvalley3.jpg?15bab0>

[Accessed 1 January 2016].

The Neverhood, Inc., 1996. *The Neverhood*, Los Angeles: Dreamworks Interactive.

TombRaiders, 2015. *Stella's Lara Croft Go Walkthrough*. [Online]

Available at: <http://tombraders.net/stella/walks/LCGOwalk/screenshots/closing-door-7.png>

[Accessed 27 April 2016].

Toxic Games, 2011. *Q.U.B.E.*, Newport: Toxic Games.

Tulleken, H., 2011. *How Are Puzzle Games Designed? (Conclusion)*. [Online]

Available at: <http://devmag.org.za/blog/wp-content/uploads/2011/06/difficulty.png>

[Accessed 1 January 2016].

Tulleken, H., 2011. *How are puzzle games designed? (Introduction)*. [Online]

Available at: <http://devmag.org.za/2011/04/16/how-are-puzzle-games-designed-introduction/>

[Accessed 1 January 2016].

Tulleken, H., 2011. *How are puzzle games designed? Guy Lima*. [Online]

Available at: <http://devmag.org.za/2011/05/02/how-are-puzzle-games-designed-guy-lima/>

[Accessed 1 January 2016].

Unity Technologies, 2005. *Unity 3D*, San Francisco: s.n.

Ustwo, 2014. *Monument Valley*, New York: Ustwo.