Aalborg University Copenhagen

Semester: MED10

Title: Adaptive Tutorials for Complex Mechanics in Video Games

Project Period: Spring 2017

Semester Theme: Master Thesis

Aalborg University Copenhagen
Frederikskaj 12,
2450 København S, Denmark

Semester Coordinator:

Secretary:

Supervisor(s): Olga Timcenko

Members:

Mathias Hjørne Fynbo

Bo Sejer la Cour

Frederik Schmidt Knudsen

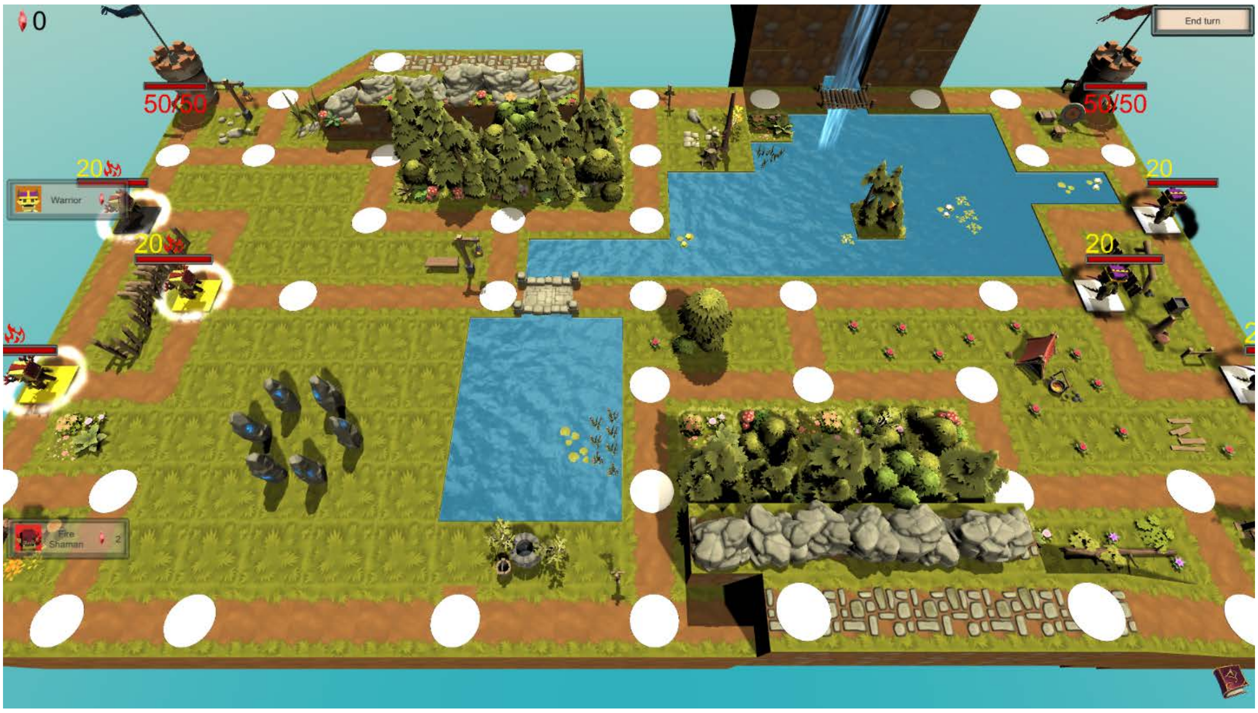
Abstract:

This project looks at adaptive tutorials in video games; a field that has not been done much work in, and could produce interesting results, in terms of enhancement of player experience.

We attempt to outline what an optimal framework for an adaptive tutorial system would look like, and implement a version of this with a more appropriate scope. This will be implemented into a game, which is created for this specific purpose.

The system aims to dynamically display tutorial messaging of complex mechanics in this game, so that inexperienced players will receive more information than experienced players, and vice versa, while still retaining an equal level of knowledge for both groups at the end.

After a series of iterative tests were run on the system, both online and in person, it was concluded that the system has potential to work if the system focuses on simple mechanics rather than complex.



ADAPTIVE TUTORIALS FOR COMPLEX MECHANICS IN VIDEO GAMES

Bo Sejer la Cour – 20125473

Frederik Schmidt Knudsen – 20125335

Mathias Hjørne Fynbo - 20125343

1 TABLE OF CONTENTS

2 Introduction ... 4
2.1 Motivation ... 4
2.2 Project Goal ... 5
2.3 Report Outline ... 6
3 What is a tutorial? ... 7
3.1 Tutorial Types ... 7
3.1.1 Text Boxes ... 9
3.1.2 Highlights ... 10
3.1.3 Action Lock ... 11
3.1.4 Exposition ... 12
3.1.5 Dialogue Tutorial ... 13
3.1.6 Training Room ... 14
3.1.7 Contextual ... 15
3.1.8 Player Selected ... 16
3.2 Creating a Good Tutorial ... 17
3.2.1 Ernest Adams ... 17
3.2.2 Extra Credits ... 18
3.2.3 George Fan ... 18
3.2.4 The Impact of Tutorials on Games of Varying Complexity ... 19
3.2.5 Takeaways ... 20
4 State of the Art ... 21
4.1 Minecraft ... 21
4.2 Desktop Dungeons ... 22
4.3 Plants vs. Zombies ... 23
5 Adaptive Methods ... 25
5.1 Player Experience Modelling (Data gathering) ... 26
5.2 Fitness Function (Evaluation function) ... 27
5.2.1 Examples of How Fitness Functions Can be Used ... 27
5.3 Adaptive Tutorial ... 28
6 Game Design ... 29
6.1 Turn-Based Strategy ... 29
6.1.1 Turn-Based Competition ... 29
6.1.2 General Game Mechanics ... 29

6.2 The Game Design ... 31
7 Dynamic Tutorial ... 34
7.1 The System ... 34
7.1.1 Preliminary Game Test ... 34
7.1.2 Data Gathering ... 35
7.1.3 Data Analysis ... 36
8 First Iteration ... 39
8.1 Tutorial Design ... 39
8.1.1 First Iteration ... 40
8.2 First Preliminary Test ... 41
8.2.1 Test Procedure ... 41
9 Second Iteration ... 43
9.1 Tutorial Design ... 43
9.2 Data Gathering ... 44
9.3 Second Preliminary Test ... 44
9.3.1 Additional Test Procedure ... 45
9.3.2 Results and Changes ... 46
10 Final Iteration ... 48
10.1 Final Tutorial Design ... 48
10.2 Final Test ... 49
10.2.1 The Overall Test Goals ... 49
10.2.2 Test Measurements ... 50
10.2.3 Test Procedure ... 50
11 Evaluation ... 53
11.1 Methods ... 53
11.1.1 Quantitative ... 53
11.1.2 Qualitative ... 53
11.2 Points of Interest ... 53
11.3 Problems and Bias ... 54
11.3.1 Distribution Problem ... 54
11.3.2 Survey misunderstandings ... 54
11.4 The Results ... 55
11.4.1 Participant Self-Assessment ... 56
11.4.2 Participant Knowledge ... 56

11.4.3 Differences in the Knowledge About Shadow Units 60
11.4.4 Differences in Unit Type Knowledge 61
11.4.5 Player Data Analysis 63
12 Discussion 65
12.1 Text Boxes 65
12.2 Make Sure the Information is Relevant 65
12.3 Information Should Come at the Correct Time 65
12.4 Displaying Messages Once 66
12.5 Problems Using Complex Mechanics 66
12.5.1 Players Need Confirmation in Their Assumptions 66
12.5.2 Tactics are Hard to Distinguish 67
12.5.3 Cross-game Similarities Can Confuse Players 67
12.6 The System 68
12.6.1 Innate Problem of the System 68
12.6.2 The Game Mechanics 69
12.7 Strategic Element 70
13 Conclusion 71
14 Future Work 72
14.1 Go Simple 72
14.2 Reminder System 73
15 Bibliography 74
16 Appendix 77
16.1 Appendix 1 – Games We Played 77
16.1.1 Games Specific to the Turn-Based Genre 77
16.1.2 Games Outside the Turn-Based Genre 78
16.2 Appendix 2 – First Questionnaire 80
16.3 Appendix 3 – Final Questionnaire 87

2 INTRODUCTION

Games today are often very complex, having many systems and mechanics that work together. It can be hard for new players to understand all these concepts. In order to make sure the player understands how to play the game, tutorials are required. Different developers try different approaches to teach players how to play their game, from games like Dark Souls [1], which provide only the most basic control information to the player via hints on the ground, to games like XCOM: Enemy Unknown [2], that follow a strict step by step procedure, guiding the player through every facet the game has to offer.

These tutorials are however not always met with praise and smiles, since players have a subjective opinion about how they want this information to be presented, and how much they want to be told. Some players, like ourselves, feel that tutorials can be very intrusive in a game, especially if they hold your hand too much, while other players really like it or have a need to get as much help from the game as they can.

We believe that one of the inherent problems that exist in tutorials currently is that there is typically only developed one for a game, and it is developed to ensure that people who have not played games before is able to begin with that game. However, we believe that a large part of a game's users, are players with previous knowledge of games and therefore do not need the same amount of information as a first-time user, in order to understand the same mechanic. We believe a way to avoid this negative feeling of a tutorial either obstructing you, or treating you like they are unintelligent, could be to devise a system which is able to determine when a player needs help in understanding a mechanic. This would allow for players who are able to grasp concepts faster to start the real game more quickly, while still assisting players who need more time and information. Each player would reach the start of the real game at different times, however it would be certain that they all have the same level of knowledge about the game and its mechanics.

Before diving into such an endeavor, we would like to make sure that this is a problem that actually exists, and not just lean on our own assumptions. Therefore, a short survey was created and sent out, which investigates the concept of tutorials in general within gaming. The results from this survey show that a substantial portion of the participants [Figure 1], have skipped information in a tutorial and later regretted it because the game later assumes the player knows this information and requires the player to utilize it.

To implement such a system on a full game would be a huge undertaking, and way out of the scope of what can be achieved in this project. We will therefore investigate a specific area within a genre to look at and find elements that could serve as a testbed to implement the system on.

2.1 MOTIVATION

The motivation for this project came from our own experience with tutorials, and observations we have made on other players. We have found that many players do not enjoy being taught how to move in a game, when they already know how to do so. Additionally, accidently skipping some information that only appear once, can severely limit a player's performance throughout the game.

In order to obtain an understanding of where problems currently exist within tutorials in commercial games, an initial survey was designed and distributed. The survey was distributed through a website where you can create giveaways for games, SteamGifts[1]. Here you can ask people to help you, in order to

[1] https://www.steamgifts.com/ is a website specifically devoted to handling giveaways. Anyone can open a giveaway offering their steam games as reward.

participate in a giveaway. By distributing through this site, we got answers from people who primarily consider themselves as gamers.

We discovered that the most impactful element of tutorials that the player's experience is that they regret skipping information.

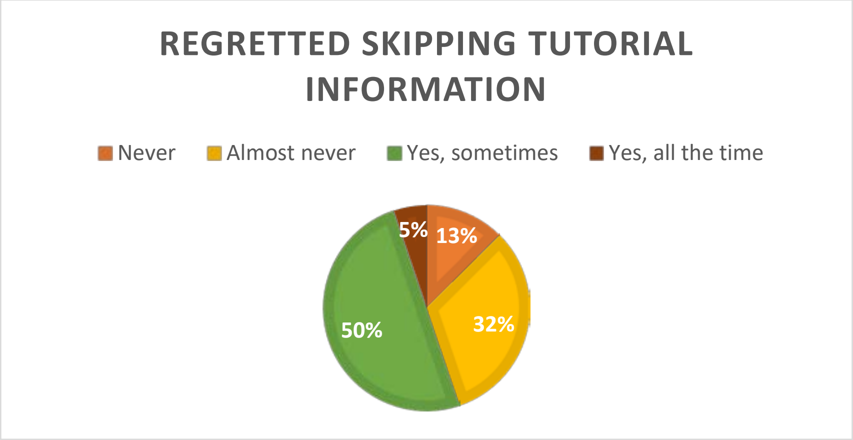


Figure 1 – Pie chart showing the answers of the question "Have you ever regretted skipping tutorial information, because it affected you later in the game?" for the initial survey. 96 participants answered the survey

We can see on the pie chart in Figure 1 that a large amount has had regret when skipping tutorial information, because it made it more difficult at a later point in the game.

This result tells us that there could be a need for an implementation of the idea described throughout this chapter. We believe that this on top of the other reasons mentioned, merits an investigation into how such a system can be made and how effective it is.

In order to properly test this project, and due to our own interests in the game development field, we decided that a game would be produced in conjunction with the investigation of a dynamic tutorial system. We want to have a full understanding of how different elements and mechanics are implemented within the game, in order for us to properly use them in a future test. Furthermore, by developing a game ourselves we will not be limited by any outside restriction a possible modding tool, or game API might have.

When choosing a genre to create the game, we looked at a collection of games within genres we had a personal interest in developing [16.1 Appendix 1 – Games We Played]. After playing each game and exploring the possibilities presented within the genres, we land on turn-based strategy as our chosen genre. This is both due to the games having generally problematic tutorial sections, how tutorializable the games are in the genre, and the possibilities we see for implementing a dynamic tutorial system for such a game.

2.2 Project Goal

The goal of this project is to create a dynamic tutorial system, which will allow players of different skill levels to go through a tutorial section in a game and receive different amounts of information, but achieve the same level of knowledge about the game in the end. The system should be able to determine what each individual player needs, and provide it.

The focus of the system will lie on complex mechanics, since these can be considered to be more taxing for a player to learn, and we believe that if a system focused on complex mechanics is proven to work, then it can be assumed that it will also work on other simpler mechanics. Whereas if a system based on simple mechanics is proven to work, it cannot be assumed that the system would work with complex mechanics due to the more complex nature of these mechanics.

By comparing a static version (control group) and a dynamic version (treatment group) of the tutorial within our game, we believe it will be possible to produce results that give us an indication of how well such a system can achieve the goal of the project.

2.3 Report Outline

The introduction and motivation sections have described our motivation for choosing this subject and stated the goal for the project. To fulfill the goal, we need to research tutorials, we then need to create a test product, and finally test the product. Each of these points are described below:

- **Research and Information**
    - What is a tutorial, and how to make a good tutorial.
    - State of the Art.
    - Adaptive methods.

We will start by investigating what industry experts, and academic papers have discussed about tutorials. We will then investigate if there are any current games that have attempted to tackle this problem. Lastly, we will investigate what adaptive systems are, how they function, and what an adaptive tutorial framework might look like.

- **Implementation**
    - The design of the game.
    - The dynamic tutorial system.

This section will discuss how the game was designed, and why it has been designed this way. After this, we will discuss how we believe an implementation of a dynamic tutorial system would work best.

- **Testing**
    - General tutorial design, using an iterative process.
    - Iterative testing.
    - Results, and their meaning.

The final part of the project is divided into different iterative test sections. Within these test section different areas will be covered, how the tests are designed and carried out, how we designed the tutorial, changes the iterative testing had on each test, and finally results, both their presentation, and their possible meaning and implications

3 WHAT IS A TUTORIAL?

After having defined the goal for this project, it is important to get an understanding of what a tutorial is. According to Dictionary.com one of its meanings is [3]:

> *"Programmed instruction provided to a user at a computer terminal, often concerning the use of a particular software package and built into that package."*

This definition is not specifically about games. To avoid having to make a distinction between games and software, our definition, and the definition that will be used when this word appears throughout this report is:

> *"A section of a videogame, which purpose is to specifically teach the player how to play the game and preparing the player for the remainder of the game."*

3.1 TUTORIAL TYPES

There are many different implementations of tutorials in games. This section will investigate many of these types, and look at advantages and disadvantages of each.

To be able to group tutorials, we investigated the subject [4], however, the results were found to be lacking. Therefore, we decided to play through a large number of tutorials, in order to create our own grouping of tutorials.

The investigation started before the genre for the test platform was chosen. This means that throughout the first part of the investigation, the games chosen were split among several genres we believed could be interesting to work on. After the genre had been chosen, the investigation changed focus to primarily look at games within our chosen genre; turn based strategy games. Apart from the genre of the games, the thought process of choosing games remained the same, we made sure to look at:

- Popular games within the genre
- Games, which might have an interesting tutorial
- Games, that had a good example of a specific type of tutorial
- Spot testing games in the genre, as it could reveal something.

Additionally, the games also had to meet the following conditions:

- It had to be accessible
    - Free
    - One of us already owned it
    - Borrowable
- The person playing it must not have tried it before

The entire list of games played can be found in [16.1 Appendix 1 – Games We Played]. Below is a short excerpt of games we tried, along with the reason they were chosen.

- **XCOM** – Very popular in the genre
- **Desktop Dungeons** – Has an interesting tutorial
- **Apotheon** – Good example of a specific tutorial type

Through this research, the most common tutorial types we encountered were these:

- Text boxes
- Highlights
- Action lock
- Exposition
- Dialogue
- Training room
- Contextual
- Player selected

All of these types will be examined in detail in the following sections. It is important to note that text boxes and highlights are often used as building blocks for other tutorial types, but they can also be used on their own.

3.1.1 Text Boxes

Text boxes, is as the name suggests, a box, which contains text. The way the text box is presented can vary wildly from implementation to implementation. It is a very common way to give the player information about anything from the simplest interactions to the most complex mechanics. Even though it is called a text box, many implementations also include images as part of the text box.

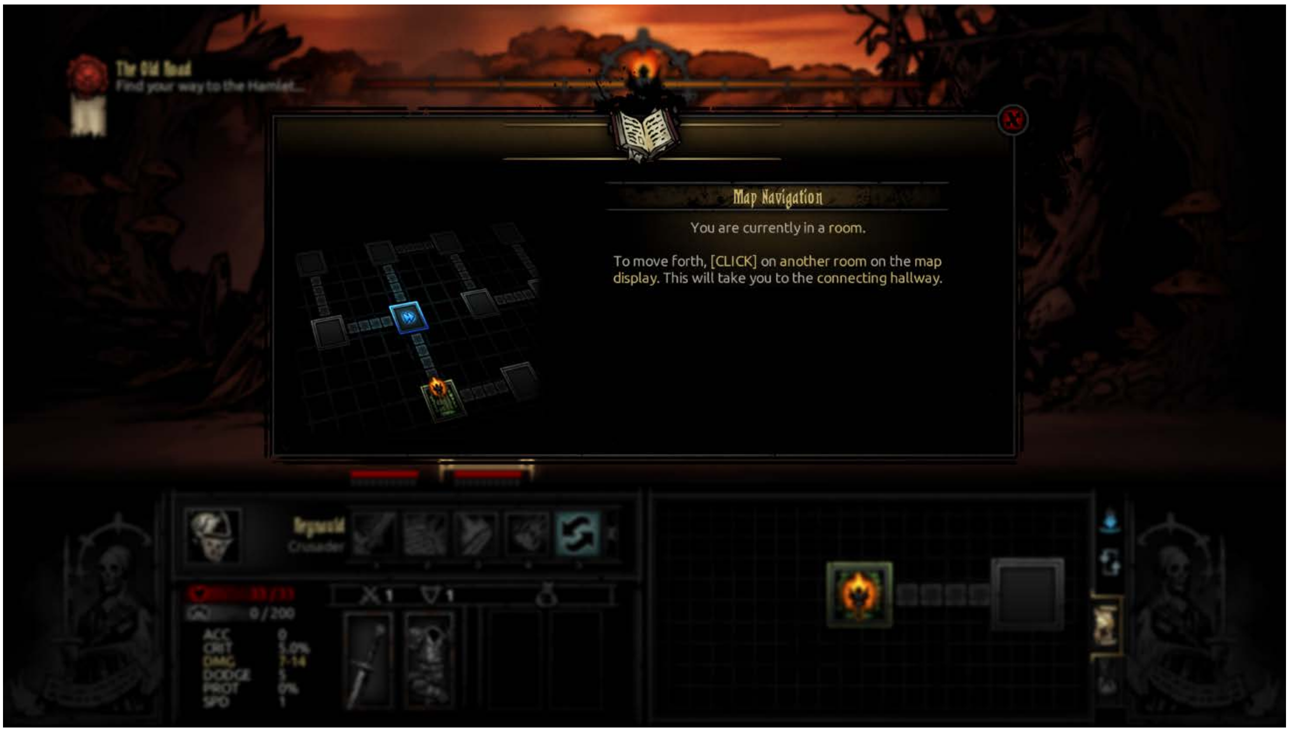
In Darkest Dungeon [5], text boxes are used to teach the player how to play. Whenever the player encounters something new, a text box will appear, informing the player how that mechanic works. Figure 2 shows an example of a text box in Darkest Dungeon, that contains a headline with a describing text underneath. On the left, there is an image, showing what the text refers to.

Pros:

- Good way of explaining a feature, without room for interpretation.

Cons:

- The player might skip it without reading it.



Map Navigation

You are currently in a room.

To move forth, [CLICK] on another room on the map display. This will take you to the connecting hallway.

Figure 2 - An example of a text box in Darkest Dungeon

3.1.2 Highlights

Highlighting, is a method where important things are emphasized, often by; fading the entire screen except for the important object, using a (pulsating) glow effect around the object, having an arrow point to the object, or a combination of these. This method is used to draw the player's attention to a specific object, which he should look at or interact with at this moment. This could be an end turn button, a button that opens the players inventory, or an indication of where the players attack value is located in the user interface.

Desktop Dungeons [6], a game that has two different tutorials will be explained in detail in [4.2 Desktop Dungeons]. The first of the two tutorials use only highlights. There is no text, no images, nothing but highlights guiding you through a dungeon. An example of such a highlight can be seen in Figure 3, where the blue particle effect, which outlines a tile, stands out from the otherwise black level, encouraging the player to press it.

Pros:

- Good way of drawing the player's attention.

Cons:

- Might not convey meaning very well.

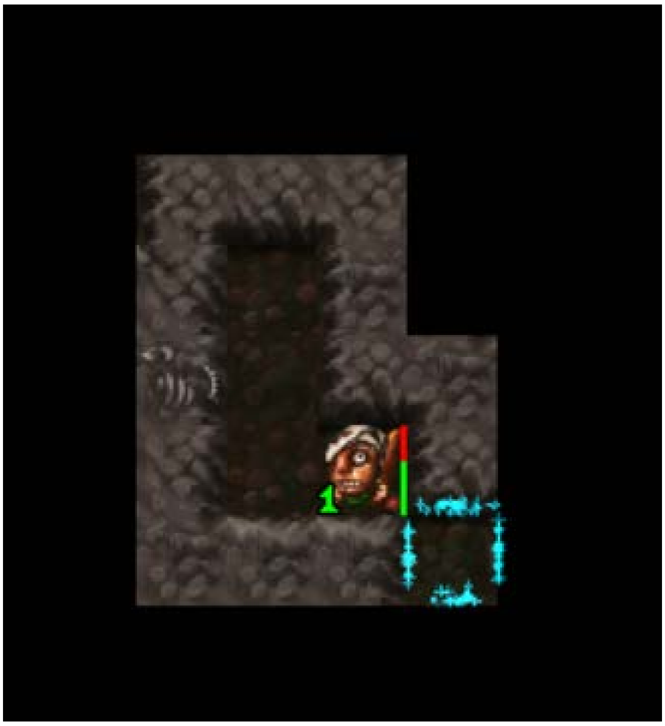


Figure 3 - A blue particle effect highlight in Desktop Dungeons

3.1.3 Action Lock

In an action locked tutorial, each time the game presents a new mechanic, the player is forced to perform a task in relation to that mechanic. The game will 'lock' and not progress until the task has been performed, and while the game is locked the player cannot perform any other actions.
Action locking is often used in conjunction with other elements, such as text boxes or highlights, to inform the player what action needs to be performed before they can move on.

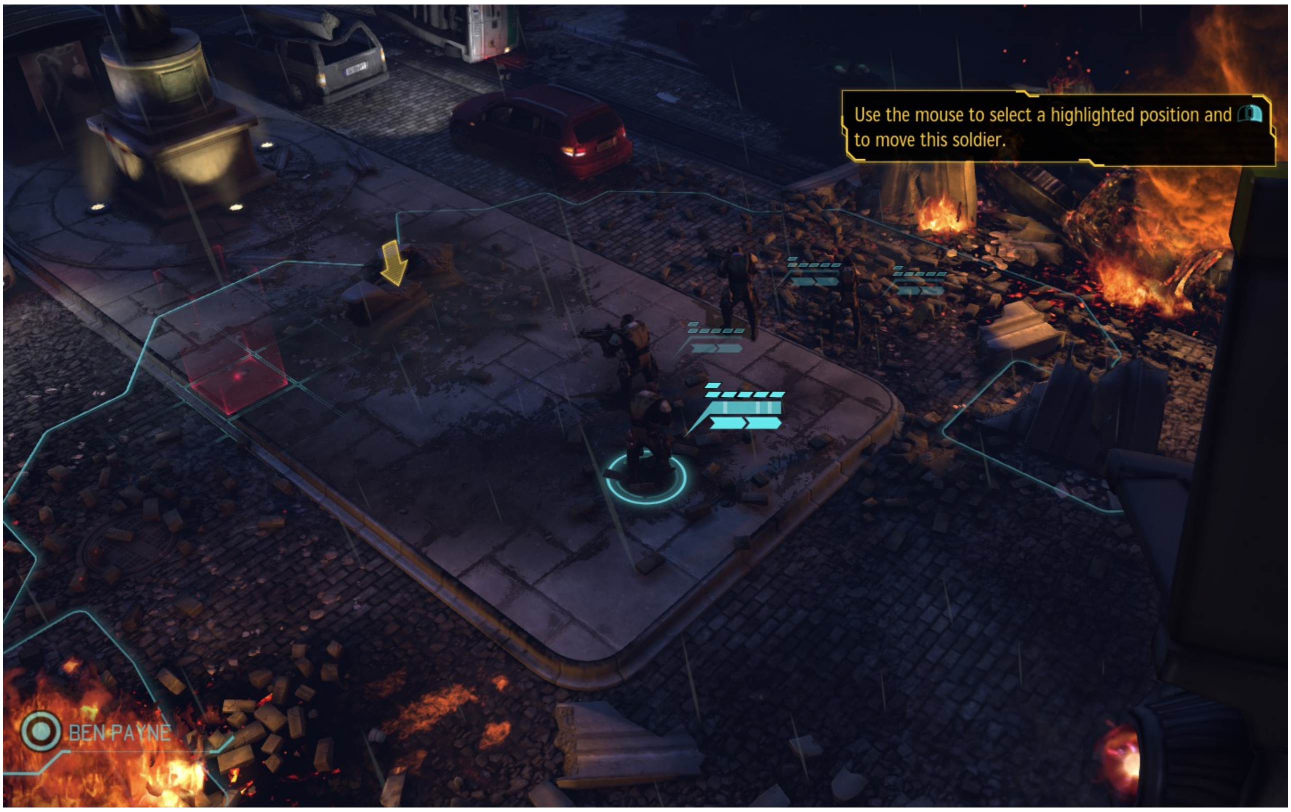
In XCOM: Enemy Unknown, the entire combat tutorial is action locked, the game gives the player one action to do at a time, the player is not able to do anything other than what the game wants him to. Figure 4, shows an example of action locking in XCOM: Enemy unknown, the player is only able to press the tile where the arrow is pointing, trying to press anywhere else, simply does nothing. Throughout the tutorial, more and more elements are introduced, one by one. Making sure that the player has tried to do everything.

Pros:

- Makes sure the player does what he is supposed to.
- Forces the player to pay attention to a specific gameplay element.

Cons:

- The player cannot make his own choices.
- Can be very abrasive to the play session.



Use the mouse to select a highlighted position and to move this soldier.

Figure 4 - Action locked tutorial in XCOM: Enemy Unknown.

3.1.4 Exposition

A tutorial form where the information is displayed in an image of the controller, where text and indicators show the mapping of the buttons.

The Last Guardian [7] uses this method continuously through the game, displaying an image at the top right corner with text describing an action, and the button, which executes this action is highlighted in yellow. These messages are not only shown as part of the tutorial, but are repeated throughout the entire game, whenever the player is in a situation where that action is possible, an example of this can be seen in Figure 5.

Pros:

- Makes sure the player always knows when he can perform a certain action.
- If the player has not played the game in a while, it is easy to get back into.

Cons:

- After having played a game for tens of hours, the player might be annoyed being told how to pull a cart, after having done so, many times already.

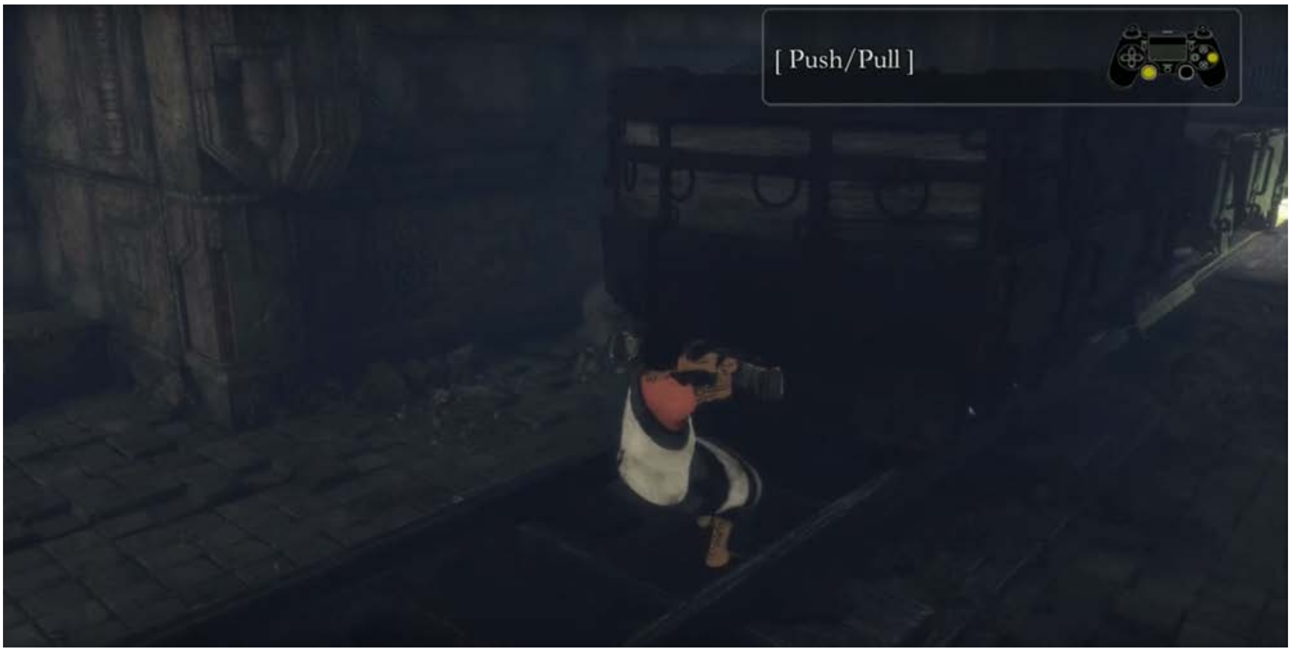


Figure 5 - Exposition of controls in Last Guardian

3.1.5 Dialogue Tutorial

A tutorial form where the information is given through conversation with non-playable characters (NPC) in the game, where the engaged NPC breaks the fourth wall and describes controller button mapping alongside story dialogue.

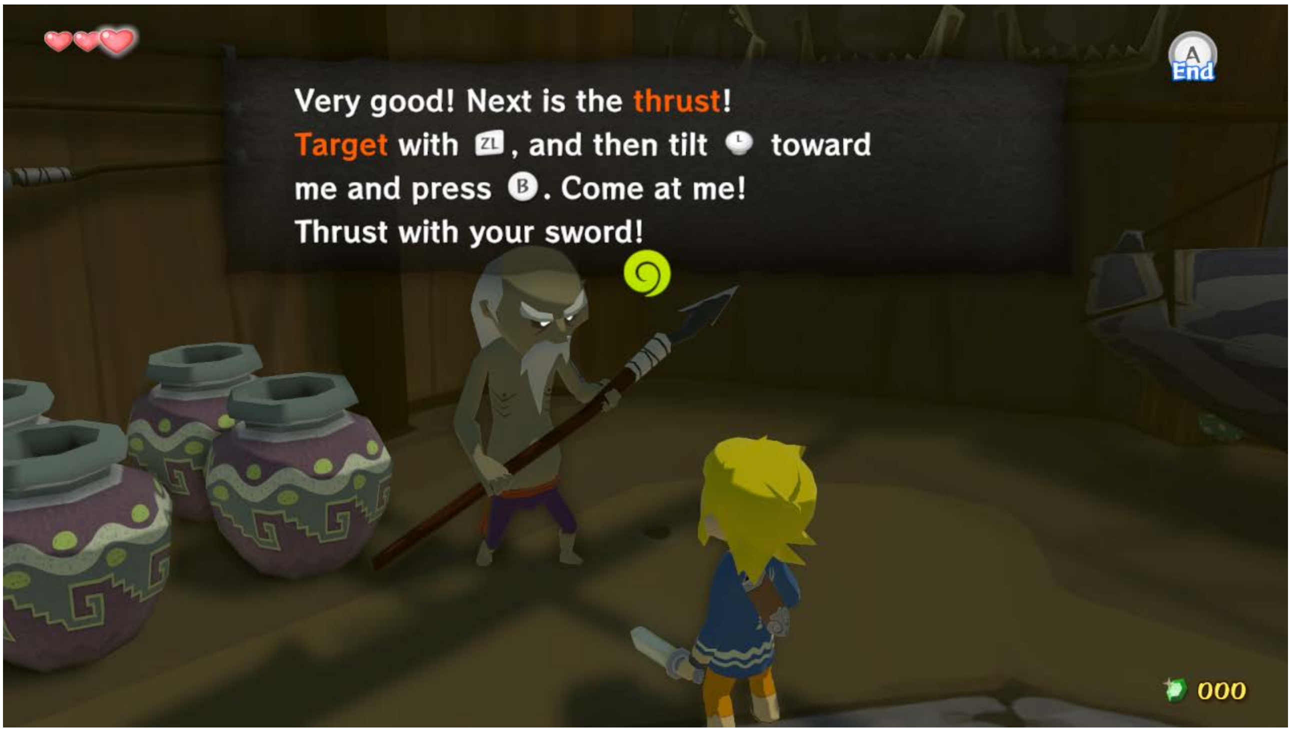
The Legend of Zelda: Wind Waker [8] uses this method as one of its tutorial elements. Where characters you interact with in the world, will talk to the player about doing certain tasks, such as learning how to fight with a sword. As they are talking about this, button icons will appear in the text as they are explaining how the player fights, as if the controller is part of the game world as well. An example of this can be seen in Figure 6.

Pros:

- Can make the player believe it is someone in the game that is explaining how to play, rather than the game itself.

Cons:

- Often breaks the fourth wall by having characters talk about the controls.



Very good! Next is the thrust!
Target with ZL, and then tilt ● toward
me and press Ⓑ. Come at me!
Thrust with your sword!

Figure 6 – Dialogue tutorial in The Legend of Zelda: Wind Waker

3.1.6 Training Room

A training room is usually a room or an area, which often exists outside the actual game world, where the player can become familiar with the controls and mechanics of the game.

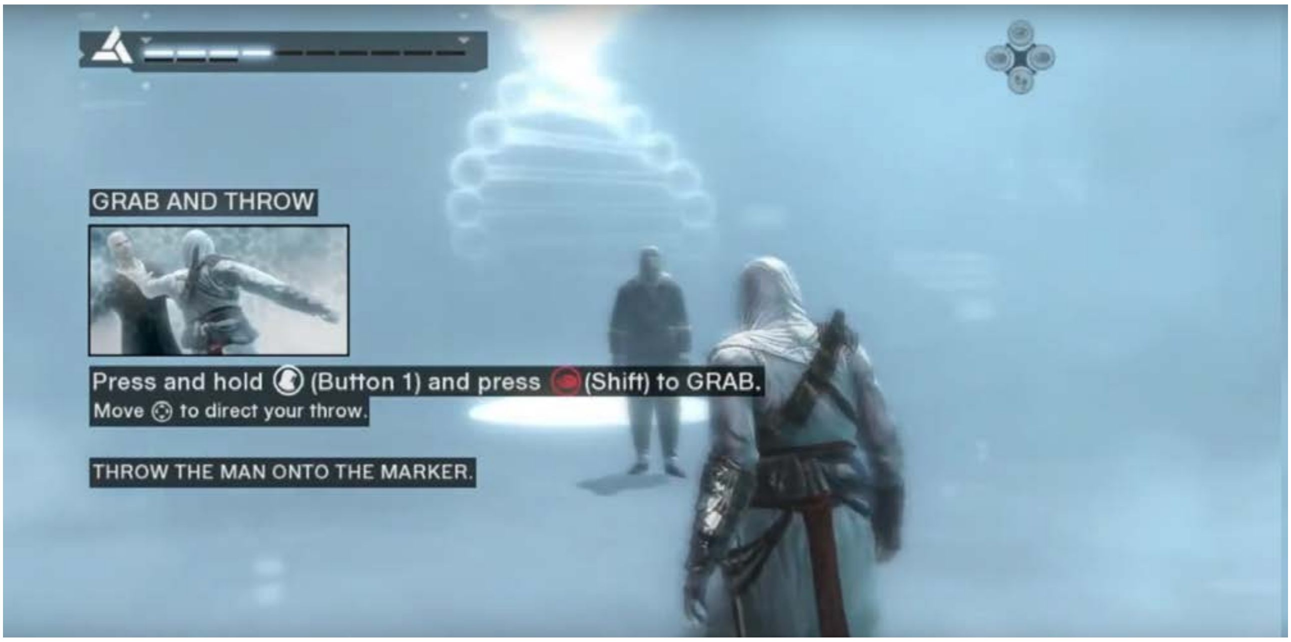
In Assassin’s Creed [9], the player is put into a dream world, where all the mechanics of the game is introduced one by one. Depending on what mechanic is currently being taught, the world will be populated with different non-playable characters or objects. For instance, Figure 7 shows an image of what the training room looks like while teaching the player how to grab and throw enemies.

Pros:

- Ability to teach the player without interfering with the story or setting of the game
- Gives the player the option to come back to practice certain skills without affecting progression in the game

Cons:

- Some players might skip it, because it is not a part of the actual game.



GRAB AND THROW

Press and hold (Button 1) and press (Shift) to GRAB.
Move to direct your throw.

THROW THE MAN ONTO THE MARKER.

Figure 7 - Training room in Assassin's Creed

3.1.7 Contextual

A tutorial form where the control scheme or mechanic information is woven into the game world, as if it was actual architecture in the physical space.

The tutorial in Apotheon [10] relies heavily on this method, where it utilizes its ancient Greek art style to implement the tutorial elements as murals in the level. This gives a visual representation of what can be expected to happen at a button prompt. Text might also be present, describing the action and the actual button. These are both represented in these murals to give further explanation. An example can be seen in Figure 8.

Pros:

- Ties into the game world.

Cons:

- Might not be obvious to all players.
- Players might misinterpret the information.
- Difficult to convey complex information.

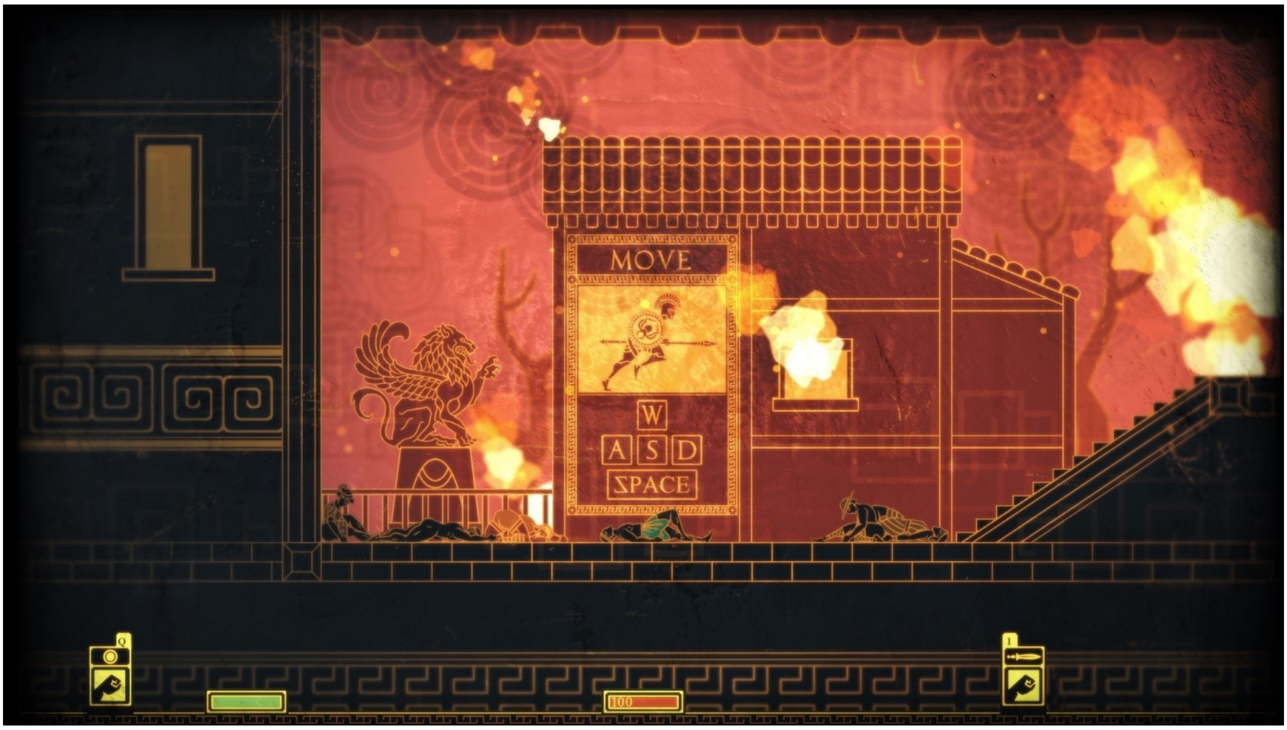


Figure 8 - Murals in Apotheon

3.1.8 Player Selected

A tutorial form where the tutorial is placed in interactable objects around the game world, which if interacted with display a message containing the tutorial information.

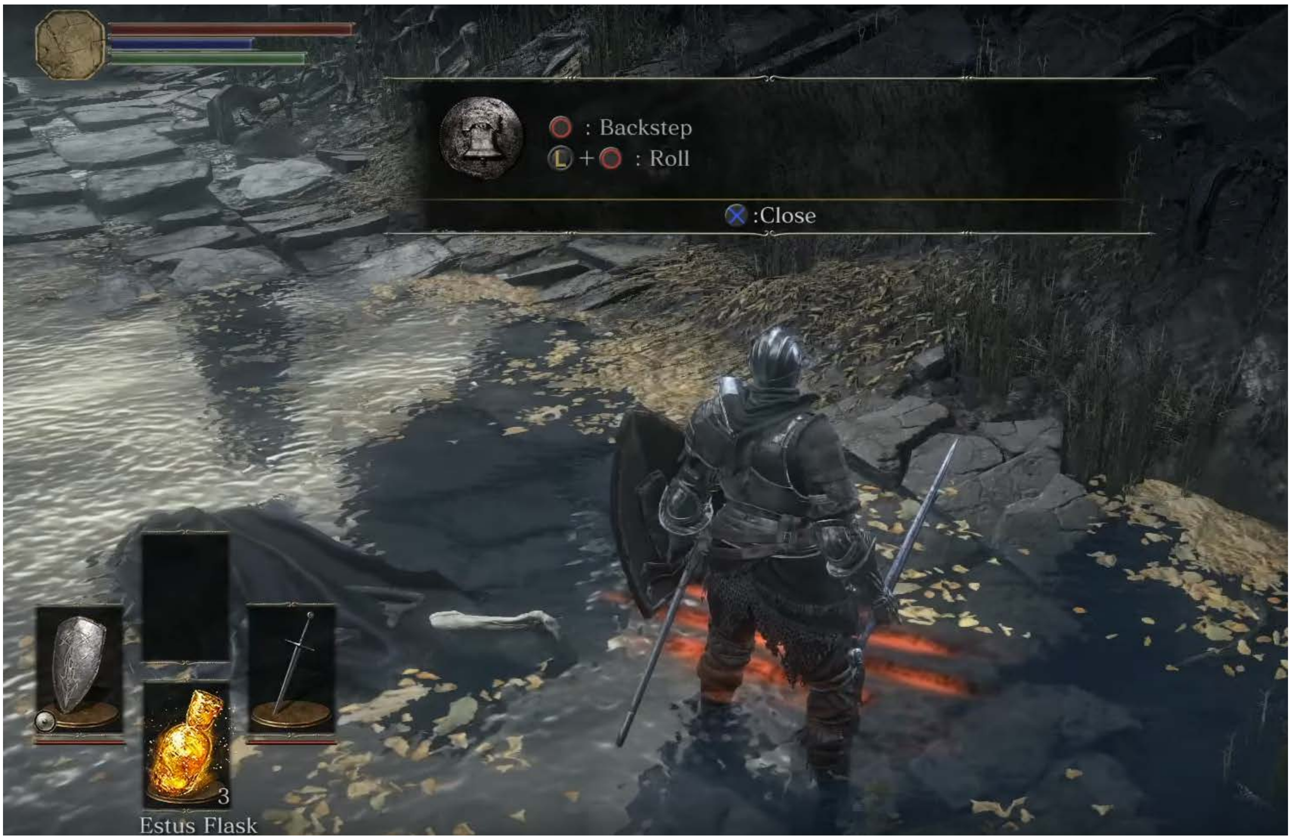
The Dark Souls series relies solely on this method for teaching the player how to play the game. The opening level is littered with glowing red marks on the ground, which contain the tutorial information. If an object is interacted with, a text box then pops up containing that information. An example can be seen in Figure 9. The objects follow the critical path of the tutorial level, offering control scheme information as it becomes necessary to know.

Pros:

- The player can choose when he wants to get more information

Cons:

- The player might miss an object containing vital information



○ : Backstep
L + ○ : Roll

Figure 9 – A message describing how to perform certain actions in Dark Souls

The tutorial types, which has just been examined were the most common in the games we investigated, it is important to note that we do not state that there are no other types, but these were the types we encountered the most. It is also worth noting note that the explanations are generalizations. Many games implement a mix of these tutorial types, utilizing different types for different parts of the game, to take advantage of the strengths of a given type, for that situation.

3.2 Creating a Good Tutorial

After looking at many different implementations of tutorials, the next step in getting an understanding of how to create our own tutorial is to investigate how people in the industry believe you make a good tutorial. However, it should be noted that they do not always agree, for example: Ernest Adams[2], writes in an article about tutorial design [11], that tutorials should always be skippable and never forced upon the player. Whereas Andy Moore[3] has said that after moving from a skippable to a not skippable tutorial, he went from losing more than 60% of his players after the first level, to having a retention rate of more than 95% [12].

Even though the professionals do not always agree on how you should approach tutorials, we still believe it is very relevant to look at what they believe is the correct way of creating it.

3.2.1 Ernest Adams

In an article on Gamasutra [11], Ernest Adams writes about eight ways not to do tutorials. This article flips the subject upside down, instead of looking at how to do it well; it looks at how you should not do it, because he believes it is hard to create rules for a medium as diverse as game development, and that different genres require different tutorials.

Do not:

1. **Force the player to take the tutorial**
   The tutorial should be optional, so experienced players do not have to waste time performing trivial tasks.
2. **Make the player read a lot**
   If the player has to read too much, they will forget a lot of the information. They will learn much more from doing rather than reading.
3. **Describe buttons and menu items badly**
   Whenever you reference a button, it should be very clear what you are referring to, there should never be any doubt.
4. **Leave steps out**
   When doing a step-by-step tutorial, you should never stop feeding the player instructions until the end of the tutorial.
5. **Punish the player's inexperience**
   If the player makes a simple mistake, he should not be punished e.g. by having to spend a long time climbing back up from a hole he fell in.
6. **Patronize or humiliate the player**
   The tutorial should have a positive tone, encouraging the player without being condescending.
7. **Force the player to complete the entire tutorial**
   The player should be able to exit the tutorial, if he figures out that he does not require further guidance.
8. **Don't give them a tutorial at all**
   No matter how well you believe your interface is, it will most likely need an explanation to someone who has never tried it before. Ernest Adams gives an example of him spending five

[2] Founder of International Game Developers Association [40].

[3] The developer behind SteamBirds [39].

minutes trying to simply open a door in a supposedly intuitive user interface, a task which should only take a few seconds.

As a concluding remark, he states that above all else, you should also test the tutorial on complete novices.

3.2.2 Extra Credits

Extra Credits[4], has made a video about how to create tutorials [13], in the video they outline five rules they believe you should follow when creating a tutorial.

1. **Less text**
   The tutorial should be interactive; the player will learn the mechanics better by trying them, compared to reading about them.
2. **No front loading**
   If you present the player with information about everything in the game right at the start, they will probably have forgotten chunks of that information before getting a chance to use it.
3. **Make it fun**
   The tutorial should be as engaging as the rest of the game. If the tutorial is not fun, many players might drop the game before getting through the tutorial.
4. **Reinforce learning through play**
   Throughout the game you should subtly highlight uses of the tools they have learned during the tutorial.
5. **Listen to your players**
   When working on a project for a long time, most mechanics seem obvious to you, however, they might not be obvious to someone who knows nothing about your game. Therefore, they suggest that you test the tutorial to figure out which elements are confusing to players.

3.2.3 George Fan

George Fan[5] gave a talk at GDC[6] back in 2012, giving 10 tips for creating tutorials [14].

1. **Blend the tutorial into the game**
   The tutorial should feel like it is part of the game.
2. **Better to have the player "do" than "read"**
   If a player performs an action there is a much larger chance that he will understand it, than from reading about it.
3. **Spread out the teaching of the game mechanics**
   The teaching of mechanics should be spread throughout the game, the further the player gets in the game, the more willing they are to learn new mechanics.
4. **Just get the player to do it once**
   When the player sees the result of their action, they will understand it.
5. **Use fewer words**
   You should try to use as few words as possible; he says you should try to have a maximum of eight words visible on the screen at a time.

[4] A team who create weekly YouTube videos about different aspects of game development. The writer of the show is James Portnow who among other things has worked as a consultant [46] on the Call of Duty [43] series and FarmVille [44].

[5] The creator of the game Plants vs. Zombies [17].

[6] Game developers conference [45].

6. **Use unobtrusive messaging if possible**
   When you need to display information, you should try to do it in a way that does not interrupt the player.
7. **Use adaptive messaging**
   Instead of telling the player everything, you should let them try to play the game, but if they are ‘doing the wrong thing’, you should give them a tip on how they should do it.
8. **Don’t create noise**
   You should only display crucial information; otherwise, the player might tune out the messages, and thus not miss important information.
9. **Use visuals to teach**
   The player should be able to look at an object and understand its function.
10. **Leverage what the player already knows**
    Use concepts people are already familiar with, thereby you do not have to introduce them. In Plants vs. Zombies, they used coins and diamonds as currency since players understand their value.

3.2.4 The Impact of Tutorials on Games of Varying Complexity

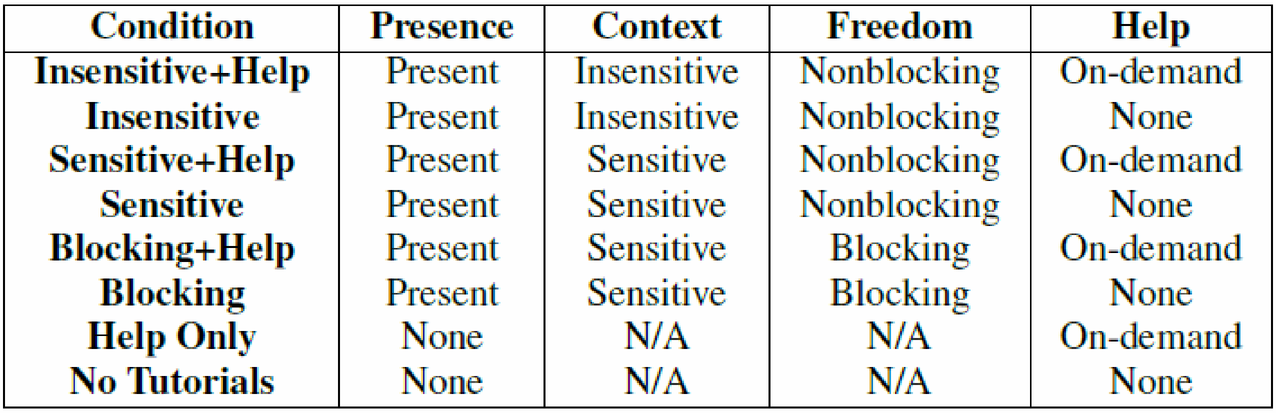
A paper by a group of researchers from the University of Washington [15], examines how tutorials affect player engagement (based on time played and levels completed) and retention in games. They used three different games, of varying complexity (two simple games and one complex game), to test eight different types of tutorials. These are composed of several different tutorial elements to try and isolate how each element affects the player.

- **Presence**
  Refers to whether a tutorial is present or not.
- **Context**
  If a tutorial is present, it can be either context sensitive or context insensitive.
    - **Context sensitive**
      Whenever the player reached a level which contains a new mechanic, it was explained at the start of the level. The context of the information being given was what the player needed to know.
    - **Context insensitive**
      For the context insensitive tutorials, an in-game manual was created, it would be displayed at the start of the first level, and contain information about all the mechanics of the game. The player could then flip through the pages to read about the different mechanics. This manual could not be reopened (unless help was enabled, which is explained below).
- **Freedom**

The context sensitive tutorials were divided into a ‘blocking’ and a ‘nonblocking’ tutorial. The blocking tutorial, would force the player to perform a specific set of actions relating to the mechanic being taught. This was done in a similar way to what has been explained in [3.1.3 Action Lock]. Whereas, for the nonblocking tutorial, the player would be free to apply the concept he had learned, for himself.

- **Help**
  For tutorials where help was enabled, a small help button was included in the UI, when pressed the in-game manual, created for the context insensitive tutorial, would be shown.

These four elements were used to create the eight unique test conditions presented in Figure 10.



Condition	Presence	Context	Freedom	Help
Insensitive+Help	Present	Insensitive	Nonblocking	On-demand
Insensitive	Present	Insensitive	Nonblocking	None
Sensitive+Help	Present	Sensitive	Nonblocking	On-demand
Sensitive	Present	Sensitive	Nonblocking	None
Blocking+Help	Present	Sensitive	Blocking	On-demand
Blocking	Present	Sensitive	Blocking	None
Help Only	None	N/A	N/A	On-demand
No Tutorials	None	N/A	N/A	None

Figure 10 - The eight tutorials [15]

They wanted to see how each tutorial element influenced the three different player metrics: time played, amount of levels completed, and return rate. Their conclusion was split into two parts, one for the two simple games, and the other for the complex game:

- **The simple games**
  They found no significant improvement to player engagement when a tutorial was present in those games, for some of their test conditions they even found a decline in player engagement when a tutorial was present compared to no tutorial at all.
  They concluded that it might not be worth the investment in creating a tutorial for games where the player can easily discover all the mechanics through experimentations.
- **The complex game**
  For the complex game the results were different, there were a significant increase in player engagement when a tutorial was present compared with no tutorial. The tutorial with the highest median play time and levels completed were the context sensitive tutorial. They concluded that for complex games it seems best to present the player with information shortly before it is required. They also found that providing the player with information before it was required, or not giving the player information at all, resulted in lower play time and less levels completed.

3.2.5 Takeaways

From the analysis of how to create good tutorials throughout this chapter, we found these three points to be recurring subjects mentioned in several of the sources.

- **Use as little text as possible.**
  Ernest Adams, Extra Credits, and George Fan, all suggested that the player should read as little text as possible, and what they had to read should be relevant.
- **Do not front load.**
  Extra Credits, George Fan, and the authors of [15], agree that mechanics should be explained shortly before the mechanic must be used for the first time.
- **Playtest the tutorial and adapt accordingly.**
  Both Ernest Adams and Extra Credits say that the tutorial should be tested, as it is the only way to figure out if it fulfils its purpose of preparing the player for the rest of the game.

Everything the experts has said should be considered when creating the tutorial for this project.

4 STATE OF THE ART

This chapter will investigate what steps others in the industry has taken to create adaptive tutorials, in an effort to see if there is anything we should be aware of when designing our system.

4.1 MINECRAFT

Minecraft [16], a very popular sandbox game, officially released in November of 2011, since then, the game has not had an introduction on how to play. This means that if you are completely new to the game, it can be very hard to get into. However, for the upcoming 1.12 update[7], a tutorial will be implemented. The tutorial is meant to teach new players the basics of the game, moving, breaking blocks, and crafting. The tutorial will only give the player information about the mechanics he has not yet done, e.g. if you do not move within the first few seconds of the game, it will display a small textbox in the top right-hand corner explaining how to move, as can be seen in Figure 11.

It is interesting to see such a popular game, that has been out for many years without any tutorial, try to implement a dynamic tutorial feature, as it shows that the industry might be developing an interest for adaptive tutorial systems. The implementation in Minecraft focuses on only the simplest mechanics in the game, and the result of this implementation could help shape a full system that looks at all complexity levels.

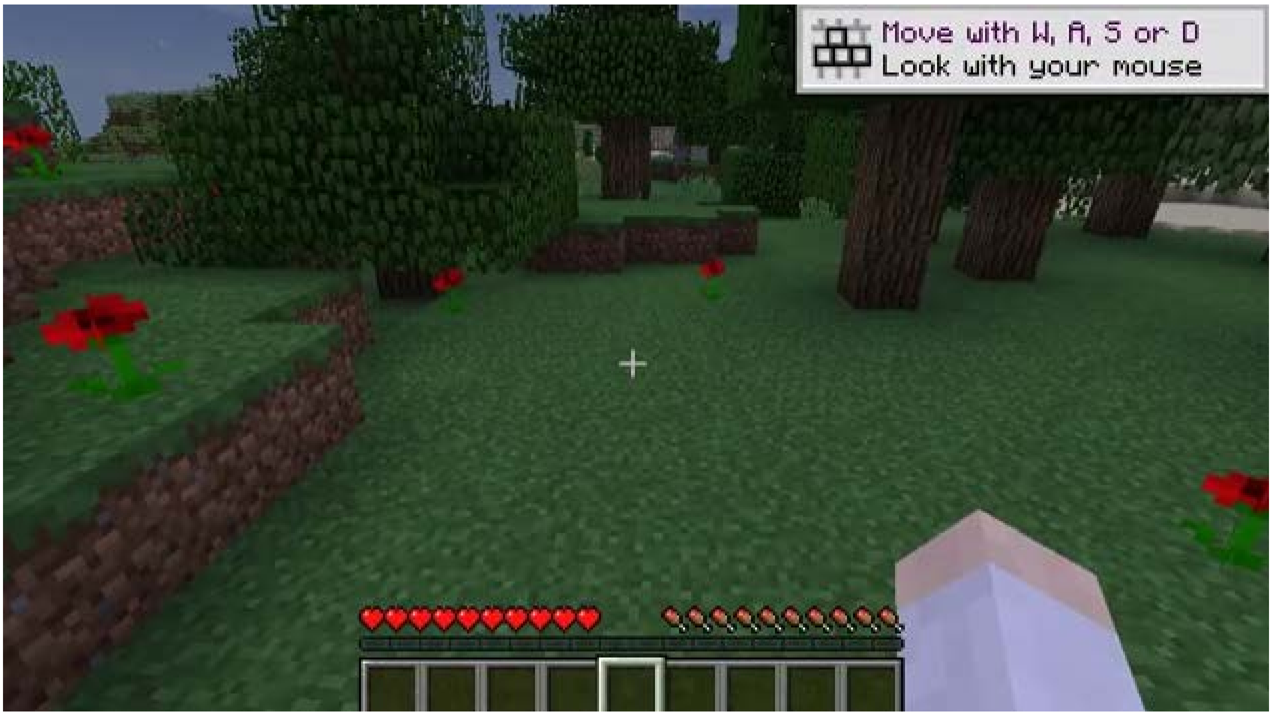


Figure 11 – Minecraft's new tutorial system

[7] 1.12 is set to release 2. June 2017 [41]. The first teaser for the tutorial system came in a tweet 19. April 2017 [42]

4.2 DESKTOP DUNGEONS

The tutorial in Desktop Dungeons is consists of two different tutorials; here they will be referred to as the 'highlight tutorial' and the 'basic tutorial'.

When you first launch the game, you will start in the middle of a dungeon, inside the highlight tutorial. There is no explanation of what to do, and no text giving you any information. Figure 12 is a screenshot of what the player sees before doing anything, the two tiles directly above and below player are highlighted using a blue particle system. This indicates to the player, that these specific tiles are of interest in some way, and attempts to guide them through using these simple markings. If the player dies during the highlight tutorial, he is given the possibility either to try again, or to try the basic tutorial instead. The basic tutorial uses a mixture of a [3.1.1 Text Boxes], [3.1.5 Dialogue Tutorial], and [3.1.8 Player Selected] to teach the player about the game, this tutorial is very in depth and does not leave anything out.

It is interesting to look at this tutorial because it is possible to go through the first tutorial rather quickly and then start the game, without having felt like being through a tutorial. If you do die during that tutorial, the game offers you more help. There are a few flaws with this design; you are able to complete the first tutorial while not fully grasping some mechanics, this might penalize you throughout game. Furthermore, if you do die in the first tutorial, and decide you would like to try the basic tutorial, no matter how far you got, you have to go through the entire basic tutorial, if there were just one mechanic you were unsure about, you also have to go through everything else.

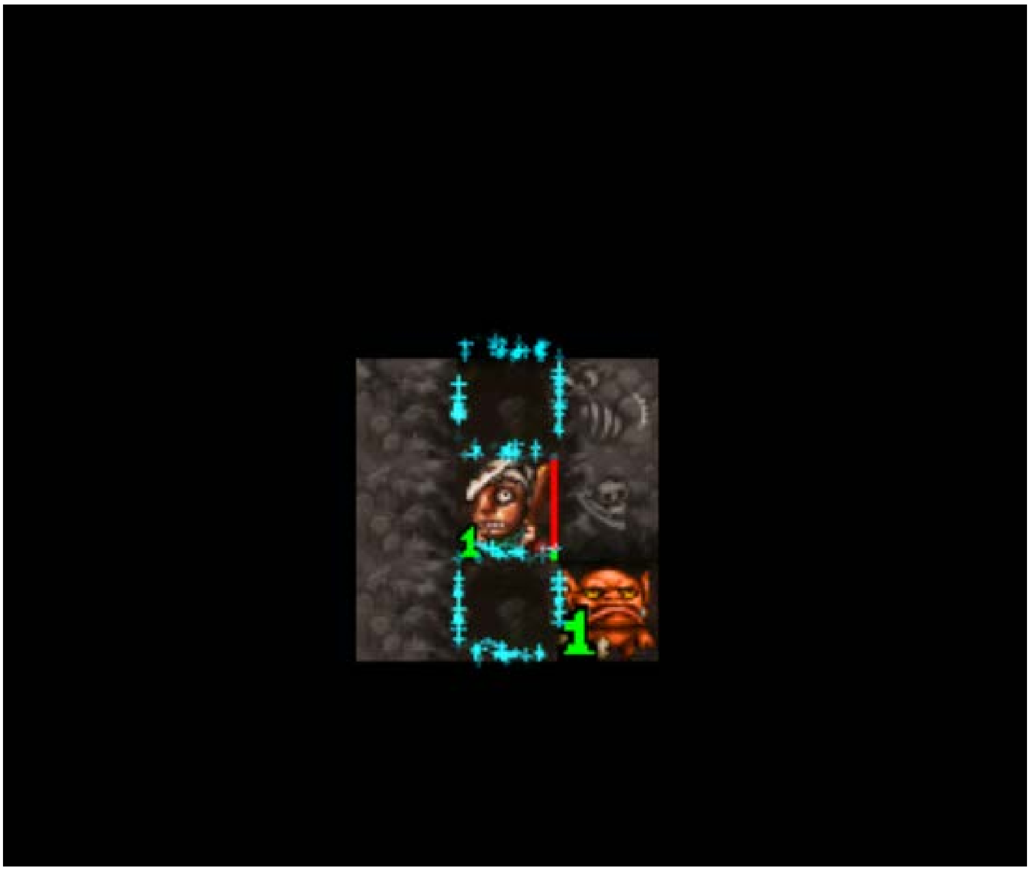


Figure 12 – What you see when you start playing the game Desktop Dungeons.

4.3 Plants vs. Zombies

In Plants vs. Zombies [17], you are defending a house from an invading horde of zombies, to defend the house; you plant flowers on the lawn between the house and the zombies. The plants are stationary, and most will shoot in a straight line towards the zombies. Figure 13 shows what the game might look like in the middle of a level.

George Fan said as tip #7 in [3.2.3 George Fan], that for Plants vs. Zombies they had great success in implementing an adaptive messaging system. This system displays a hint to the player if he is “doing the wrong thing”, one of these “wrong things” is planting the flowers on the right side of the screen, as they would not have as much time to shoot the zombies before the they reached the flowers. In this case, a tip would be displayed stating that it would be a good idea to plant the flowers on the left side instead. This tip would never be displayed to players who instinctively understood how to play optimally.

This implementation showcases a different approach to the dynamic teaching system, where it is used to give hints throughout the game, rather than focus on initial learning. This possibility is also interesting to consider, since it uses a non-forceful approach, and simply chooses to give hints to the player of how the game could be played.

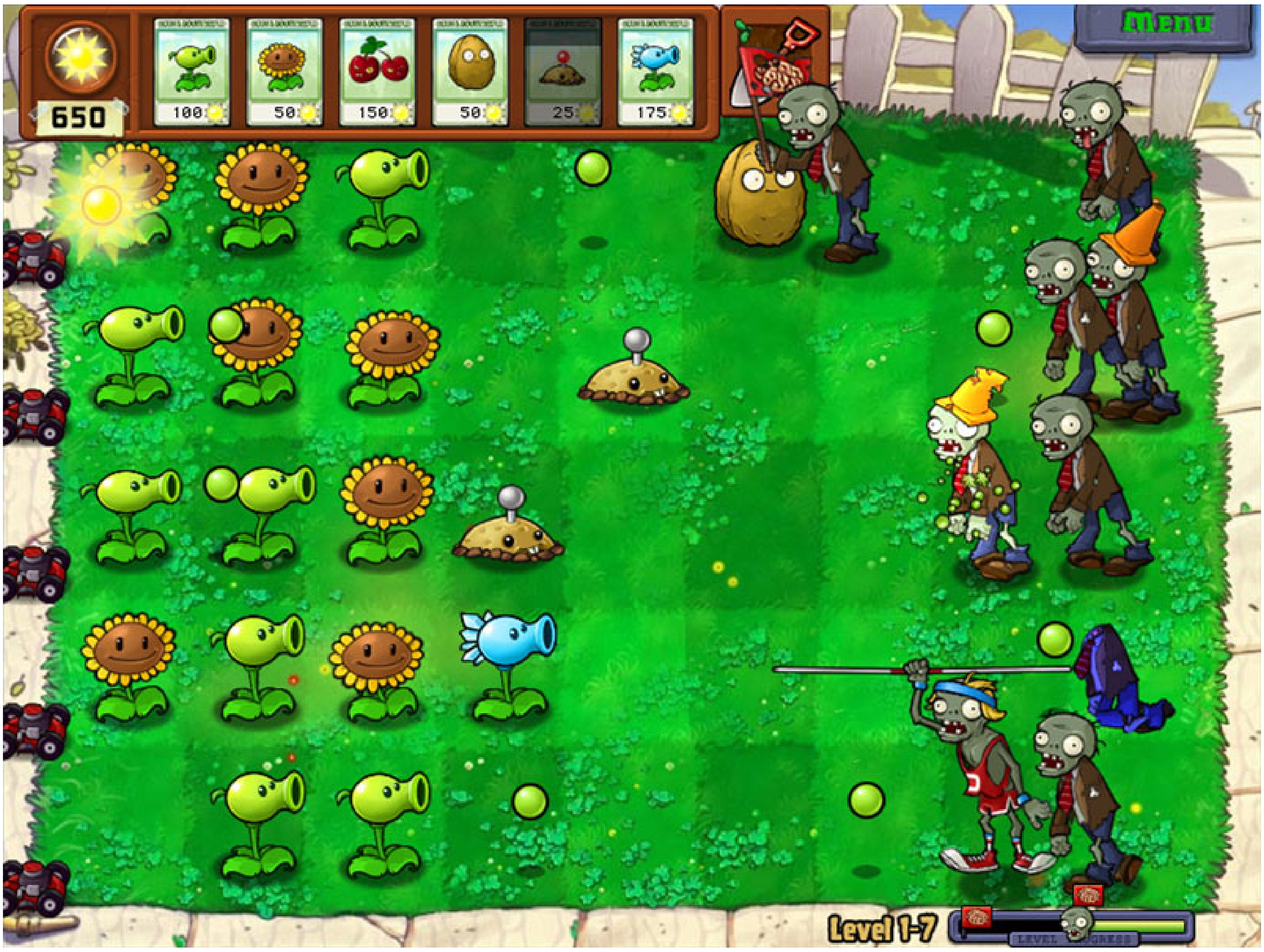


Figure 13 - Plants vs. Zombies battlefield [17]

Through research, these were the only examples of games, that tries to adapt their tutorial to the player, we encountered.
The tutorial currently being developed for Minecraft, attempts to teach new players the simplest mechanics, without interrupting experienced players.
Desktop Dungeons allow skilled players a quicker way through the tutorial. While providing additional help for players who need it.
Plants vs. Zombies, allows the player to jump straight into the action, but will attempt to help the player if he is doing something subpar.

Based on the limited number of existing implementations, and their apparent limited scope, it seems that not a lot of work has been done in the field of adaptive tutorials so far, but it does seem that there is a huge potential.

5 Adaptive Methods

Adaptiveness in games is an area which strives towards tailoring a game experience to each individual player. The focus of adaptive research is currently around adapting specific elements within a game to optimize player experience [18]. These elements are most commonly the player avatar, non-playable characters (e.g. dialogue changes, or AI difficulty scaling), the game environment or states of the game (e.g. changing the narrative, or changing how a player can interact with the world). In order to adapt to a player, one must understand the player, this is most commonly done by modelling the player based on different sets of data. A system then analyzes this data and categorizes each player into different categories which then should trigger a change in the game itself and make it more enjoyable for players in the different categories [18]. An example of an adaptive system framework can be seen in Figure 14.

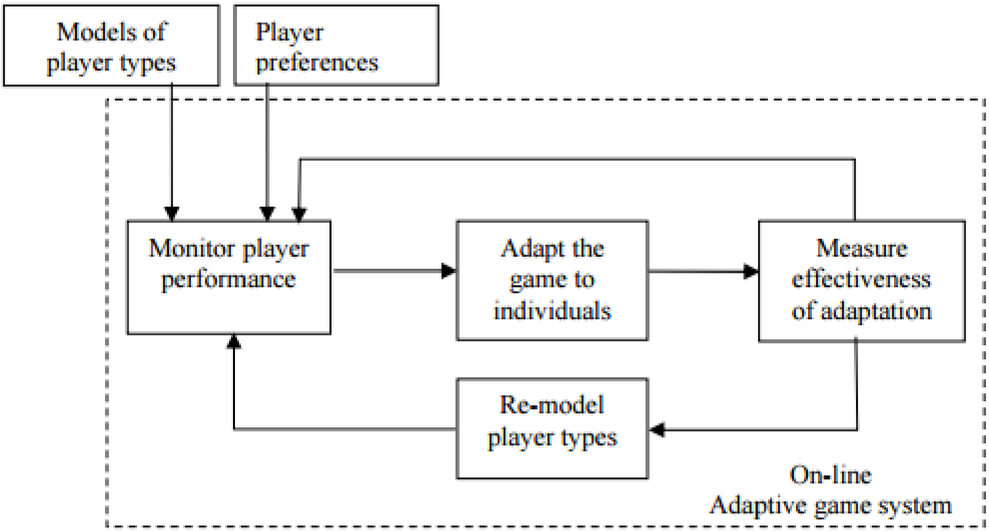

Models of player types
Player preferences
Monitor player performance
Adapt the game to individuals
Measure effectiveness of adaptation
Re-model player types
On-line Adaptive game system

Figure 14 - A proposed framework for adaptive system [18]

The framework presented in this figure is imagined to be used for alternating game elements throughout an entire game in order to grant a player a better game experience. During the game, the system will continuously update and check its adaptation, to make sure that it is still providing the best game experience for the player as possible.

The figure proposes to use player modelling in order to obtain an understanding of which type of player, is playing the game, and uses this to attempt to adapt content that fits the player's model.

5.1 Player Experience Modelling (Data gathering)

The player experience modelling (PEM) field contains three different types of PEM's, Subjective PEM, Objective PEM, and Gameplay-based PEM [19]. These three terms cover different areas of modelling a player's experience utilizing different methods. PEM's can be used in adaptive systems to understand how the adaptive system should adapt to the player.

The subjective PEM uses direct contact to the player either while he is playing the game, or directly after being done playing the game. This includes asking the player questions, or giving the player surveys (both while playing and after). This information could then be used by a system to figure out and adapt to each individual player, optimizing their experience. These methods can be considered intrusive, as the investigation itself would disconnect the player from the game, forcing him to interact with elements outside of the game.

The objective PEM uses physiological measurements. These include observing heart rate, monitoring eye movement, analyzing facial expressions, etc.
By using this form of player modelling, a system would be able to make assumptions about the affective state of the player by not interrupting the player's play session.
Multiple issues with this method have however been discussed, e.g. heart rate monitoring (and other physiological measurements) usually requires the player to wear some sort of physiological sensor which could be considered intrusive if you are simply just trying to play a game. Players might be less inclined to play a game and return to it if it is necessary for them to attach physiologic sensors to their body. An additional problem is of a computational nature, specifically if you are trying to use facial expression, as analyzing this in real-time requires a lot of computational power, and would not be possible to do on a larger scale on all players' personal computers.

The gameplay-based PEM uses data gathered while the player is playing the game. This data usually consists of player behavior within a game, both the actions of the player e.g. how he is moving, how he handles dialogue, or how he handles combat. This is considered to be the least intrusive PEM method, however the correlations between the data and the players affective state are not as strong as the other methods.

A combination of these could also be used to form a hybrid PEM method, to allow for more data to correlate between it and to possibly reduce necessity of assumptions inherent in the different PEM's.

These concepts discuss the complexity of understanding a player's affective state in a video game environment. This project revolves around understanding the player's knowledge of the game, during the game and is therefore less complex. We believe that understanding whether a player understands a specific mechanic can be determined without analyzing their affective state. This is imagined to be possible by analyzing different in-game variables about how the player is utilizing the knowledge he currently has about the game.

As the system is not investigating the affective state, nor should it adapt in an offline method, the best method for gathering data is through gameplay-based PEM.

Gathering data is not enough for a system to work properly. The data must also be analyzed and processed. We believe this can be done by using a fitness function also called an evaluation function.

5.2 FITNESS FUNCTION (EVALUATION FUNCTION)

A fitness function is a mathematical algorithm which based on incoming data specifies how well the e.g. generated content performs, based on the designers' goals. A fitness function can be devised focusing on multiple different aspects, e.g. fun, anxiety, frustration, accomplishment, etc. Generally, three different types of fitness functions exist [20] [21]:

- **Direct fitness function**
  The fitness function receives data directly from generated content. This can be data such as height differences in a map, scattering of resources, or maximum reachable speed.
- **Simulation-based fitness function**
  This type is based on an AI playing through content. This AI then evaluates the produced content, based on different points of interests, such as: playability, time to learn the content, or variety in gameplay.
- **Interactive fitness function**
  The interactive function obtains data directly from the player while the player is playing the game. It can evaluate different areas, such as: time spent at a specific point in the level, amount of times a player died, or explicit words provided by the player.

5.2.1 Examples of How Fitness Functions Can be Used

In the article written by Ponsen, Muñoz-Avila, Spronck, Aha [22], they present their work on an adaptive AI which analyzes the states of the game, and the possible options that exists for both the player and the enemy. The paper talks about how to optimize the AI further, using an evolutionary algorithm and having it develop actions for itself. This is done in an RTS game. They used a fitness function, which determines which action is best suited for the current state of the game, by incorporating both the current state of itself and the state of the player. It then weighs the different obtained scores, after which the system picks the one that has the best relative score. The states of the games were defined by what buildings the AI had and which the player had. It chooses the highest relative scored action, by selecting what is best from a knowledge base describing different tactics. This selection process is based on both the military score (current combat units) of the actions and the building score (currently owned buildings) of the action.

Togelius, Nardi, Lucas have presented a method for procedurally generating race tracks for a racing game [23]. They tried to optimize the generated track to provide the most fun experience for a player. They used three different fitness values in order to select the best generated tracks, these are: target final fitness, maximum speed achieved, and maximum fitness variance. These were chosen to allow for the tracks to keep within the design goals. They wanted to try and push the content to contain straight areas which allowed for high speeds, and to allow for tracks to generally differ from each other to obtain more variance in the generated content. They believe that these conditions are a part of making entertaining racing tracks.

The game Galactic Arms Race [24] is a 2.5D action space shooter. This game uses procedural content generation for their weapons. The goal is to generate new weapons the player will find attractive to use. The way this generation is controlled, and evaluated is using fitness which is based on what weapons the player is using. Every time a player fires a weapon, the fitness score of that weapon is increased, while the other available weapons the player is not using is decreased accordingly. This allows for the system to provide the player with new weapons which are evolved from weapons he is already familiar with, and enjoys using, they therefore have a high likelihood of being used by the player [25].

5.3 ADAPTIVE TUTORIAL

The interest in using an adaptive tutorial is somewhat different compared to the goal of other adaptive systems within games. The difference is that the focus should not be to optimize the gameplay experience by adapting game content to each individual player. We believe that an adaptive tutorial should instead attempt to help players of all skill levels to have the same amount of understanding of the game and its mechanics, when the tutorial is over.

The data collection for many adaptive systems revolves around being able to categorize players into different types, which then allows for the system to adapt the game's AI/mechanics to be more suited for that player. A tutorial system would focus more around teaching, and the player learning rather than the AI/mechanics adaption. Therefore, the need to categorize and model players based on data can be simplified. As we believe it is not necessary to categorize players into different categories, but have a system which can recognize behavior which could indicate that the player does not understand a specific mechanic. Based on this we believe an adaptive tutorial system could therefore look like the framework presented in Figure 15.

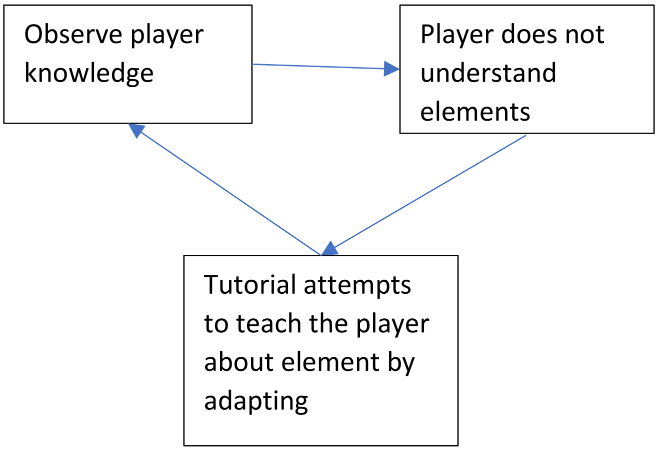

Observe player knowledge
Player does not understand elements
Tutorial attempts to teach the player about element by adapting

Figure 15 - Our proposed framework for an adaptive tutorial

This system attempts to understand the player's knowledge through data collection from within the game. The system, based on this data, decides how much/how little the player understands and adapts the tutorial in order to best support this player in the goal of learning how to play the game. This theorized system is very similar to the framework of the system presented in Figure 14. The main difference between the two is, that the need for modelling the player is removed. As the system in Figure 15 does not attempt to provide the best game experience but attempts to teach game mechanics.

6 GAME DESIGN

This section will investigate the turn-based strategy genre and describe the design of the game that will be developed as a test platform for the project. The section will focus on the different elements which are normally found in games within the genre, and how these elements will be implemented in our game. This will help us craft the tutorial that should accompany this game, and to find the specific test focus points. We investigate this subject both from literary sources and from our own investigation into the genre from the collection of games we played in the early stages of the project [16.1 Appendix 1 – Games We Played].

6.1 TURN-BASED STRATEGY

In order to properly make a turn-based strategy game, we first look at what basic and complex elements characterize this genre. To do that we look at popular examples and analyze their mechanics and find the key elements we need to complete the design.

6.1.1 Turn-Based Competition

Turn-based strategy in its base is about the two competing sides taking turns to execute their actions, although there are many different ways to split these sequences with very different effects on gameplay. The main types are [26]:

- **Classic Turn Based (I Go, You Go!)** - The most traditional system, where one player is in control at a time and makes all their available moves, after which they turn control over to the other player who then makes all their available moves, this process is repeated throughout the game.
- **Timed Turn-Based –** Similar to the classic system, but with a time constraint on each turn, meaning that a player only has a limited time period to make all their moves. If they do not make it within the allotted time, the remainder of their moves are forfeit.
- **Turn-Based with Reaction –** Also similar to the classic system, however the inactive player is allowed to react to the actions of the active player. An example of a game using this is Magic: Duels [27], where the inactive player is able to counter attacks from the active player using spells or choosing a unit to defend with.
- **Phase-Based –** This system has both players simultaneously plan out their actions for an upcoming round, and it depends on various factors, such as dice rolls or passive modifiers in the game, determine in what order the actions are executed.
- **Activation-Based –** This system sort of resembles the classic system, however each player has to perform actions with a different unit each turn until all units have made a move. When all units have been "activated" the restriction resets, and the process repeats.

Additionally, three types are mentioned in the article, yet these have too minor variations from already listed systems to warrant and inclusion to the list.

6.1.2 General Game Mechanics

Apart from the actual game rules of how the turns function, many games utilize different, similar mechanics. To be able to create a turn based game we decided to look at some of these different mechanics to obtain an understanding of how they function within games, and what they provide to a game. These points were discovered through our initial investigation into the genre when playing through the games listed in [16.1 Appendix 1 – Games We Played].

6.1.2.1 Unit Management

Although not necessarily present in all turn-based strategy games, one key element found in many games within the genre is a unit management system. This mechanic is seen used in a great deal of turn-based strategy games, such as Heroes of Might and Magic [28], Civilization [29], and even in Chess [30]. It can be considered one of the staples that define the genre and is utilized in many different ways from game to game, however the core of the system revolves around the player having an inventory of useable units, which usually has some kind of a restriction, such as limited space or the player not having enough resources to either acquire or use his units. The player is then tasked with managing this roster properly and utilize the right units for the right situations.

6.1.2.2 Type Relationships

Another system, which can be considered a core element in the genre is a type relationship between the units. This means that each, or some, of the units have a specific type assigned to them, and different types have different effects on each other. The most common examples are a strength and weakness relationship, such as seen in the Pokémon games [31], where e.g. a water type unit is strong against a fire type unit and therefore deals double the damage to it. Another example is Stratego [32], where a specific unit can only be beaten by different prespecified units.

6.1.2.3 Special Abilities

This system is also seen in many turn-based strategy games, but has been implemented in many different ways. In its essence, the system is used to widen the strategic possibilities, by adding unique sub mechanics to the game that go out of the conventional abilities that most, if not all, other units have. In Heroes of Might and Magic, it is implemented as a spell system, where the player can learn and cast spells that are independent of the units they have and are considered separate actions. In Pokémon Duel [33], a unit can have a set of abilities that can make it special. An example of one such ability is e.g. being able to jump over other units, or displace its position to a different place on the board.

6.1.2.4 Field of Combat

A very important aspect of the genre, is the field of combat, the place where the battles take place. Most turn-based strategy games use a tiled or sectioned combat map, where units are able to stand in specific spots and move in certain patterns around these spots. A clear example is again Chess where each unit has a specific pattern in which they can move around an 8x8 square tiled board, or Heroes of Might and Magic where the units have a set number of move points and can move freely in a radius that covers this between hexagonal tiles on the map. This is where a great deal of the strategy comes into the game, as the player needs to constantly think about the position of both theirs and the enemy's units, and the possible related actions they can do.

6.1.2.5 Overworld

Many games in the genre also feature a split between being in combat and navigating an overworld, and the overworld usually also serves as a hub where the player is able to prepare for the coming battles, e.g. managing the unit inventory or buying new spells. The overworld often contains a variety of locations to visit, objects to interact with, and is used for spatial progression in the game. A prime example of this is seen in the Heroes of Might and Magic series, where everything from the game's economy to town building and unit management are contained to the overworld.

6.2 The Game Design

The scope of the design and the choices within are made to facilitate a time pressured development schedule, as well as taking into consideration that players will have a shorter time with the game than if it was a release they played at home for themselves. This makes it both more feasible to develop, and possible for the players to learn in the short time they have with it. This is due to both amount of content, and the fact that the game will only be played in a test session.

An outline of the most basic mechanics is listed below. However, these will not be explained in-depth, apart from the combat map, since it introduces the strategic elements. As the focus of the project is on the complex mechanics, only the complex mechanics will be explained in greater detail.

The game will use a classic turn based system (I Go, You Go!), since, if we look at the investigation of the genre in the previous section, nothing apart from artistic choice indicates which system would be best suited for this project. Therefore, going with the simplest solution is gainful for development time.

The basic mechanics and features of the game are:

- **An overworld**
  A hub for the player to plan and prepare for coming battles, and our method of spatial and story progression. This overworld includes player movement and interactable objects or buildings to facilitate planning and unit management from the player.
- **Unit management**
  A roster of recruitable units that the player can unlock and bring into battle. This will take place at a shop in the overworld, and in an inventory management menu.
- **Combat board**
  A map separate from the overworld in which battles will take place. This will resemble the board from a board game, where the places the units can move between are predetermined.
- **Unit combat**
  A combat system where the player and an AI enemy can attack each other's units. Each side will also have a set of towers that can be attacked, destroying these function as the win condition for the battle.
- **Unit summoning**
  The game will also feature a system that determines how units are activated in battle. Each side will have a replenishable currency that they can spend to bring a unit from their battle inventory onto the combat board.

The game will also feature two complex mechanics which are the focus of the project and the testing:

- **Different unit types**
  Each unit will be a specific type, altering the way it interacts with some of the other types of units
- **Special abilities**
  A unit having a property or an ability which changes the way it can be used on the battlefield.

A total of 13 units will be featured in the game, which provides room to have several units with type relationships and special abilities, each with a basic and strong variant. There is one 'normal' type unit with neither a special ability, a type relationship, nor a stronger version of itself.

Unit Type Relationship:

- 2 Water Types
- 2 Fire Types
- 2 Nature Types
- 2 Lightning Types

This is a common variation of type relationships, seen in many games, where they follow four natural elements, which is a good instinctive indicator to help the player figure out what is strong against what. Figure 16 shows a graph of the relationship between the types.

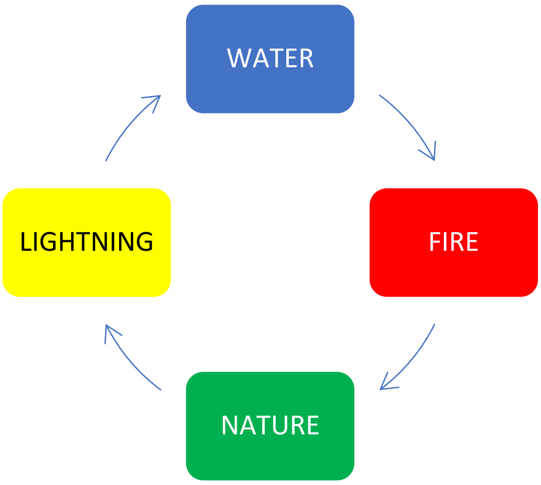


Figure 16 - A graph over the type relationships

When one unit is strong against another unit, the other unit is considered a weak unit.

When a strong unit attacks a weak unit, it will deal twice its normal damage. However, if a weak unit attacks a strong unit, the weak unit will only deal half its normal damage.

Units with Special Abilities:

The units with special abilities are 'normal' type, and do not have a relationship with any other types, and will both give and receive the indicated amount of damage with no multiplier.

Rather than having a spell system, the design features two forms of units with an inherent special ability. This is done to lessen the amount of actions the player has to memorize in order to utilize the feature, so that it will be easier for them to focus on the unit combat mechanics.

The units we have created, were made with a specific synergy in mind, so that they fit within the rest of the unit roster, and has a function between them as well.

- **2 Stone Units**
  The stone unit can be activated so it becomes much harder to kill, however it cannot move or instigate attacks. Its intended purpose is to block the path of enemy units.
- **2 Shadow Units**
  The shadow unit can move through other units. This allows the player to play around with board management, while also offering up a counter to a stone unit used as a roadblock.

The Combat Map

The combat map will try to utilize these elements by being similar to a board game with pre-determined interconnected spots that the units can move between. This allows us to control the flow of a match and incentivize varied unit usage by having multiple access points in three paths, as seen in Figure 17, which heightens the strategic requirements of the player.

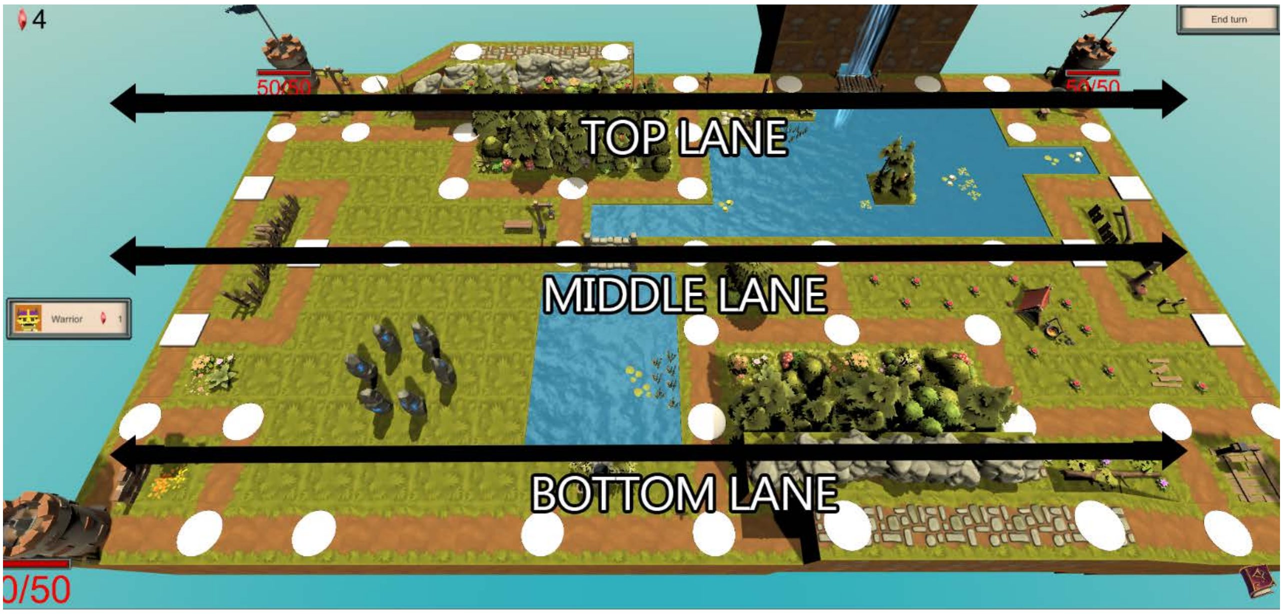


Figure 17 - An image of the combat map, showing the 3 lanes

7 DYNAMIC TUTORIAL

The idea of the dynamic tutorial is to have a system which can determine whether a player does or does not understand a specific mechanic. If the system determines that a player does not understand the mechanic, it should then help the player by giving information about the mechanic. The dynamic tutorial system will utilize text boxes, similarly to how the static tutorial will use them.
Additionally, the player should not be able to move on from an area containing a combat that attempts to teach a mechanic, before he has learned that mechanic. This will be achieved by a different part of the same system, to check for understanding at the end of a combat, to understand if it should let the player move on. If the system deems the player does not have an understanding of the specific mechanic, he will be forced to reenter the combat which will attempt to teach him this mechanic again.

7.1 THE SYSTEM

The system presented in [5.3 Adaptive Tutorial] revolves around the knowledge of the player, and it is therefore important to collect, and analyze the “correct” data. Gathering the data should therefore be focused around the understanding/knowledge of a player within the game. A problem with adapting tutorials is the nature of them, they are presented in the very beginning of a game, and therefore we will have little knowledge about the player before the dynamic system would need to trigger information. This means that we will need to design the system without relying on categorizing the player, meaning we will need to assume the data which is analyzed can describe a player’s understanding properly.

To be able to understand which areas players seem to have a difficult time understanding, and what assumption can be made, a preliminary test was made using the game developed for this project.

7.1.1 Preliminary Game Test

The test we refer to here is a test which is not a part of the iterative tests described later in this report. The goal for this test was to play test to see if there were any game breaking issues that needed to be fixed, and to understand what aspects should be considered to be included and focused on within the tutorial system. This test also allowed us to obtain an understanding of which mechanics seemed to be intrinsically understood by the player.

For this test, there was no tutorial implemented. This means that the observer functioned as the tutorial.

The test was conducted using observation with in-depth questions where it felt necessary. The observer told the participants that he would not help or give advice per request of the participant, but as deemed necessary by the observer himself. The situation in which the observer did answer questions or gave advice was if it demanded more information. Specifically, in regard to either a stalemate due to misconception of how the game functioned, or frustration which lead to the participant not being able to continue playing. Additionally, the observer would comment if the participant experienced any bugs which lead to confusion.

The observer would ask pre-decided questions at specific times throughout the process. He would also ask questions if they seemed relevant for the given situation. Some of the pre-decided questions were:

After a few turns ask:

- **What are the win and lose conditions?**

After a few additional turns:

- **When do you gain summon points?**
- **When is it possible to attack?**

At the end of the test:

- **Could you explain how the different unit types interact with each other?**
- **What did the number above the health bar mean?**
- **Are there any units that have special abilities compared to others?**
- **Was there any information you felt could have helped you during the game?**

We believe that by asking these questions along with in-depth digging questions, as well as with other questions asking about game related elements, it would be possible to obtain an understanding of what the participants felt they needed the most through a tutorial.

The participants provided these four different elements that they would have liked to have more in-depth information about, in order to feel they have a better understanding of the game.

- **Strength/weakness system**
- **Special units**
- **Win condition**
- **How you get summon points**

Based on this we found the most interesting mechanics which might benefit from being dynamic, would be the strength/weakness system, and the shadow unit, of the special units. This was chosen due to the nature of these mechanics, that the participants should be able to gather some understanding of them initially, and then if it seemed that they did not understand the mechanic fully, the system would be able to give them more information about it.
The other mechanics that were mentioned either in the points above, or in [6.2 The Game Design] are considered to be a one-time information necessary in order for the player to understand the mechanic completely.

There are two inherent problems and concerns regarding the data collection and data analysis of the system. These are:

Which data entries are required and important?

How can the data tell us what the player does not understand?

By looking over the different possibilities, we believe that we will be able to form a coherent system which will be able to fulfill the requirements of the idea of the dynamic tutorial.

7.1.2 Data Gathering

In order to understand the player's knowledge about the different elements, multiple data points should be gathered. Although multiple points may be valid, we will go through the options and limit the list to those that make the most sense and are most feasible to implement.

7.1.2.1 Data to be Gathered

The data points which are mentioned below have been gathered through observation. Additionally, we have also found data points which might be of interest, based on elements which were used in the three examples of fitness function in [5.2.1 Examples of How Fitness Functions Can be Used].

7.1.2.1.1 Strength/weakness

- **Strong hits**
  The amount of engagements the player enters where he is at an advantage (also possibly engagement the enemy enters where the player is strong).
- **Weak hits**
  The amount of engagements the player enters where is at a disadvantage (also possibly engagement the enemy enters where the player is weak).
- **Outcome of engagement (state of game)**
  Both, if a unit got killed (both enemy and player), and the offensive state (player can attack towers now or soon) and defensive state (enemy can attack towers now or soon).
- **Strategic placement in the combat environment (state of game)**
  If the player is blocking a path by putting an arbitrary unit in front of the tower (this might be needed to avoid influence about weak/strong if the goal is to stall)
- **Possibly preferred unit**
  To understand if the player has a preferred unit (this unit might be stronger than the others), and to negate having to tell the player information he already knows but chooses to ignore.

7.1.2.1.2 Shadow Unit

- **State of the game**
  Would special units allow for better optimization, e.g. stone blocking path to tower which allows player to build up more to defend, or enemy has multiple stone units/strong blockades which a shadow unit can bypass and get “free damage” on tower.
- **Units’ special ability used during combat**
  If units have been brought and their special ability has been utilized.

For the first element, the strength/weakness system, it is believed that by gathering information about how many strong and weak attacks the player makes, it will be possible to make a decision about the player’s knowledge. For the second, the shadow unit, we believe that looking at how often the player uses the special ability will make it possible to determine if they have an understanding of the shadow unit’s special ability.

7.1.3 Data Analysis

In order to understand how the system should react; the system must understand incoming data and be able to make a decision based on this. To form a functionality as such requires analysis of game data, and interpretation of the incoming game data. This could be done by modelling the player to specific categories, by utilizing a player model method. However, if we were to use a player model from [5.1 Player Experience Modelling (Data gathering)], the amount of assumptions that we would have to make about the player early on in the playthrough, would, in our opinion, introduce a large amount of bias or problems. Therefore, we will attempt to make a system which revolves around looking at specific numeric values, with less room for interpretation. This will then trigger dynamic system events.

The design of the dynamic system should depend on two elements, firstly a fitness function which utilizes different data points to calculates a score. Secondly one or more conditions which looks at the score and can trigger an event if the score meets the condition(s).

Type relationship – information trigger

$$playerScore = x_g - x_{bnk} \qquad aiScore = y_g - y_{bnk}$$

In order to best determine whether or not the player understood the different unit relationships, we believe that using the difference between good trades (strong hits) x_g, y_g and bad trades (weak hits) that were not kills x_{bnk}, y_{bnk} will provide the best opportunity. In addition to this we believed that it would be beneficial to also base their understanding of the mechanic based on how the AI was doing. This is believed to give a more overall view of how well the player performs, and if he is able to utilize this mechanic to his advantage, both for his attacks and for how effectively he can keep the AI from utilizing this mechanic.
The system would trigger information if any of the following conditions were met, based on the score of the fitness functions:

$$playerScore < -2 \quad or \quad aiScore > 2$$

The conditions were designed based on our own experience of how the game felt when playing it ourselves, and how fast you would understand the implications of your trades, as well as observations from the preliminary game test. In this test, the participants figured the relationship out very quickly, and they seemed to have an understanding of the game mechanics when the score would be within these two conditions, meaning the event would not be triggered.

The information the dynamic system will present if this event is triggered, will contain an explanation of which unit types are strong against the others.

Type relationship – does not understand mechanic

The trigger for a player not having learned the type relationship, is examined in the exact same way as the mechanic's information trigger.

Shadow mechanic – information trigger

$$score = x_{wmt} - x_{mt}$$

Where x_{wmt} denotes the amount of times any player controlled shadow units has moved without moving through another unit, and x_{mt} denotes the amount of times any player controlled shadow unit has moved through another unit.
The system would trigger information if this condition was met:

$$score > 2$$

This condition was also designed based on our experience with the game and the shadow units, in conjunction with the observations made through the preliminary game test.

We implemented an additional condition which was not based on the score, but based on a very effective way of using the shadow unit during a combat. The condition is if the player uses a shadow unit to move through another unit and then attack a tower. This describes a very effective and optimal way of using a shadow unit. We believe that if the player used the shadow unit in this manner it describes a fundamental understanding of the shadow unit mechanic.

The information the dynamic system will present if this event is triggered, is an explanation of the shadow unit's special mechanic, specifically that they can walk through other units.

Shadow mechanic – does not understand mechanic

This examination uses the same method as the information trigger. There is however one additional check point which is assessed before the information trigger conditions are examined.

The additional checkpoint is if the player has ever summoned a shadow unit. If the player has not summoned a shadow unit, we believe that he does not understand the mechanic and the system should therefore attempt to teach the player this mechanic again.

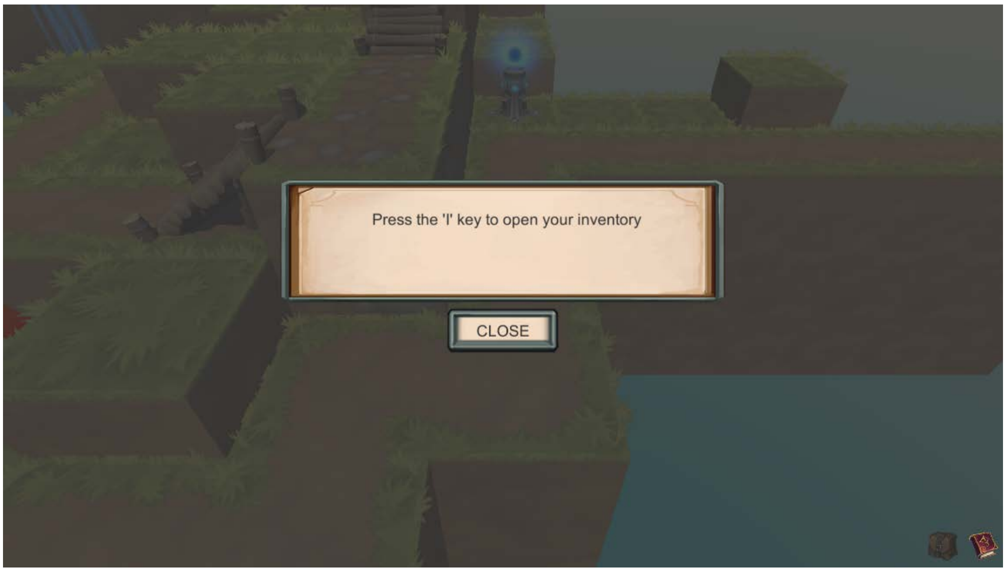
The way the dynamic tutorial system is designed, is believed to provide a player with the appropriate information when he does not understand a mechanic. This will help the player to understand a mechanic he has trouble with understanding, and allow for easier progression through the game.

8 FIRST ITERATION

This chapter will describe the first iteration of both our static tutorial design, and the procedures from the first preliminary test on this design.

8.1 TUTORIAL DESIGN

Based on the test game's design, we are able to construct a tutorial that will teach the player how to play the game. As this project does not focus on the specific way of communicating information to the player, but rather on the system that determines when and what will be told, the tutorial will feature only text boxes as its means of communication. While this form of information delivery may not be the most liked by players, as is advised in our investigation of [3.2 Creating a Good Tutorial], after considering the implementation possibilities, we find this to be the most achievable of the options. However, if we receive bad feedback in the initial test regarding the delivery method, we will reconsider. An example of such a text box can be seen in Figure 18.



Press the 'I' key to open your inventory

Figure 18 - A text box in the game showing how to open the inventory.

The triggers for tutorial information will be predefined. We will try to make sure that information about a mechanic is presented as close as possible to when it needs to be used for the first time. An example of this could be, before the player enters his first combat he is informed about how you acquire units. By acquiring units, the player will then more easily be able to defeat the first enemy he encounters.
This is done to avoid front loading information as explained in [3.2 Creating a Good Tutorial].

This will allow us to get started in the iterative process quickly, and adjustments to the delivery method can be made if found necessary. Text boxes are also a widely-used method in many games, and from initial discussions about implementation is found to be the easiest to implement for us, and the most feasible to create an adaptive system around. We will however aim to make the presented text as concise as possible, while still providing the necessary information, in order to lessen the amount of text the players receive [3.2 Creating a Good Tutorial].

8.1.1 First Iteration

First, we need to look at what the player needs to be taught in order for them to understand and play the game. This was done using both information gathered from the [7.1.1 Preliminary Game Test], and by dissecting the elements of the game design. These can be split into two groups, the basic and the complex elements. In the basic we have:

- Player movement
- Win and lose conditions
- Unit acquisition
- Inventory management
- Combat engagement
- Unit summoning
- Unit movement
- Unit combat

These basic elements can be considered the minimum learning requirements of how to actually play the game, while the complex mechanics can be considered the elements needed to be learned in order to master the game. The complex mechanics are:

- Unit type relationships
- Unit special abilities

These are also the two systems which the bulk of the test will focus on, as described in [10.2 Final Test]. The first iteration will however not feature any dynamic elements, since the purpose is to shape subsequent tests and discover flaws.

The player will be tasked with going through four battles, each fight introducing new enemy types and unlocking new types for the player to use. The first battle will focus on teaching basic combat controls, the second will start to introduce type relationships, the third will have the full range of type relationships, and lastly the fourth will teach unit special abilities. A layout of the overworld map can be seen in Figure 19.

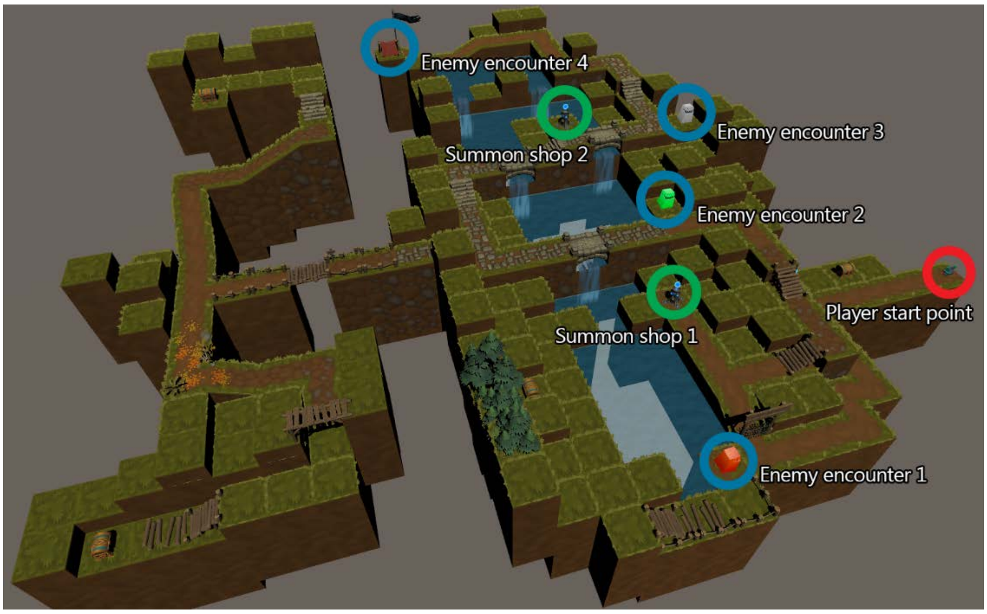


Figure 19 - Overworld map with highlights

8.2 First Preliminary Test

After the initial tutorial design was completed, the first preliminary test was run. This section details the procedures and how they were performed.

Test Goal – The goal of this test was to gain an understanding of how the different tutorial elements presented their information. Specifically, it was of interest if the participant felt that some information was missing from the tutorial, if the information presented by the tutorial was understandable, and any general comments they had after playing the game. Also, it was important to gain an understanding of how much of the information the player understood based on other game experiences and if any of the tutorial elements contained information that was not necessary for the player to receive. This was done to help shape the coming tests, both the questions which should be asked, and to reiterate over the implementation of the tutorial and its presented information. In addition to this it was also of interest to understand how the flow of the tutorial was experienced by a new player.

Implementation goal – The implementation goal of this test was to assess the difficulty of the AI, the player's understanding of the controls, and to have a non-developer play the game to locate unforeseen bugs or other problems within the game. Another implementation goal of this test was to obtain additional observational data to possibly change the way the dynamic tutorial was imagined to be implemented. We were interested in what signified the player understanding a game mechanic.

8.2.1 Test Procedure

The first preliminary test was done having two different observers. One would function as a guide to ensure the participant would not get frustrated if there were any missing information preventing him from progressing. While the test was ongoing, an additional observer sat and took notes of behavior, bugs encountered, questions asked, and general interaction with the game.

The observer who functioned as a guide explained that he would not answer questions regarding game play mechanics such as "What is strong against the water type?", but would answer questions such as "How do I deselect a selected unit?". The participant was still encouraged to ask questions if there were anything he was unsure about, so the second observer would be able to take note of the player's confusion at that specific time in the game.

Before the participant started playing the game the observer explained the implications of the paragraph above, as well as described in a rough sense what the participant should expect to do. The participant was also asked two demographic questions: *Do you consider yourself a gamer?* And *Do you play turn based strategy games?*

Then he would play the tutorial in which the guide would stop him after each combat to ask three questions.

- **Do you feel there was anything "too simple" to you, to be presented in the tutorial?**
- **Was anything intrinsically understandable, where a tutorial was not needed?**
- **Was there any of the presented information that you did not understand until you tried it/used it?**

After having completed the tutorial, the participant was asked a series of questions in three different categories:

- **Self-assessment of game understanding**
- **Questions about specific elements within the game, to test the participants understanding**
- **General comments and feedback**

Self-assessment of game understanding – This was done using three different overall questions:

- **Do you feel you have a general understanding of how the different mechanics works?**
  This was asked to understand the players own perception of his understanding of how the game functions in its whole.
- **Do you feel the game has prepared you to go past a tutorial and into the "real" game?**
  This was asked to understand if the player perceived that his abilities had been developed enough in order to continue defeating increasingly difficult enemies, if the game would continue.
- **Were there any elements that confused or otherwise influenced your play experience?'**
  This was asked to understand any confusing elements, and to give the participant a chance to describe these.

Questions about specific elements within the game, to test the participants understanding – Multiple questions were used pertaining to different game mechanics, three of these are:

- **What was the relationship between fire and water units?**
  This question was asked to understand their knowledge about the type interactions between the units during combat.
- **Are there any units that have special abilities compared to others? And if so please elaborate**
  We asked this question in order to check if they understood that some of the units had additional properties compared to the other units. And to see if they understood what these specific properties were.
- **When do you gain summon points?**
  This question attempts to gauge the participants understanding in relation to one of the more simpler mechanics, to see if he understood the mechanic fully.

General comments and feedback – This section focuses on anything in the test, and also anything the player liked/did not like in the tutorial. Multiple questions digging into different experiences and elements were asked, three of these were:

- **How did you find the flow of the game to be?**
  To get firsthand knowledge about how the participant specifically felt the flow of the game was.
- **Was the tutorial information intrusive to your play experience?**
  This was asked in order to figure out if the participants felt that the way the tutorial is implemented had an influence on how much they enjoyed the game, and/or their performance within the game.
- **Was the tutorial information understandable?**
  We asked this to understand if the participant noticed anything odd in the tutorial information boxes, specifically something he did not understand, which would then need to be rephrased to be understood more easily.

All of these questions were asked during a face to face interview, in which the interviewer had the opportunity to ask follow-up questions, to dig for more information, if needed.

9 SECOND ITERATION

Based on results from the first iteration of tests, a number of changes will be made to the tutorial design and this second iteration of the test in order to get better results in future tests. These will be described throughout this chapter.

9.1 TUTORIAL DESIGN

Firstly, the unit roster of several enemy encounters will be changed in order to give a better flow in the tutorial and minimize the chances of players encountering a stalemate situation, where the player is perfectly countered in each move they make and neither side can make any progress. This was noted by some participants, as they would spend a great deal of time on one encounter since the enemy units and the players own countered each other perfectly.

Additionally, what will be taught in each battle will also be changed, as well as adding a fifth battle. The first two encounter will remain the same, however in order to get better measurements on shadow units they will now be introduced along with stone units in the third encounter. The fourth and fifth encounter will start mixing both shadow, stone and elemental type units together to get data on how well the player has understood the mechanics and is able to use them in a real battle scenario.

Lastly, the participants had many comments regarding the overworld level, in that it was too big and hard to navigate, especially in terms of knowing where to go for the next fight. To rectify this the map will now be split into five smaller islands that each hold a single encounter and a shop which the player can recruit new units from. Figure 20 shows an overview of island 2 from the new overworld map layout.

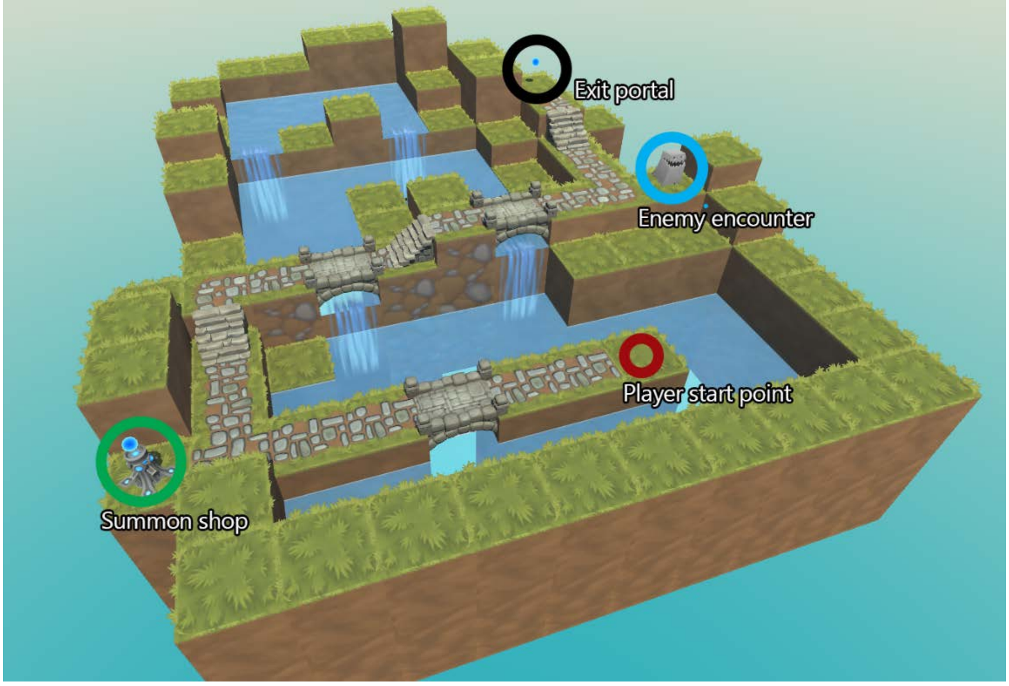


Figure 20 - Island 2 from the new map layout

9.2 Data Gathering

For this project, it is important to gather data about the player's behavior, in order to be able to know what he did while playing. For this reason, a data gathering system was implemented. The system would receive data based on many different events, e.g. whenever the player (or the AI) made an attack, or if the player was shown a dynamic tutorial textbox.

Some of this data was used during play for the dynamic tutorial system, to analyze whether or not the player required additional information. Furthermore, it might be interesting to be able to look at the data afterwards, therefore an online database was set up, and when the player was finished playing the game, all the data would be send from the computer to the database.

These are the data points which were gathered:

- **ID**
  A unique identifier, containing information about whether the player is in the control or treatment group. It is also passed along to the survey, which makes it easy to correlate a player's response with their in-game data.
- **Attack data**
  An extensive list of every attack that has occurred throughout the entire game.
- **Summon data**
  A list of every unit the player has summoned throughout the game.
- **Roster data**
  A list of the units the player brought to each battle.
- **'Did not understand mechanic' data**
  The amount of times the dynamic system deemed that the player had not learned a specific mechanic by the end of a fight.
- **'Information trigger' data**
  The amount of times the dynamic system decided the player should receive additional information about a mechanic.
- **Shadow unit data**
  The amount of times a shadow unit was moved through another unit and the amount of times a shadow unit was moved without moving through another unit.

All of the gathered data might not be used, but it was decided that it would be better to have and not use it, than wanting to use it and not having it. The data could also be important in further development of the system as every player's actions could be retraced.

9.3 Second Preliminary Test

Based on the changes to the tutorial design, and changes to the test parameters, the second iteration of the test was run. This section outlines the procedures and execution of this test.

Test goals – The goals of this test were the same as the first preliminary test with three additional goals. These were: testing improvements in the tutorial, testing the functionality of the dynamic tutorial, and testing the questions in the survey.

Implementation goals – This second test's implementation goals also incorporate all of the same goals from the first iteration test with two additional goals, the dynamic tutorial, and player data being sent to the database.

Testing the dynamic tutorial was done forcing the participants to play through the dynamic version of the game. Through this playthrough we observed the behavior of the system, e.g. how many times it showed up, did it make sense that it showed up there, or if there was something wrong with the way it was implemented.

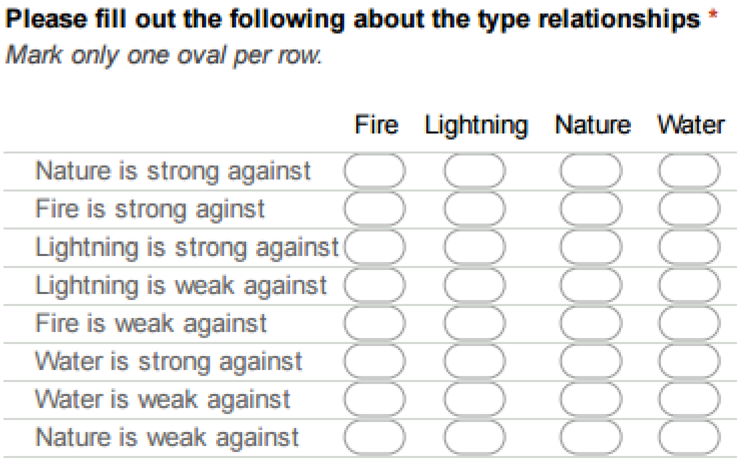
9.3.1 Additional Test Procedure

This test followed roughly the same procedure as described in the [8.2 First Preliminary Test], however the data gathering in the end was done through a survey and not a face to face interview. This was done to gain input on any questions on the survey which might not be completely clear. The full list of questions can be seen in [16.2 Appendix 2 – First Questionnaire].

After looking through the results and observations from the first preliminary test, it was decided to add additional questions after the initial questions had been converted to a survey. In order to avoid having the players realize that mechanics they did not understand existed, multiple yes or no questions were devised about different mechanics. If the participant then answered yes to one of these questions they would be lead to a follow up question to try and gauge how well they understood that specific mechanic.

Additionally, multiple questions were added or reformulated for in the **Questions about specific elements within the game, to test the participants understanding** and **General comments and feedback**, these are:

- **Is there a relationship between the different unit types?** If the participant answers yes to this question he is asked to fill out the following table in Figure 21.



Please fill out the following about the type relationships *

Mark only one oval per row.

	Fire	Lightning	Nature	Water
Nature is strong against	○	○	○	○
Fire is strong aginst	○	○	○	○
Lightning is strong against	○	○	○	○
Lightning is weak against	○	○	○	○
Fire is weak against	○	○	○	○
Water is strong against	○	○	○	○
Water is weak against	○	○	○	○
Nature is weak against	○	○	○	○

Figure 21 - Table to fill out if previous question was answered with a yes

- **Could you see why you had the units in battle, you had?** – The answer for this question is in a text format, in order to allow for an in-depth answer.
  This is a question which tests the participants' knowledge in one of the simpler mechanics.
- **Do you consider this to be a good tutorial? –** Followed by a text field asking the participant to describe why he answered the way he did.
  This is asked to understand what the participant thought about the tutorial.
- **Is there any information in the tutorial you feel did not need to be told? –** The answer for this question is in a text format, in order to allow for an in-depth answer.

Apart from the survey, the test concluded with a few questions regarding the survey done in a face to face interview format, these questions were:

- **Did you understand all the questions in the survey?**
  This was done to check the phrasing of the questions in the survey made sense to real participants, and that they were understandable by regular people (gamers).
- **Was there anything that was unclear in the survey you just filled out?**
  Similar to the question above, however phrased a little differently. This was asked in order to get the participant's impression of the survey, and if there might have been a question or an element that did not make sense.

Additionally, the observer paid attention to how the participant filled in the survey, and if he saw any odd answers while the participant was filling it out. The observer then asked the participant how he understood the question, and then discussed briefly the intended meaning of the question to possibly be able to phrase it better with the help of him.

9.3.2 Results and Changes

The results of this test lead to multiple changes both to the **survey** and the **dynamic tutorial**.

Survey – During the test none of the participants mentioned anything they were unsure about, nor did they seem to give any odd answers to the posed questions. The changes to the survey were done based on our need to understand the participant's knowledge. To obtain a better understanding another method apart from questions would therefore be utilized; showing an image of an in-game event, with a question such as "What can the fire unit do on its next turn?" this will be followed by a subset of actions which can either be possible or not. A question such as this can be seen in [10.2.3 Test Procedure].
Additionally the survey provided very little demographic knowledge, and we therefore added age and gender to the initial questions of the survey.

Dynamic Tutorial – The dynamic tutorial showed to have multiple problems when being tested. This test revealed different issues with the dynamic tutorial:

- **Fitness function for shadow**
- **Fitness function for type interaction**
- **Alterations to the AI**

Fitness function for shadow – The fitness function for shadow revealed to be too sensitive, meaning the shadow unit's dynamic tutorial element would show by simply moving a shadow unit once or twice after it was spawned. It would show even if the player had had no real strategic advantage of moving the shadow unit through other units. This nullified the way the dynamic tutorial works, as it is supposed to know if the player understands the mechanic. Therefore, the fitness function for this tutorial element was changed to:

$$score = x_{wmt} - x_{mt} - x_{spawned}$$

Where x_{wmt} denotes the amount of times a shadow unit has not moved through a unit, x_{mt} denotes the amount of times a shadow unit has moved through a unit, and $x_{spawned}$ denotes the amount of shadow units spawned. The trigger limit was kept at three.
The reason for subtracting the amount of shadow units spawned was founded in, the shadow unit rarely had the possibility of moving through a unit during its first turn due to the state of the game. Therefore,

triggering the tutorial element for situations where the player actually did not have the opportunity to actually use the shadow unit's mechanic.

Fitness function for type interaction – The fitness function for the type interaction revealed to be too dependent on the behavior of the AI. Basing the dynamic tutorial elements trigger on the AI's good trade/bad trade differences, did not seem to indicate whether or not the player understood the mechanic but more of a reflection of the player's skill of maneuvering and strategic placement, in combination with the AI's design to only ever instigate good trades. We found the problem to be twofold, the first was how the condition of this tutorial element was designed, and second the way the AI was designed (this will be explained below). This was therefore changed to always take the player's actions into account.
The trigger was therefore changed to use either of these two conditions:

$$playerScore < -2 \mid\mid playerScore - aiScore < -2$$

Using these two different conditions allows for the tutorial element to be triggered based on the player's behavior, while still having the amount of good trades the AI has done within the condition.

AI alteration – The behavior of the AI up until this test had been made with careful aggressiveness. The AI had only made trades which greatly benefitted it, resulting in almost zero cases where the AI would ever instigate an attack if it would be a bad trade for the AI.
This was not what was desired of the test as it had two impacts, firstly it was too good at exploiting game mechanics, sort of unknown or at least unfamiliar to the player. Secondly, it also increased the average time of an encounter by a fair amount.
To combat this, the AI was made more aggressive wanting to instigate more attacks, even if the attack wasn't necessarily a good trade to make.
The way the AI was balanced before this change also had an impact on the dynamic system. The reason being that the condition for the dynamic system was based on the AI's score alone, and since it primarily only made good trades the AI score would only increase, triggering the condition at times where it would not reflect the player's understanding of the strength/weakness mechanic.

10 Final Iteration

Throughout this section, we will describe how the final iteration of the static tutorial was designed, and what it specifically teaches. We also go over the test setup for the final test, and what we hope to be able to determine by analyzing results of the final test.

10.1 Final Tutorial Design

This section will present the final tutorial design in full detail, which has been changed from the first and second iteration based on participant feedback and data. The specific tutorial which is presented during this section is the static tutorial, that will not adapt to the player.

In order to create a pleasing flow in the tutorial, the elements that are taught are split into sections, so that the player will be introduced to one element after another and not overwhelm them with front-loaded exposition [3.2 Creating a Good Tutorial], starting with the simplest information and introducing new features gradually, working towards the most complex mechanics.

10.1.1.1 Phase 1 – Basic Control

This phase will teach the player how to control their character and to navigate the simplest elements of the user interface. This will be introduced to the player within the starting seconds of their playthrough, as it is required for them to even being the game, this includes:

- Player movement
- Unit acquisition
- Inventory management
- Combat engagement

This information is presented on the first island, before any combat encounters take place.

10.1.1.2 Phase 2 – Basic Combat

The second phase will start when the player enters the first combat of the game on the first island. This section will teach the player about the general unit control, and the win condition of the combat. The enemy's combat ability will be low, as it is expected that the player will not be familiar with advanced tactics yet. This phase covers:

- Win and lose conditions
- Unit summoning
- Unit movement
- Unit combat

10.1.1.3 Phase 3 – Type Relationship

In this phase, on the second island, we start going into the complex mechanics, and expand the combat by adding type relationships. To teach the player about both strength and weakness, the player and computer will each have two units who can counter each other, the player will have fire and lightning units available, while the enemy has water and nature. This ensures that the player will counter both something that is strong and weak against their units.

10.1.1.4 Phase 4 – Special Abilities

The fourth phase, on the third island, will introduce the last mechanic to the player, unit special abilities. At this point the player has unlocked a variety of different units, so in order to introduce it properly the enemy will only have stone units, which makes it very difficult to beat without using shadow units. This forces the player to use the new units with special abilities, instead of relying on old units and old strategies. This also removes the type relationship units from the battle, giving the player opportunity to focus on one mechanic and on learning how to use that.

10.1.1.5 Phase 5 – Mixing Units

The last phase, which takes place on the last two islands, mixes the unit types that the enemy has between all types, challenging the player in the form of variety, tasking them to use all the mechanics they've learned at the same time. This phase also serves as a point to see whether the player has forgotten some older mechanics, now that new ones has been introduced.

10.2 Final Test

The final test iteration was conducted using both versions of the tutorial, the static and the dynamic, and incorporated both the changes mentioned in the section above, and alteration to the test procedure, which will be detailed in this section.

10.2.1 The Overall Test Goals

The goal of the test is to obtain an understanding of whether or not there is a basis for a dynamic tutorial system which helps players understand complex mechanics. The reason for this is to lessen the amount of time a player, who is able to understand a mechanic quickly, spends on it and therefore allows for faster progression to the real game, and reduces the information noise for experienced players [3.2.3 George Fan].

In order for us to be able to understand if the dynamic tutorial works, and has a positive impact on players, this null hypothesis was formed:

Players who have played the dynamic tutorial has the same, or more knowledge of the game compared to the players who have played the static tutorial.

If this hypothesis is not disproven, we believe that a basis of a dynamic tutorial would exist and therefore warranted more research, both on a larger scale and in games with more complex mechanics.

The test in this project is done using a game developed for this project, which incorporates two mechanics which are of a more complex nature, that players would not necessarily know inherently from playing other games in the past. These are:

- **Shadow units**
  The unit with the special ability to move through other units
- **Type interactions**
  How the different elemental types interact with each other, specifically which are weak or strong against another elemental type.

In order to obtain an understanding of this, a test was devised comparing a group of participants who had experienced a static tutorial the control group, versus participants who experienced a dynamic tutorial the treatment group.

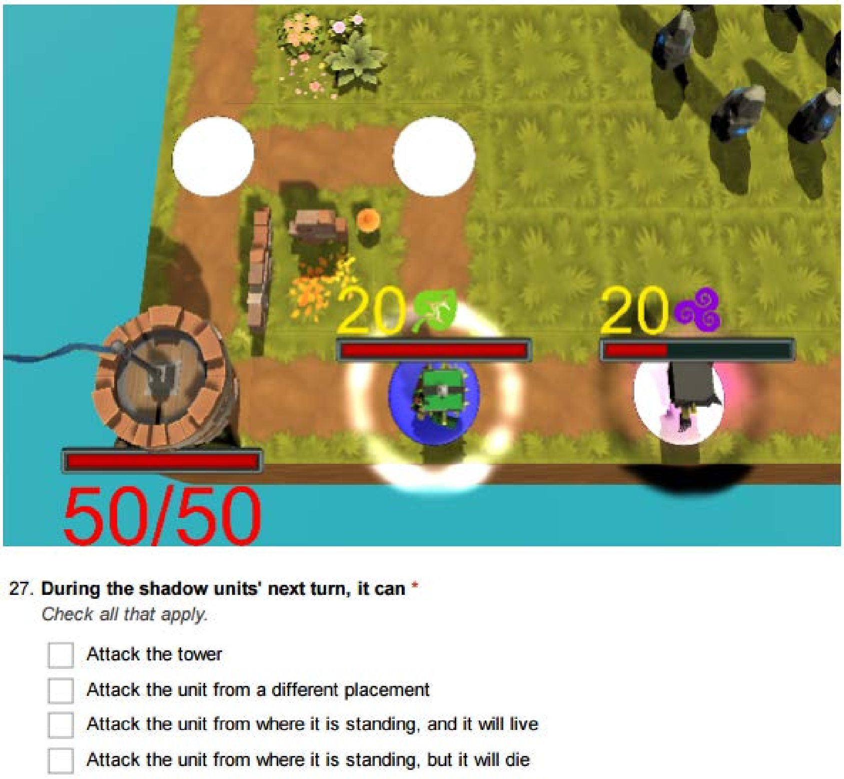
10.2.2 Test Measurements

Understanding the knowledge level of a participant through in-game data collection, can be difficult. It becomes increasingly difficult when there is a limited amount of content, which limits the amount of data that can be collected about a player. Therefore, in order to gain a better understanding a survey was also used, which aimed to test the player's knowledge of different game mechanics and elements. By analyzing the result of such a survey would grant a more specific understanding of a player's knowledge of the game. By using this in conjunction with in-game data, we believe it will be possible to see a correlation between the in-game data, and a player's understanding of the game. Furthermore, it would also allow for a very specific and quantifiable set of data which can be compared between the two groups.

10.2.3 Test Procedure

The test features both a face to face interview test and an online test utilizing the survey.

As mentioned in the [9.3 Second Preliminary Test] an additional element was introduced to the survey, presenting the participant with images and getting them to mark what outcomes are possible in the scenario presented in the image. A total set of five images were generated, two of these can be seen here:

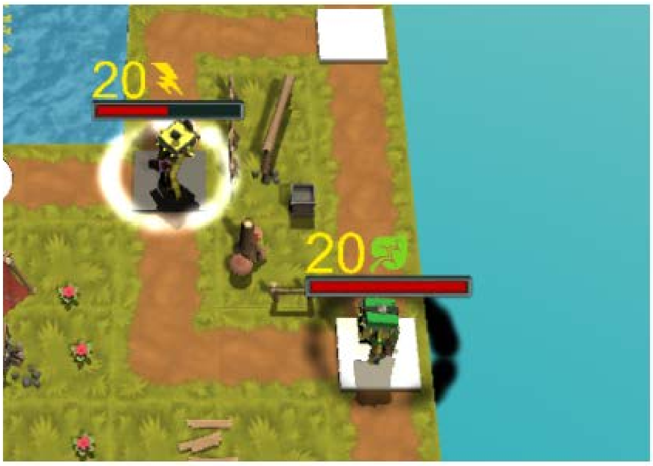


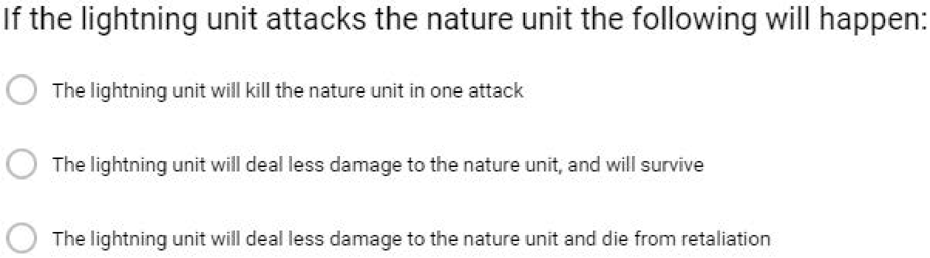
27. **During the shadow units' next turn, it can** *

Check all that apply.

- □ Attack the tower
- □ Attack the unit from a different placement
- □ Attack the unit from where it is standing, and it will live
- □ Attack the unit from where it is standing, but it will die

Figure 22 - Tests the understanding of the shadow unit




If the lightning unit attacks the nature unit the following will happen:

○ The lightning unit will kill the nature unit in one attack

○ The lightning unit will deal less damage to the nature unit, and will survive

○ The lightning unit will deal less damage to the nature unit and die from retaliation

Figure 23 - Tests the participants understanding of a specific type of type interaction

Figure 22 tests the participant in their understanding of how the shadow unit can interact with the world. Figure 23 tests the participant's knowledge in regard to the way different types interact with each other.

10.2.3.1 Interview Procedure

A set of rules were devised for the face to face part of the final test, to both try to keep the amount of time the participants spent on the test down, and to ensure that all of the tests followed the same structure with the observer acting the same way in all of them.

The role of the observer was to both observe behavior, and to assist the player in an acceptable way to ensure smooth guidance through the game if the participant got stuck during the playthrough. The observer followed this set of rules:

- Explain to the participant that you will be observing his playthrough.
- The observer is allowed to assist the player if there is clear frustration (e.g. the player does not understand that he needs to click on a unit to buy it) or the participant specifically asks a question that has no impact on the area that is being tested (e.g. How do I know which units are mine?).
- The observer is not allowed to give any hints or advice as to which tactic a player should use.
- The observer should take notes if any bugs occur.
- The observer should note if there are any mechanics that seem very difficult for the player to grasp.
- The observer functions as the interviewer at the end of the test.

By following these rules, we believe that the observer is able to provide enough guidance for players who might have a hard time grasping simpler mechanics. This also allows for a reduction in time spent testing each participant, and having similar observer behavior across all of the tests.

The face to face interview and the online survey followed the same pattern. The questions which were asked in the face to face interview were taken directly from the online survey. This allows for us to understand if there are anything which is unclear or misunderstood. It also allows for more specific and guaranteed description of all the described questions, while also giving the interviewer the possibility of digging for more information about specific subjects by asking additional questions.

10.2.3.2 Participant Gathering

In order to obtain a desirable amount of data, we need a fair number of participants. To obtain participants that actually try the game through their own free will, fall into the gamer category, and are willing to spend 1-2 hours on testing the game, we believed that it would be easier to gather participants by offering entrance into giveaways.

We have had success with this method in previous projects. A benefit of this is it makes people more inclined to join as they have a chance to gain personal rewards while also assisting us with our test. This also helps gathering the amount of participants that we need, which allows us to understand how a small population reacts to the dynamic system.

10.2.3.3 Online Test

The prize of this test is the entrance into a multitude of giveaways created and managed on the site SteamGifts[8]. The participants will receive the link to these giveaways upon submission of the survey. The participants used a download link to get a build version of the game. The first time a player entered the game a random choice was made by the system which either put them in the dynamic group or the static group [9.2 Data Gathering].

10.2.3.4 Face to Face Test

The prize of this test is entrance to a separate giveaway containing a headset for first place and two different games for second and third place.

The distribution of this test has been by hanging up flyers around Aalborg University Copenhagen, asking people to sign up using the giveaway as incentive. Additionally, each of the us have asked friends to help out, which have also been a big help to increase the amount of participants that have been gathered. Due to the long nature of this test ranging from 45 minutes – 2 hours and 20 minutes, it was difficult to gather participants. Distributing which group each participant belonged to was done using manual division.

This test gathers its data in an in-depth interview format, where it is possible for the participant to give more depth to any answer, and also allow for the interviewer to obtain more in-depth knowledge about the answers the participant gave by digging in to areas.

[8] https://www.steamgifts.com/ is a website specifically devoted to handling giveaways. Anyone can open a giveaway offering their steam games as reward.

11 Evaluation

Throughout this section, the way we process the data will be explained. Furthermore, we will also go over the different interest points which will be used for data analysis. We will also go over problems we discovered during the test. Lastly, we will present the data we have gathered, and our analysis of the data. The data which will be used for the data analysis is only from the online test to avoid having a set of participants who have been treated differently to influence the results. The face to face interviews are used for observation, both during game play and while answering the survey questions. The face to face interview will throughout this section only be used to explain either problems, or be used as a reference if the data seems unclear.

11.1 Methods

The Kolmogorov Smirnoff test [34] is used to determine whether the data can be considered to be from a normal distribution. Many of the questions have been converted to either correct or incorrect, meaning we will deal with binary data as well as either parametric and non-parametric data. Depending on the type of data, and what the normality shows, these different approaches will be used:

11.1.1 Quantitative

- **Parametric**
  - The two-sample t-test [35] will be used to determine whether or not the two samples are significantly different from one another.
- **Non-Parametric**
  - The two sample Kolmogorov Smirnoff test [36] will be used to determine whether the data can be considered to come from the same distribution.
  - The Wilcoxon rank sum test [37], will be used to determine whether two data sets are significantly different.
- **Binary data**
  - The Fisher's exact test [38], will be used to determine if there are differences in the correct and incorrect answers between the two groups.

We believe that by using these in conjunction with each other it is possible to understand the implications of the data. It will also allow us to see if the dynamic system has had an effect on the treatment group versus the control group.

11.1.2 Qualitative

The qualitative data which covers points of interests will be converted by hand into more processable data. This can either be binary data or a different representation of the data depending on what is being investigated. This allows us to compare the qualitative and the quantitative data from the survey.

11.2 Points of Interest

During this section, the different areas of knowledge that will be used to both compare the groups between each other, and the specific areas which are of interest in order to prove or disprove the null hypothesis will be explained.

The main comparative points which have been investigated during this test, is the relationship between the unit types, and the special abilities of the shadow unit. However, simply comparing these two areas could introduce a bias consisting of one group being generally better at understanding the mechanics. Therefore, it is of interest to understand if there are inconsistencies between the groups in knowledge about all of the game mechanics being analyzed.

Additionally, we also believe the way the different groups answered the general understanding questions, might reveal a tendency, therefore these will also be investigated.

This leads us to analyzing these sets of data:

- **Units you bring to battle**
- **Win condition**
- **Summon points**
- **Attacking**
- **Retaliation**
- **Type relationship**
- **Special units**
- **General understanding of mechanics** – Self assessment
- **'Does not understand mechanic' event –** Player data
- **'Information trigger' event –** Player data

We believe that by analyzing this data we will be able to gather an understanding of how well the dynamic system worked, and what influences the dynamic system has had on the players' knowledge.

11.3 Problems and Bias

This section will cover various biases and problems that occurred in the test process from the final iteration of the test.

11.3.1 Distribution Problem

After posting the giveaway online and keeping an eye on the participants we received, we realized after 33 participants, that the random distribution of the two different groups was skewered towards the static with 24 participants versus 9 in the dynamic. The expected amount of participants we hoped to gather was around 50. Therefore, if this tendency would continue we would end up with a very large difference between the two groups. This made us force the dynamic version game onto the participants, by uploading a version which could only be dynamic. This reduced the skewer greatly, and after getting the number of participants in each of the two groups within a ±2, we uploaded the original version of the game again.

11.3.2 Survey misunderstandings

A different problem which was discovered during the face to face test was some of the questions in the survey had a tendency to be misunderstood, or was poorly designed which made it difficult for the participants to answer.

These questions will still be included in the data analysis, as it is not completely certain if the tendency observed in the face to face test also occurs in the online survey. If any of the questions fall into a specific category, either a difference between the two groups, the amount of correct answers falls below a certain threshold, or if they were about one of the complex mechanics used in the dynamic tutorial system. If any question falls into either of these categories, it merits further investigation. If one of these questions seem to have a problem, they will be explained during that section.

11.4 THE RESULTS

The total amount of responses we received on the survey summed up to 86. However, due to the distribution format, we expected to receive participants who was not interested in giving helpful answers. Therefore, these responses had to be cleaned. The way this is done is by looking through the answers manually. If any of the following criteria were fulfilled the participant was removed and not used for data analysis:

- The ID not following the convention.
- The ID appeared more than once (in this case the first appearance of the ID was used)
- Any participant younger than 13
- Any participant who had tried the game before
- Any participant who did not consider himself a gamer
- Multiple obvious text answers that has nothing to do with the associated question
  - One example of this was a participant who filled out most of the text answers with the number 15.

This lead us to a total number of participants of 73 responses, of which 41 had the dynamic version of the game, the treatment group, and 32 had the static version of the game, the control group.

The data gathered directly through the game was only of interest to us if it shared an ID with a valid data entry of the survey. Therefore, only player data entries which contained IDs which had an equivalent in the survey was used. If two instances of the IDs existed the first entry with that ID instance was used.

All of the statistical analyzes are done using an α-value of 0.05.

11.4.1 Participant Self-Assessment

The participants answered two questions regarding how they felt about their knowledge of the game mechanics. These two questions were "*Do you feel you have a general understanding of how the different mechanics works?*" and "*Do you feel the tutorial has prepared you to progress further in this game?*". To see if the dynamic tutorial has had an impact on their own perception on their knowledge of the game, these two are tested.

Do you feel you have a general understanding of how the different mechanics works?

Test	h	p
kstest static	**1**	**8.7956e-05**
kstest dynamic	**1**	**7.0598e-08**
kstest2	**0**	**0.98**
ranksum	**0**	**0.31**

Table 1 – Tests for the question "Do you feel you have a general understanding of how the different mechanics works?"

Table 1 shows that there are no differences between the two groups, meaning the dynamic tutorial has not influenced the way the participants feel their knowledge is about the game.

Do you feel the tutorial has prepared you to progress further in this game?

Test	h	p
kstest static	**1**	**0.0054**
kstest dynamic	**1**	**1.2494e-05**
kstest2	**0**	**0.96**
ranksum	**0**	**0.24**

Table 2 – Tests for the question "Do you feel the tutorial has prepared you to progress further in this game?"

Table 2 shows the same as Table 1, that the dynamic tutorial has not had an influence about how the participants assessed their understanding of the game mechanics. There are no differences between the two groups.

We will now investigate the actual knowledge of the participants, through the questions which they had to answer during the survey about the game.

11.4.2 Participant Knowledge

To understand if there is a difference in the knowledge of a participant between the groups, we tested both correct vs incorrect answers on each question, and overall amount of correct answers between the two groups.

Test	h	p
kstest static	0	0.1786
kstest dynamic	0	0.3338
Two sample t-test	0	0.6540

Table 3 – Normality test and between group sum of correct answers test

Table 3 shows that the data can be considered to be normally distributed, and therefore our parametric test is run, there is no significant difference between the sum of correct answers between the two groups, which does not reject the hypothesis.

Next, we will look into the specific questions. If any of the questions shows either a difference between the two groups, or falls below a correct answer percentage of 80%, we will investigate them further to understand why this might happen. We believe that any question where more than one out of five participants had a problem answering the question correctly merits analysis to understand why this happens.

Any question that merits further investigation will be written as text when they are covered.

Identifier	h	p	Identifier	h	p
10	0	1	25	0	0,82
11	0	1	26	0	0,77
12	0	0,81	27	0	0,49
13	0	1	29	0	0,50
14	0	1	30	0	0,39
15	0	1	31	0	1
16	0	1	32	0	0,58
17	0	0,22	34	0	0,24
18	0	0,50	35	0	0,77
19	0	1	36	0	0,80
20	0	0,33	37	0	1
22	0	0,65	38	0	0,23

Table 4 – Fisher's exact test results for questions – the identifier denotes the number of the question in [16.3 Appendix 3 – Final Questionnaire]

Table 4, shows a comparison of all of the questions used for the sum in Table 3, between the two groups. All of the p-values are above the α-value, therefore we consider all of the questions to be answered equally right or wrong in each of the two groups.

We believe however that the participants had a harder time answering some compared to others. We therefore look at the mean value of the binary answers for each question.

Identifier	mean	Identifier	mean
10	0,97	**25**	0,58
11	0,67	**26**	0,81
12	0,63	**27**	0,88
13	1,00	**29**	0,97
14	1,00	**30**	0,78
15	0,93	**31**	0,66
16	0,96	**32**	0,96
17	0,92	**34**	0,53
18	0,97	**35**	0,81
19	0,97	**36**	0,30
20	0,85	**37**	0,90
22	0,93	**38**	0,90

Table 5 – The mean value of the binary data for the different questions the identifier denotes the number of the question in [16.3 Appendix 3 – Final Questionnaire]

By looking at the Table 5 it is clear that some of the questions have less correct answers than others. We are interested in anything that has a lower percentage of correct answers than 80%, meaning anything lower than a mean of 0.8 are of interest to us. These will therefore be further investigated. This is to understand which questions the participants had the most difficulty with.

In addition to this, other questions might be hand-picked for further analysis as well if they have a direct relation to the two main complex mechanics being investigated.

11.4.2.1 The Questions with a correct answer percentage lower than 80%

Question 11 (67% correct) and 12 (63% correct) are "*Is there a limit to how many units you can take to battle?*" and then the follow-up question "*How many units can you bring to battle?*" respectively. We discovered through the qualitative face to face interviews that this question seemed to be misunderstood by multiple participants (3 out of 10). They thought the question asked about the total amount of units which could be spawned during a combat (there is no limit as you can keep spawning any unit you have chosen to bring to the battle). When the participants who misunderstood this in the face to face then was asked the follow up question, how many they could bring they answered this correctly, namely 4.

We therefore believe that this question had a tendency to be misunderstood in the survey as well as the face to face interviews.

Investigating the amount of people who received the follow-up question and who answered correctly was 94%, which is a quite high correct answer percentage. We believe that this is an indication that the initial question had a tendency to be misunderstood.

Question 25 (58% correct) is *"When do you gain summon points"* was a text answer. The correct answer contained three elements, killing an enemy tower, killing an enemy unit, and at the start of each turn. As it is a three-part question, and there are three events that grant summon points we found it interesting to see which events were not obvious to all players.

Mentions:	All three	Only killing	Only turn	Tower and kill	Turn and kill
Percentage of answers	58%	6%	6%	6%	24%

Table 6 – Percentage of participants that mentioned the different ways to obtain summon points
Separate ways which had no mentions are not shown

We see in Table 6, that tower seems to be the one event that they are less likely to mention, with only 6% mentioning tower and kill, and the other 36% omitting the tower completely from their answer. We theorize that this could be because of the frequency of gaining summon points from the different events. Killing a unit and gaining the turn is something that happens often as compared to killing a tower, as only one can fall during a combat that will impact the combat by granting summon points, as when the second tower is killed the combat is over.

Question 30, 31, 34, 36

All of these four questions relate to either specific use of the shadow unit, the special ability of the shadow unit, or how the shadow unit can move on the combat map.

- **Question 30**
  What unit(s) were special?
- **Question 31**
  What made it/them special?
- **Question 34**
  During the shadow units' next turn, it can (Paired with an image)
- **Question 36**
  What can the shadow unit do on its next turn? (Paired with an image)

Question 36 had a correct answer percent of 30%, when investigating a possible reason for this low percentage, we realized that the wording was imprecise, and none of the available answers were completely correct. We therefore omit this question from our analysis of the knowledge about shadow units.

11.4.3 Differences in the Knowledge About Shadow Units

To test the differences in the participants' knowledge about shadow units, we will investigate the remaining three questions regarding the mechanics of the shadow units.

The binary data for the three questions were added together, to be able to gain an overall understanding of how well the participants understood the shadow unit's mechanics within the two groups.

Test	h	p
kstest static	0	0.0601
kstest dynamic	1	0.0371
kstest2	0	0.9342
Ranksum	0	0.2916

Table 7 – Tests for sum of correct answers to shadow related questions

As can be seen in Table 7 there are no significant differences between the two test groups, which means that they answered the questions in a similar fashion, and not one of the groups had a better understanding of the shadow units. However, as this is one of the core mechanics we are testing we are interested in seeing if there are slight differences between the groups, even if they are not statistically different.

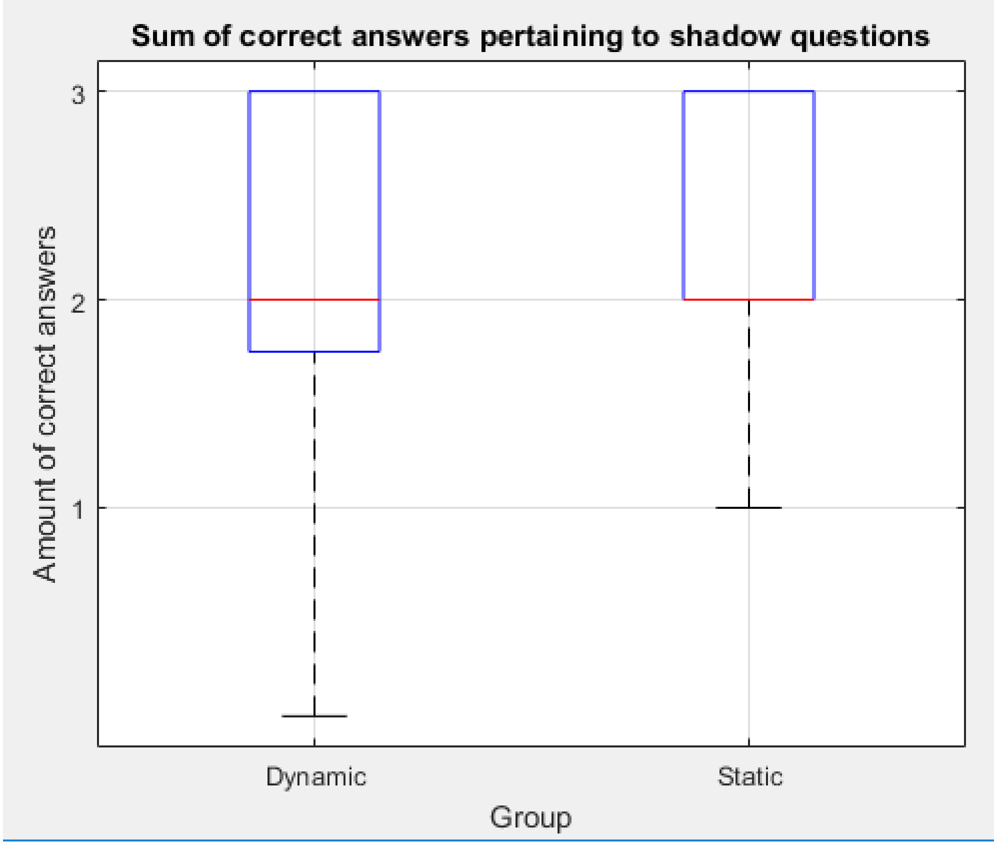


Figure 24 – Boxplot of the sum of correct answers for shadow questions compared between the two groups

Figure 24 shows us that the there is an indication of a small difference between the groups. It seems here that the dynamic group has been worse at answering the questions pertaining to shadow units compared to the static counterpart. As this is an interesting tendency we wanted to see if there are any specific questions the dynamic group seemed to be worse at answering compared to the static group.

Test	Question 30	Question 31	Question 34
Dynamic	0.88	0.66	0.46
Static	0.94	0.66	0.63

Table 8 – Mean of correct answers for the two groups for the shadow questions

Question 30 *"What unit(s) were special?"* required a participant to select both special units, therefore the answer was not considered correct in Table 5 unless both units were selected. However, as we are investigating the shadow units, we are more interested in how many people understood that shadow was special as compared to selecting both special units.

Table 8 shows that the dynamic group seemed to have a harder time selecting all the correct possibilities for the first image question, which asked the participant to check all of the things the shadow unit would be able to do on its next turn. However, we have also seen that there are no significant differences between their answers.

11.4.4 Differences in Unit Type Knowledge

Testing the differences in the knowledge of the unit types and how they relate to one another, we have summed up the correct answers for each of the questions relating directly to the relationship between the units.

Test	h	p
kstest static	1	3.1357e-07
kstest dynamic	1	5.0407e-07
Kstest2	0	0.8246
ranksum	0	0.1319

Table 9 – Tests for sum of correct answers to unit type relationships related questions

These test in Table 9 reveals that there are no differences between the two groups when it comes to the amount of correct answers made for the unit type relationships.

Visualizing this data however does reveal a tendency.

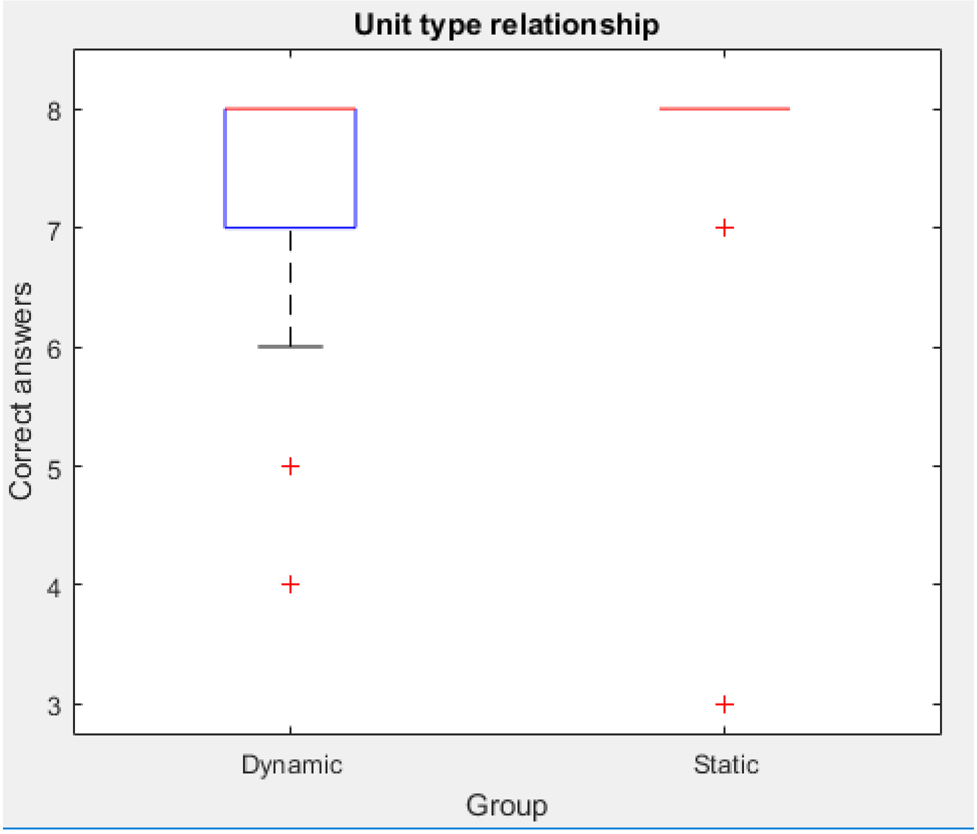


Figure 25 – Boxplot of correct answers for unit type relationship questions

The tendency seen in Figure 25 seems to favor the static group, which looks to have only a few outliers that were not able to answer correctly on all eight questions relating to these questions. Therefore, we find it interesting to understand which questions relating to the unit type relationship the dynamic group found most difficult.

	Fire -	Fire +	Lightning -	Lightning +	Water -	Water +	Nature -	Nature +
Sum of correct answers	41	41	38	39	36	39	40	33
Percentage correct answers	100%	100%	92,7%	95,1%	87,8%	95,1%	97,6%	80,5%

Table 10 – Sum of correct answers and the percentage of correct answers it represents for the dynamic test group
The + denotes what the type is strong against, the – denotes what it is weak against

Looking at Table 10 we can see that the people who played the dynamic version did not completely know what water is weak against and what nature is strong against.

Possible reason for these fluctuations will be discussed in the [12.5.3 Cross-game Similarities Can Confuse Players] chapter.

11.4.5 Player Data Analysis

In order for us to understand the implications of the dynamic tutorial, both how it functioned and if it has had an influence on the data, we decided to find out if there were any differences between the two groups, for the dynamic events.

The amount of times a participant triggers the 'did not understand mechanic' event can be skewered. This is due to the nature of the dynamic version of the game versus the static version of the game. When the system determines that the player has not learned a mechanic in the dynamic version of the game, it forces the player to complete the same combat again, whereas the dynamic system has no influence on the tutorial in the static version of the game.

kstest	p	h	kstest	p	h
Dynamic Shadow information trigger	7,52E-11	1,00	Static Shadow information trigger	4,62E-07	1,00
Dynamic Type information trigger	1,56E-08	1,00	Static Type information trigger	5,79E-07	1,00
Dynamic Shadow did not understand mechanic	5,49E-09	1,00	Static Shadow did not understand mechanic	1,67E-07	1,00
Dynamic Type did not understand mechanic	3,03E-11	1,00	Static Type did not understand mechanic	Never occurred	

Table 11 – Normality test for all data point from player data

As can be seen in Table 11, none of the data sets contain data that can be considered to be normally distributed. Therefore, the non-parametric tests are used. The dynamic version of the game does not show information about the two mechanics being investigated. We therefore assume that the participants will not be able to completely understand this mechanic, and therefore will trigger more dynamic events compared to the static who will receive all information initially.

Test	p	h	Test	p	h
kstest2 Shadow information trigger	1,00	0	kstest2 Shadow did not understand mechanic	1	0
ranksum Shadow information trigger	0,23	0	ranksum Shadow did not understand mechanic	0,91	0
kstest2 Type information trigger	1	0	kstest2 Type did not understand mechanic	1	0
ranksum Type information trigger	0,95	0	ranksum Type did not understand mechanic	0,22	0

Table 12 – Non-parametric test for comparing each data set between the groups

As Table 12 shows there are no differences between the two groups. As there are no differences it is of interest to find out why. Therefore, we look into the raw summarized data in order to see if there are any obvious reasons why there would be no differences.

Looking at Table 13, we can see that the numbers in general are very low considering the total number of participants is 73. Therefore, it would be hard to find anything meaningful within this data, due to the small impact the dynamic system seems to have had on the participants.

	Showed or would have showed shadow popup	Showed or would have showed type pop up	Times not learned shadow was triggered	Times not learned type was triggered
Sum	14	18	26	2
Max for one participant	3 (static)	3 (static)	6 (dynamic)	1 (dynamic)
Sum dynamic	4	10	15	2
Sum static	10	8	11	0

Table 13 – Raw player data numbers

Throughout this section, we have presented the data we found most important to help make a decision and conclude on what this test has shown. Throughout the next section the data will be discussed in regard to what the data tells us. For some of the data this will be done in conjunction with the qualitative results, and for other data it will be used as standalone results.

12 DISCUSSION

This chapter will examine the results gathered from both the online test, but also the face to face test, trying to see which parts of the system worked, and which could be improved.
The online test consisted of 73 participants after being cleaned and the face-to-face test had 10 participants. This is not an amount which is able to represent a larger population; however, we believe it is a large enough group to listen to in order to learn what should be changed. However, gathering a consensus about what was good and what was bad was difficult, due to the fact that the test participants had wildly different opinions on the tutorial in the test. Some said it was excessively long, others said it had a perfect length. A couple of people stated that it was far too difficult, whereas some found it to have a very fitting difficulty. However, combining the data gathered from the online test, our observation of the face to face, and the subsequent interviews, did result in several points worthy of discussion.

12.1 TEXT BOXES

From [3.2 Creating a Good Tutorial], almost all experts said that large amounts of text should be avoided, and having this would result in a suboptimal tutorial. However, after making a text box only tutorial for the first iteration of the test [8.1 Tutorial Design], we see that the participants did not mind having information given to them through text boxes, it was decided to continue using that presentation form.

Very few people in the final test mentioned that they did not like the text boxes. Whereas, an equal amount said that they liked the way the information was presented. This means that, at least for this particular game, using text did not seem as bad as it was laid out to be from the experts.
It should be noted that a good amount of people also said they would have liked some visual representations along with the text, e.g. arrows pointing to what the text was referring to, or an image showcasing a scenario. Having visual representations could help them understand the text.

12.2 MAKE SURE THE INFORMATION IS RELEVANT

Throughout our observations, we noticed a trend where some players would follow the tutorial along in the start of the game, but after realizing that they knew, or easily could have guessed most of the initial information, they sometimes quickly skipped over future information. This could be because they might have found it to be information that was not relevant for them. It should be noted that no one stated this as a reason for skipping any information. This observation suggests that providing the player with information he deems irrelevant increases the likelihood that he will skip relevant information at a later stage, thinking it also is irrelevant.

Ensuring relevance in the information will also help reduce the amount of 'noise pollution' in the information, which could have a negative impact on the player, as presented in [3.2.3 George Fan], as it would provide fewer messages, but with more significant content.

12.3 INFORMATION SHOULD COME AT THE CORRECT TIME

One thing many people agreed on, was that presenting information shortly before it could be used, was a good thing. That way they could try what they had just learned immediately. This point backs up what many experts believed [3.2 Creating a Good Tutorial], that the tutorial should not be frontloaded, the information should be given just as it is needed, not before.

This leads into a huge issue with the dynamic tutorial system that was created. The system needs time to know if it should inform the player about a mechanic. As the system is reactionary to what the player is doing, the information it gives might come too late. A participant wrote as a comment in the survey, that he was informed about the relationship between the different unit types a few rounds before he lost a combat. The information came too late for him to be able to use the information. A few others also noted that the information appeared at inconvenient times. Receiving information when it is too late to use it, “back loading”, seems to be very discouraging for the player. This issue really needs to be addressed before moving forward with the system. Adapting the system, to function as a reminder system, instead of a standalone tutorial, might be a better solution, and would solve the back loading issue.

12.4 Displaying Messages Once

A major complaint we saw on numerous occasions regarding the fixed information that is present in both the dynamic and the static versions of the game, was that the information was only displayed once, and was not accessible later on. The main issues we saw was that some players accidentally skipped information on instinct, or simply didn’t want to read it. In some cases, this led to the player immediately regretting skipping the information and a worry that they had now missed some valuable information.

While this does speak positively towards a dynamic system, where the game will display the information again if it calculates a need for it, many players noted that they would like to access the information on their own accord, when they feel the need. This is worth taking into consideration when looking at a dynamic system, especially when keeping in mind that players have a feeling of a “correct” time to receive information, as discussed in the previous point.

A common suggestion was the ability to access the information through a menu, which gives the player freedom to choose for themselves. From the comments, we saw that this does seem to pertain more to complex mechanics, rather than simple features, such as controls.

12.5 Problems Using Complex Mechanics

One of the biggest takeaways from the test sessions, is regarding implementing the dynamic system on the complex mechanics, as was chosen in the beginning of the project, rather than on the simple mechanics of the game.

12.5.1 Players Need Confirmation in Their Assumptions

In the dynamic system, the information regarding the two complex mechanics were withheld from the player until the system deemed it necessary to display. While many of the players did in fact learn or at least have an idea about how the mechanics worked, their comments showed that they had a nagging insecurity that they may not understand it completely after all.

From our observations, we saw that the players who did receive the tutorial message from the system sighed in relief when they were finally told that their suspicions had been correct all along, and commented in interviews that it felt like a burden being lifted from their mind. One player mentioned in his interview that he was actually using a lot of focus in the game trying to figure out if he was right or wrong, by experimenting with his actions, rather than playing the game as he otherwise would. This both ties into the point of withholding information and displaying information at the right time, but also speaks against implementing a dynamic system on complex mechanics. It is harder for the player to get feedback that confirms or oppose their assumptions about the game when it comes to complex mechanics, rather than simple mechanics. In many cases the player can get immediate verification by seeing how the game reacts on the simple features such as basic controls, e.g. clicking the mouse to move the player – there is much less room for interpretation about how that works.

We believe that repurposing the system into a reminder system, instead of a tutorial system could work better. The reason for this is, that no information would be withheld anymore, which removes this element of frustration, while it still is able to help players who have trouble grasping a mechanic fully, to obtain a complete understanding of the specific mechanic, or has simply forgotten how it works.

12.5.2 Tactics are Hard to Distinguish

While this point may be more prevalent in the turn-based strategy genre than some others, it is still a point of contention against a dynamic system on complex mechanics. Both when designing the system and in our observations, we saw that when gathering and analyzing data about the players it is harder to determine what is actual misuse or misunderstanding of a mechanic, and what is an intended tactic from a player who fully understands the game. An observed example of this is a player who had summoned a number of units who were weak against what the enemy had on board, but he also had a clear route towards the enemy tower in the bottom lane, so he rushed the bottom lane with a couple of his units, while using the weak units to block the path of the enemy so that they couldn't interfere. While the system would qualify this as 'wrong', it is in fact a completely valid tactics to use in order to win the battle.

Looking at a comparison between using complex and using simple mechanics, the simple features would be much clearer for the system to interpret than complex, as the correct/wrong usage is easier to identify, and in turn makes the data easier to analyze.

12.5.3 Cross-game Similarities Can Confuse Players

One often occurring situation we noticed during the test and in the data analysis, was that players will draw assumptions based on their experience with other games, even if evidence of this assumption has not been presented in our game. A good portion of the players who answered wrong about type relationships had answered based on the system featured in the Pokémon games, where grass types (called nature types in our game) are strong against water types as seen in Table 10. While no information was presented to them which told them otherwise, they would rely on their instincts and experience from that game, and answer based on that.

This leads us to believe that complex information, especially if it deviates from other usages of similar systems, does need to be explicitly explained in some form and should not be withheld.

12.6 The System

The data presented in Table 14, shows the amount of times the dynamic system deemed that the player needed additional information, and how many times it deemed the player did not understand a mechanic.

	Showed or would have showed shadow popup	Showed or would have showed type pop up	Times not learned shadow was triggered	Times not learned type was triggered
Sum	14	18	26	2
Max for one participant	3 (static)	3 (static)	6 (dynamic)	1 (dynamic)
Sum dynamic	4	10	15	2
Sum static	10	8	11	0

Table 14 – This table is also shown in the evaluation chapter

The total sum of times that either group has triggered one of the events the dynamic system is reacting on, seems low comparing them to the number of participants that took part in the test. Going over each of the categories, there is not much of a difference between the groups of how many times the events triggered. However, we expected that the dynamic group would trigger a lot more events compared to the static. This was not the case. We believe that this can mean two things:

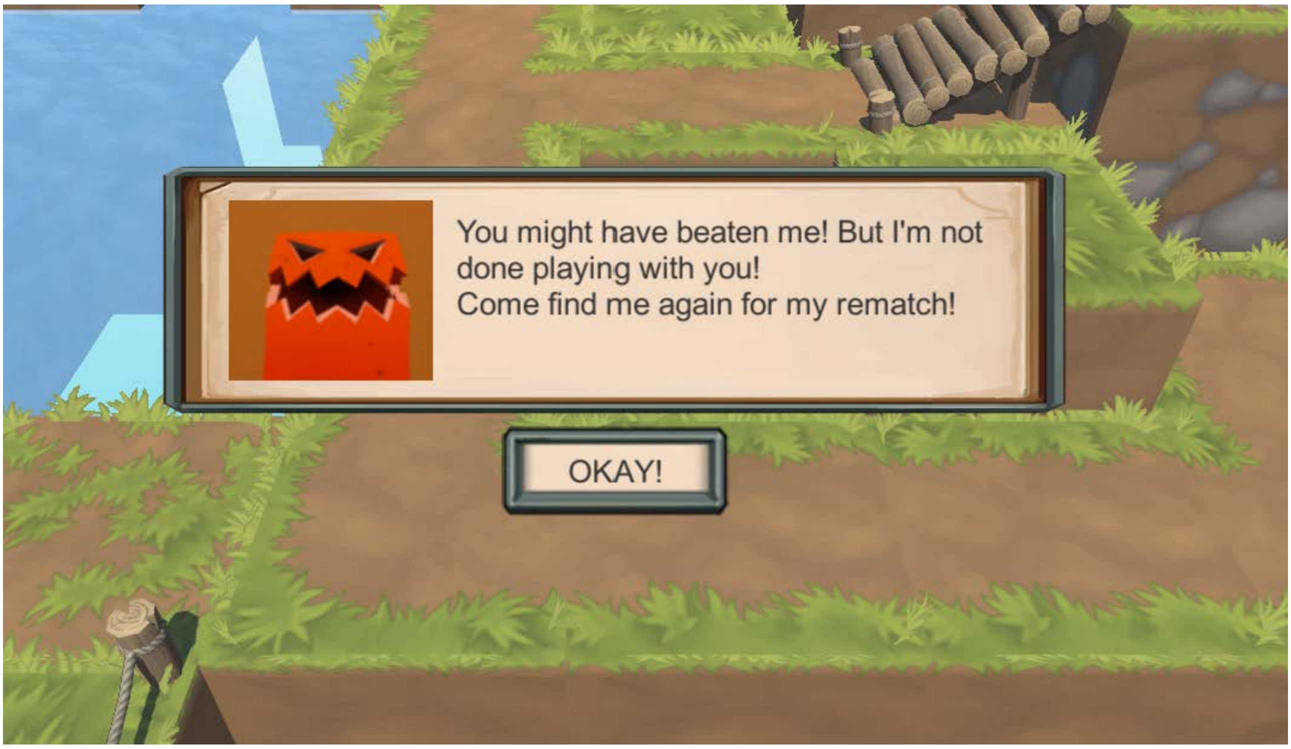
- **The way the system was implemented does not function as expected**
- **The problem lies with the game and not the system itself**

12.6.1 Innate Problem of the System

The dynamic system was imagined to assist the player and provide the same amount of knowledge as compared to a static tutorial. As we have seen throughout the presentation of data there has not been a statistical difference in the knowledge level of the different mechanics between the control and treatment group. However, by looking at the data, we do see that the knowledge of the dynamic group does not seem as good as the static group.

We therefore believe that the system did not function as imagined it would have, meaning it has not provided the dynamic group with as much information as they needed, and this is the reason why we see the small difference. This can therefore indicate that the system is not able to determine whether a participant has a full understanding of a specific mechanic or not.

An interesting number to look at in Table 14, is the maximum number of the times a player triggered the 'does not understand mechanic' event for the shadow unit. The fact that he triggered this event six times we can deduct that he has beat this combat seven times before being able to move on to the next island. Additionally, this participant did not trigger one single shadow 'information trigger' event. This presents a flaw in the way the system currently works. When a participant triggers a 'does not understand mechanic' event, he is not told why this happens or what he should do to in order to beat it next time. The message that the player receives is seen in Figure 26. If a 'does not understand mechanic' is triggered the enemy simply does not unlock the things it does when it is completed as intended.
This specific incident has occurred because the player did not utilize shadow, and was therefore not able to learn this specific mechanic, which the system did recognize. It did however keep the player in the dark about what to do next. We believe this to be bad practice.



You might have beaten me! But I'm not done playing with you!
Come find me again for my rematch!

Figure 26 - The message the player receives when they triggered 'does not understand mechanic' event

The other problem which also could have an influence on the results of this test is that the game mechanics, which has been the focus of the test, are not as complex as imagined from the beginning. This could also explain why the results have turned out as they have.

12.6.2 The Game Mechanics

The data presented during the [11 Evaluation] chapter compared with the numbers seen in Table 14 shows that there are no real differences between the control and treatment group. Apart from the reason presented above, it could also have something to do with the mechanics we are trying to teach the players, is something that they can find out themselves, without necessarily needing any information from the game, for them to utilize it properly.

One thing that could explain this is, the similarity the game mechanics has with other games. Therefore, the players already know something similar to what is presented to them throughout this game. They already have this information of how the mechanics function and also understand how to best utilize these mechanics in their favor.
The two different complex mechanics we used were a unit which was able to move through other units, a mechanic utilized in many other games. The other a type based strength/weakness system which is also present in many other games, as also described previously.

These reasons could have an effect on the knowledge of the player, and we do not know which is more prominent.

12.7 Strategic Element

Because the game revolves around unit usage, it might not be enough to only look at simple data points (Amount of strong hits, special ability used during combat) in order to gain a completely coherent understanding of the players' knowledge regarding the complex mechanics. Due to this there is an inherent form of strategy and tactic involved [6.2 The Game Design] e.g. how the units are moved up the field, which units to summon, where to block the path of the enemy and so on. This leads us to the problem:

The game revolves around strategic gameplay

The fact that the game revolves around strategy makes it difficult to decipher specific intentions with the user's actions. It would be necessary to gather a more overall understanding of how the player likes to play the game, how he uses the different units, which units he brings to what combats, etc. This reveals an issue for the dynamic tutorial implementation.

Understanding a player's strategy in a game is difficult and complex

The problem with understanding strategy is that it differs from player to player. It is therefore necessary to obtain a very large amount of data about how the game is being played. This requires many upon many of hours played by many different players. Additionally, as this system is focused around the tutorial of the game, it would be necessary to substitute many hours of gameplay, by a vast amount of participants to observe and hopefully gather all strategies, both the very viable ones and the not as viable ones.

13 CONCLUSION

The goal of this project was to create a dynamic tutorial system, which would allow players of different experience levels to go through the tutorial section of a game and receive different amounts of information, but achieving the same level of knowledge about the game mechanics in the end. The system should be able to determine what information each individual player needs, and provide it. This led to the null hypothesis:

Players who have played the dynamic tutorial has the same, or more knowledge of the game compared to the players who have played the static tutorial.

Throughout the evaluation and discussion chapters we saw that the data showed no differences in knowledge level between the two groups. This could indicate that the system has provided enough information for the player to reach the end of the tutorial with equal levels of knowledge. However, based on further analysis of our data and the additional points mentioned in the discussion, we do not believe that this can be accredited to the system.

We perceive the biggest takeaways from the results are a problem with the mechanic complexity, and how withholding information affected the players' experience.

Firstly, we learned that players, even if they feel as though they have learned a component of the game, still need certainty in order to alleviate their insecurities. This meant that since the system focused on complex mechanics, the players were unsure about how to properly utilize them in their favor, which lead to frustration, confusion, and worrying. We believe that this system might be better suited for simple mechanics, such as basic controls, and that it could be worth looking into in the future, as we see the developers of Minecraft are currently working on a seemingly similar system [4.1 Minecraft].

Secondly, the timing of the information also revealed an issue with the system, where the player would receive information at an inopportune time, e.g. close to the end of a battle. This meant that it both felt useless to them, and caused frustration if the information was something they could have used. We see an opportunity to utilize this within the context of a reminder system instead, where the dynamic element is used to determine whether the player needs a reminder of possibly forgotten information. While the result from the test of the system did not achieve our initial goal, we did make these unintended discoveries:

- **Withholding information is bad**
  We discovered throughout the test that withholding information caused the players to use their attention on figuring out this mechanic, as opposed to using their attention on other areas within the game.
- **Using text box as a tutorial type is acceptable if used correctly**
  What has been presented in [3.2 Creating a Good Tutorial] about text boxes were not observed. The participants did not have a problem with the way the tutorial was implemented. We believe that by keeping the information very concise and making sure it is presented at the correct times, it reduced the problems experts have observed through their research.
- **Previous knowledge from experience has a lot to say when learning a game**
  It is important when using mechanics which are similar to mechanics from other games, but not completely the same, to clearly inform the player that this mechanic is not the same, otherwise they will use their previous knowledge to fill in blanks which might be missing in the explanation of the new mechanic.

14 Future Work

During the [12 Discussion] we have presented alternative ways to reach the presented goal for this project. In this section, we will describe the two previously mentioned methods.

One of the main reason why we believe that the current dynamic tutorial system does not provide a good method of reaching the goal of this project, is that withholding information about a complex game mechanic has a negative influence on how the player experiences the game, and how they play it.

14.1 Go Simple

One of the alternative methods described earlier is focusing on simpler mechanics. The reason for this is two-fold, one the analysis of a players understanding will be less complex, and two, a gamer will easier be able to transfer their knowledge from previous games to this game.

We believe that if a player is able to utilize knowledge from previous games for mechanics that will most likely be similar, e.g. click to move, or space to jump [6.2 The Game Design] also reduce the amount of information noise presented to the player which could allow for the player to use more focus on learning other mechanics. Furthermore, we believe that withholding information about simpler mechanics' will have less negative impact on the player's game experience. Due to the fact that players' will be able to transfer knowledge from previous games into this game, and that simpler mechanics do not usually have a different mechanical effect on the game other than maneuvering or acquiring units.

The other part of the reason is simplification of data analysis points. We believe that it is possible to define more specific and simple player data points to base the analysis on. Simple game mechanics rarely involve utilizing a strategy in order to understand them completely. They are part of strategies but on a very subtle level as in to follow a strategy you will have to perform the mechanical actions of the simple mechanics.

Therefore, we believe looking at simple mechanics will work better, as it is possible to create a better generalization of when a player does not understand a mechanic, thereby reducing the implementation effort, and withholding information not having a big impact on the player's game experience.

14.2 Reminder System

Instead of looking at simple mechanics, the system could be altered to function as a reminder system instead. Creating a reminder system would mean that the player would receive all the information that he would have received in the static version of the game, but whenever the system believed that he might not have fully grasped a concept, it could remind him about that mechanic. For instance, if the player never moves a shadow unit through other units, the system could say 'remember that shadow units can move through other units'.

Going this direction with the system is not without issues, it would alleviate the problems of back loading as described in [12.3 Information Should Come at the Correct Time], but it would not solve the inherent issues with the analysis of the gathered data. It will still be very hard to distinguish whether or not the player has not fully understood a mechanic, or that the player is using a strategy the system is not aware of. Such an implementation would also not have to be limited to the tutorial of the game, but could be running in the background throughout the entire game.

These two new possible directions to take the project, is the directions we believe is the most promising. However, it is very important that the discussion points brought up in [12 Discussion], should be closely examined before continuing.

15 BIBLIOGRAPHY

[1] From Software, "Dark Souls 3," 2016. [Online]. Available: https://www.darksouls3.com/en/.

[2] Firaxis Games, "XCOM: Enemy Unknown," [Online]. Available: https://xcom.com/.

[3] Dictionary.com, "Tutorial," [Online]. Available: http://www.dictionary.com/browse/tutorial.

[4] P. Suddaby, "The Many Ways to Show the Player How It's Done With In-Game Tutorials," Envato Pty Ltd., 31 08 2012. [Online]. Available: https://gamedevelopment.tutsplus.com/tutorials/the-many-ways-to-show-the-player-how-its-done-with-in-game-tutorials--gamedev-400.

[5] Red Hook Studios, "Darkest Dungeon," [Online]. Available: http://www.darkestdungeon.com/.

[6] QCF Design, "Desktop Dungeons," [Online]. Available: http://www.desktopdungeons.net/.

[7] Team ICO, "The Last Guardian," [Online]. Available: https://www.playstation.com/en-us/games/the-last-guardian-ps4/.

[8] Nintendo, "The Legend of Zelda: Wind Waker," [Online]. Available: http://www.zelda.com/.

[9] Ubisoft, "Assassin's Creed," [Online]. Available: https://www.ubisoft.com/en-GB/game/assassins-creed/.

[10] Alientrap Games, "Apotheon," [Online]. Available: http://www.apotheongame.com/.

[11] E. Adams, "The Designer's Notebook: Eight Ways To Make a Bad Tutorial," UBM, 14 06 2011. [Online]. Available: http://www.gamasutra.com/view/feature/134774/the_designers_notebook_eight_.php.

[12] F. Merlin, "Chat Event - Developer Success Story Andy Moore," 31 08 2010. [Online]. Available: https://home.fgl.com/forums/view-thread?thread_id=20466.

[13] Extra Credits, "Tutorials 101 - How to Design a Good Game Tutorial - Extra Credits," Extra Credits, 16 03 2011. [Online]. Available: https://www.youtube.com/watch?v=BCPcn-Q5nKE.

[14] T. Curtis, "GDC 2012: 10 tutorial tips from Plants vs. Zombies creator George Fan," Gamasutra, 09 03 2012. [Online]. Available: http://www.gamasutra.com/view/news/165359/GDC_2012_10_tutorial_tips_from_Plants_vs_Zombies_creator_George_Fan.php.

[15] E. Andersen, E. O’Rourke, Y.-E. Liu, R. Snider, J. Lowdermilk, D. Truong, S. Cooper and Z. Popovic, "The Impact of Tutorials on Games of Varying Complexity," *CHI,* p. 10, 2012.

[16] Mojang Synergies AB, "Minecraft," Mojang Synergies AB, 2017. [Online]. Available: https://minecraft.net/en-us/.

[17] Electronic Arts Inc., "Plants vs. Zombies," Electronic Arts Inc., [Online]. Available: http://www.popcap.com/plants-vs-zombies-1.

[18] D. Charles, M. McNeil, M. McAllister, M. Black, A. Moore, K. Stringer, J. Kücklich and A. Kerr, "Player-Centred Game Design: Player Modelling and Adaptive Digital Games," in *Proceedings of DiGRA 2005 Conference: Changing Views – Worlds in Play.* , 2005.

[19] G. Yannakakis and J. Togelius, "Experience-Driven Procedural Content Generation," *IEEE Transactions on Affective Computing,* pp. 1-16, 2011.

[20] J. Togelius and N. Shaker, "The search-based approach," in *Procedural Content Generation in Games*, www.pcgbook.com, Springer, 2016, p. Chapter 2.

[21] J. Togelius, G. N. Yannakakis, K. O. Stanley and C. Browne, "Search-Based Procedural Content Generation: A Taxonomy and Survey," *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES,* pp. 172-186, 2011.

[22] M. J. Ponsen, H. Muñoz-Avila, P. Spronck and D. W. Aha, "Automatically Acquiring Domain Knowledge For Adaptive Game AI Using Evolutionary Learning," in *IAAI*, 2005.

[23] J. Togelius, R. D. Nardi and S. M. Lucas, "Making Racing Fun Through Player Modeling and Track Evolution," 2006.

[24] Steam, "Galactic Arms Race," 2017. [Online]. Available: http://store.steampowered.com/app/249610/?l=english.

[25] E. J. Hastings, R. K. Guha and K. O. Stanley, "Evolving Content in the Galactic Arms Race Video Game," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG09)*, Piscataway, NJ, 2009.

[26] Rampant Games, "Whadayamean, “Turn-Based?”," [Online]. Available: http://rampantgames.com/blog/?p=6310.

[27] Stainless Games Ltd., "Magic: Duels," [Online]. Available: http://magic.wizards.com/en/content/magic-duels.

[28] Ubisoft, "Heroes of Might and Magic 7," [Online]. Available: http://mmh7.ubi.com/en/blog.

[29] Firaxis Games, "Civilization," [Online]. Available: https://www.civilization.com/.

[30] Wikipedia, "Chess," [Online]. Available: https://en.wikipedia.org/wiki/Chess.

[31] The Pokemon Company, "Pokemon Video Games," [Online]. Available: http://www.pokemon.com/us/pokemon-video-games/.

[32] Wikipedia, "Stratego," [Online]. Available: https://en.wikipedia.org/wiki/Stratego.

[33] The Pokemon Company, "Pokemon Duel," [Online]. Available: http://www.pokemon.com/us/app/pokemon-duel/.

[34] Mathworks, "One-sample Kolmogorov-Smirnov test," 27 May 2017. [Online]. Available: https://se.mathworks.com/help/stats/kstest.html.

[35] Mathworks, "Two-sample t-test," 27 May 2017. [Online]. Available: https://se.mathworks.com/help/stats/ttest2.html.

[36] Mathworks, "Two-sample Kolmogorov-Smirnov test," 27 May 2017. [Online]. Available: https://se.mathworks.com/help/stats/kstest2.html.

[37] Mathworks, "Wilcoxon rank sum test," 27 May 2017. [Online]. Available: https://se.mathworks.com/help/stats/ranksum.html.

[38] Mathworks, "Fisher's exact test," 27 May 2017. [Online]. Available: https://se.mathworks.com/help/stats/fishertest.html.

[39] Armor Games Inc., "SteamBirds," [Online]. Available: http://armorgames.com/play/5426/steambirds.

[40] IGDA, "History of the IGDA," IGDA, [Online]. Available: http://www.igda.org/?page=history.

[41] Jens Bergensten, "Twitter," Mojang, 30 05 2017. [Online]. Available: https://twitter.com/jeb_/status/869477395916611584.

[42] N. Adams, "Twitter," Mojang, 19 04 2017. [Online]. Available: https://twitter.com/Dinnerbone/status/854597026042007556.

[43] Activision, "Call of Duty," [Online]. Available: https://www.callofduty.com/da/.

[44] Zynga, "FarmVille," [Online]. Available: https://www.zynga.com/games/farmville.

[45] UBM, "Game Developers Conference," [Online]. Available: http://www.gdconf.com/.

[46] DigiPen, "The Many Roles of James Portnow," DigiPen, 19 03 2013. [Online]. Available: https://news.digipen.edu/academics/the-many-roles-of-james-portnow/#.WS_x2WjyhhE.

16 Appendix

16.1 Appendix 1 – Games We Played

16.1.1 Games Specific to the Turn-Based Genre

- **Dragon Fin Soup**
  Spot test in the genre.
- **Costume Quest 2**
  Spot test in the genre.
- **Invisible Inc.**
  Good example of a specific tutorial type.
- **South Park: The Stick of Truth**
  Good example of a specific tutorial type.
- **XCOM: Enemy Unknown**
  A game we knew had a “bad” tutorial.
- **XCOM 2**
  A game we knew had a “bad” tutorial.
- **Desktop Dungeons**
  Had an interesting tutorial to look at.
- **Legend of Grimrock**
  Spot test in the genre.
- **Hack, Slash, Loot**
  Spot test in the genre.
- **Armello**
  Good example of a specific tutorial type.
- **Darkest Dungeon**
  Popular game in the genre.
- **Heroes of Might and Magic 7**
  Popular game in the genre.
- **Civilization 5**
  Popular game in the genre.
- **Persona 4**
  A game we knew had a “bad” tutorial.
- **The Elder Scrolls: Legends**
  Popular game in the genre.
- **Hex: Shards of Fate**
  Spot test in the genre.
- **Hearthstone**
  Popular game in the genre.
- **Child of Light**
  Spot test in the genre.

16.1.2 Games Outside the Turn-Based Genre

16.1.2.1 2D Sidescroller

- **Thomas Was Alone**
  Popular game in the genre.
- **Shovel Knight**
  Good example of a game lacking a tutorial.
- **Rogue Legacy**
  Popular game in the genre.
- **Guacamelee**
  Good example of a specific tutorial type.
- **Apotheon**
  Good example of a specific tutorial type.
- **Axiom Verge**
  Spot test in the genre.

16.1.2.2 Stealth

- **Hitman**
  Popular game in the genre.
- **Gunpoint**
  Spot test in the genre.
- **Hotline Miami**
  Popular game in the genre.
- **Volume**
  Good example of a specific tutorial type.
- **Assassin's Creed**
  Good example of a specific tutorial type.

16.1.2.3 Real-Time Strategy

- **Starcraft 2**
  Popular game in the genre.
- **Star Ruler**
  Spot test in the genre.
- **Anomaly Warzone**
  Good example of a specific tutorial type.
- **Faster Than Light**
  Popular game in the genre.

16.1.2.4 Shooters

- **Counter-Strike: Global Offensive**
  Popular game in the genre.
- **Sniper Elite 3**
  Spot test in the genre.
- **Tribes Ascend**
  Spot test in the genre.

- **Fallout 3**
  Popular game in the genre.

16.1.2.5 Action-Adventure

- **The Last Guardian**
  Good example of a specific tutorial type.
- **Dark Souls**
  Good example of a specific tutorial type.

16.2 Appendix 2 – First Questionnaire

Master

*Required

1. Do you feel you have a general understanding of how the different mechanics works? * Mark only one oval.

○ Yes, completely
○ To a certain degree
○ No, there are things I am still unsure about
○ No, I am unsure about almost all the interaction

2. Do you feel the tutorial has prepared you to progress further in this game? * Mark only one oval.

○ Yes, completely
○ To a certain degree
○ No, there are things I am still unsure about
○ No, I am unsure about almost all the interaction

3. Were there any elements that confused you? * Mark only one oval.

○ Yes
○ No

4. If there were, could you explain these briefly?

____

____

____

____

____

5. Is there a relationship between the different unit types? * Mark only one oval.

○ Yes   Skip to question 8.
○ No

6. Is there a limit to how many units you can take into battle? * Mark only one oval.

○ Yes Skip to question 7.

○ No Skip to question 9.

7. How many units can you take into a battle? *

Skip to question 9.

8. Please fill out the following about the type relationships * Mark only one oval per row.

	Fire	Lightning	Nature	Water
Nature is strong against	○	○	○	○
Fire is strong aginst	○	○	○	○
Lightning is strong against	○	○	○	○
Lightning is weak against	○	○	○	○
Fire is weak against	○	○	○	○
Water is strong against	○	○	○	○
Water is weak against	○	○	○	○
Nature is weak against	○	○	○	○

Skip to question 6.

9. Could you see why you had the units in battle, you had? *

10. What are the win conditions? *

11. What are the lose conditions? *

12. How does summoning work? *

13 When do you gain summon points? *

14. When is it possible to attack? *

15. When a unit is attacked under which conditions can this unit attack back? *

16. What do the icons above the units signify? *

17. Are there any units that have special abilities compared to others? * Mark only one oval.

○ Yes Skip to question 18.

○ No Skip to question 22.

18. What unit(s) were special? * Tick all that apply.

□ Shadow

□ Fire

□ Lightning

□ Nature

□ Stone

□ Water

19 What made it/them special? *

20. Is there a way to see stats of a unit? * Mark only one oval.

○ Yes Skip to question 21.

○ No Skip to question 22.

Skip to question 22.

21. Where can you see the stats of a unit? *

Skip to question 22.

22. Was the information intrusive to your play? * Mark only one oval.

○ Yes, very

○ Yes, somewhat

○ No, not really

○ No, not at all

23. Was the information understandable? * Mark only one oval.

○ Yes, completely

○ Yes, most of the time

○ No, it felt out of place some times

○ No, I did not understand any of it

24. Do you consider this to be a good tutorial? * Mark only one oval.

○ Yes, very

○ Yes, at times

○ Not really

○ No, not at all

25 What makes you like it/not like it? *

26. Is there any additional information you would have liked to have gotten through the tutorial? *

27. Did you have fun playing the game? *
Mark only one oval.

○ Yes, alot

○ Yes, at time

○ Not really ○ Not at all

28. What made the game fun/not fun? *

29. General comments

____

____

____

____

____

Skip to question 30.

Please just continue through this section without making any changes, the giveaway will be presented once you have pressed submit!

30. ID - Should self filled DONT TOUCH *

16.3 Appendix 3 – Final Questionnaire

Basic Information

*Required

1. Age *

____

2. Gender * Mark only one oval.

○ Male

○ Female

○ Other

3. Have you tried this game before? * Mark only one oval.

○ Yes

○ No

4. Do you consider yourself a gamer? * Mark only one oval.

○ Yes

○ No

5. Do you play turn based strategy games? * Mark only one oval.

○ Yes, all the time

○ Yes, sometimes

○ No, but have tried them

○ No, never

General Understanding

6. Do you feel you have a general understanding of how the different mechanics works? * Mark only one oval.

○ Yes, completely

○ To a certain degree

○ No, there are things I am still unsure about

○ No, I am unsure about almost all the interaction

7. Do you feel the tutorial has prepared you to progress further in this game? * Mark only one oval.

○ Yes, completely

○ To a certain degree

○ No, there are things I am still unsure about

○ No, I am unsure about almost all the interaction

8. Were there any elements that confused you? * Mark only one oval.

○ Yes

○ No

9. If there were, could you explain these briefly?

10. Is there a relationship between the different unit types? * Mark only one oval.

○ Yes Skip to question 13.

○ No

11. Is there a limit to how many units you can take into battle? * Mark only one oval.

○ Yes Skip to question 12.

○ No Skip to question 14.

12. How many units can you take into a battle? *

Skip to question 14.

13. Please fill out the following about the type relationships * Mark only one oval per row.

	Fire	Lightning	Nature	Water
Nature is strong against	○	○	○	○
Fire is strong aginst	○	○	○	○
Lightning is strong against	○	○	○	○
Lightning is weak against	○	○	○	○
Fire is weak against	○	○	○	○
Water is strong against	○	○	○	○
Water is weak against	○	○	○	○
Nature is weak against	○	○	○	○

Skip to question 11.

14. Could you see why you had the units in battle, you had? *

15. What are the win conditions? *

16. What are the lose conditions? *

17. How does summoning work? *

18. When do you gain summon points? *

19. When is it possible to attack? *

20. When a unit is attacked under which conditions can this unit attack back? *

21. What do the icons above the units signify? *

22. Are there any units that have special abilities compared to others? * Mark only one oval.

○ Yes Skip to question 23.

○ No Skip to question 25.

23. What unit(s) were special? * Tick all that apply.

□ Shadow

□ Fire

□ Lightning

□ Nature

□ Stone

□ Water

24. What made it/them special? *

25. Is there a way to see stats of a unit? * Mark only one oval.

○ Yes Skip to question 26.

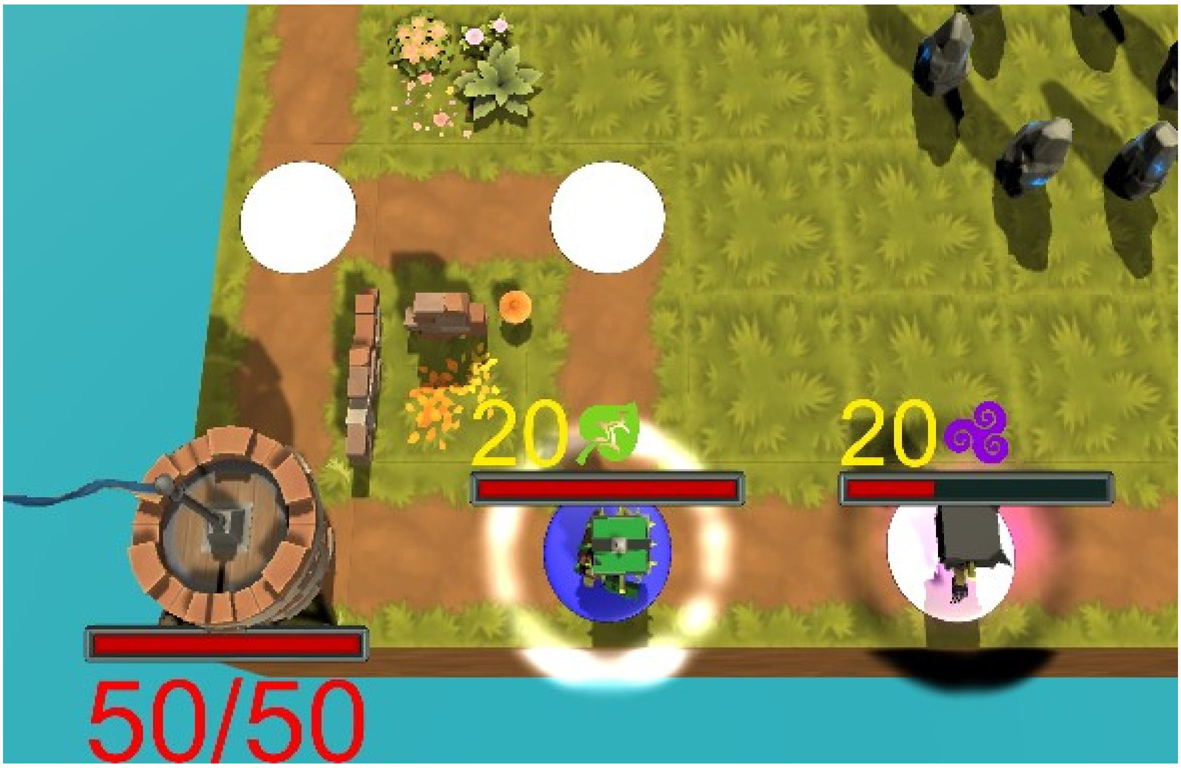
○ No Skip to question 27.

Skip to question 27.

26. Where can you see the stats of a unit? *

During this section different scenarios will be shown as a picture. We ask you to answer to the best of your ability what the outcome of the specified scenario is

Question 1



27. During the shadow units' next turn, it can * Tick all that apply.

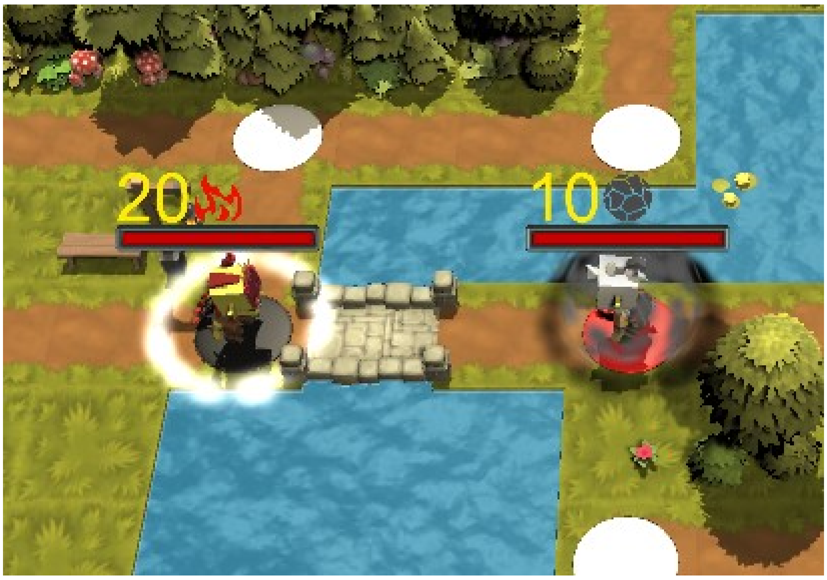
□ Attack the tower

□ Attack the unit from a different placement

□ Attack the unit from where it is standing, and it will live

□ Attack the unit from where it is standing, but it will die

Question 2



28. What will the outcome be if the left unit attacks the right unit? * Mark only one oval.

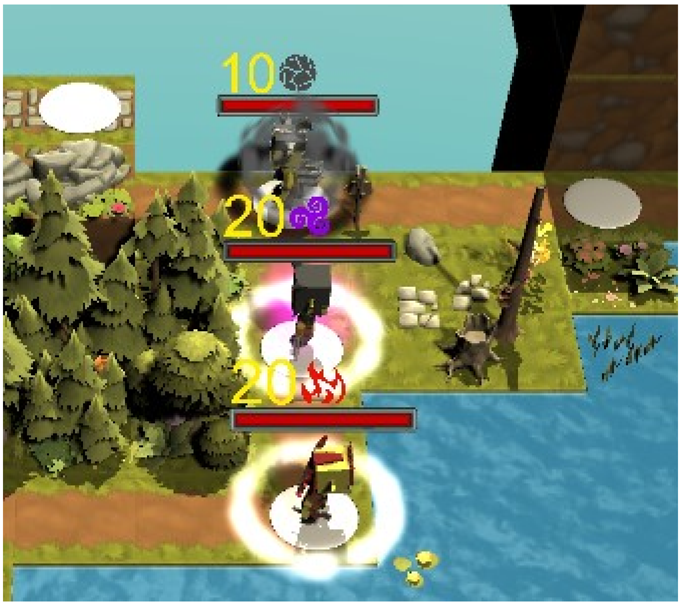
○ The left unit does full damage and kills the right unit

○ The left unit cannot attack the right unit

○ The left unit does reduced damage, and receives full damage from the right unit

○ The left unit deals full damage, and receives full damage from the right unit

Question 3



29. What can the shadow unit do on its next turn? * Mark only one oval.

○ The shadow unit can only move to the right most tile

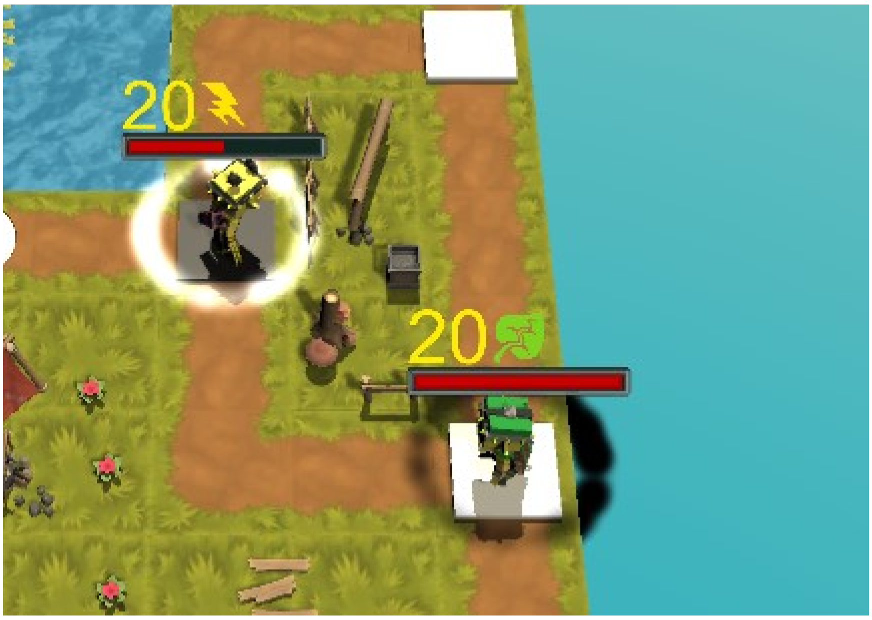
○ The shadow unit can attack both of the other units

○ The shadow unit cannot move because it is blocked

○ The shadow unit can only move to the left most tile

○ The shadow unit can move past both units to all visible tiles

Question 4



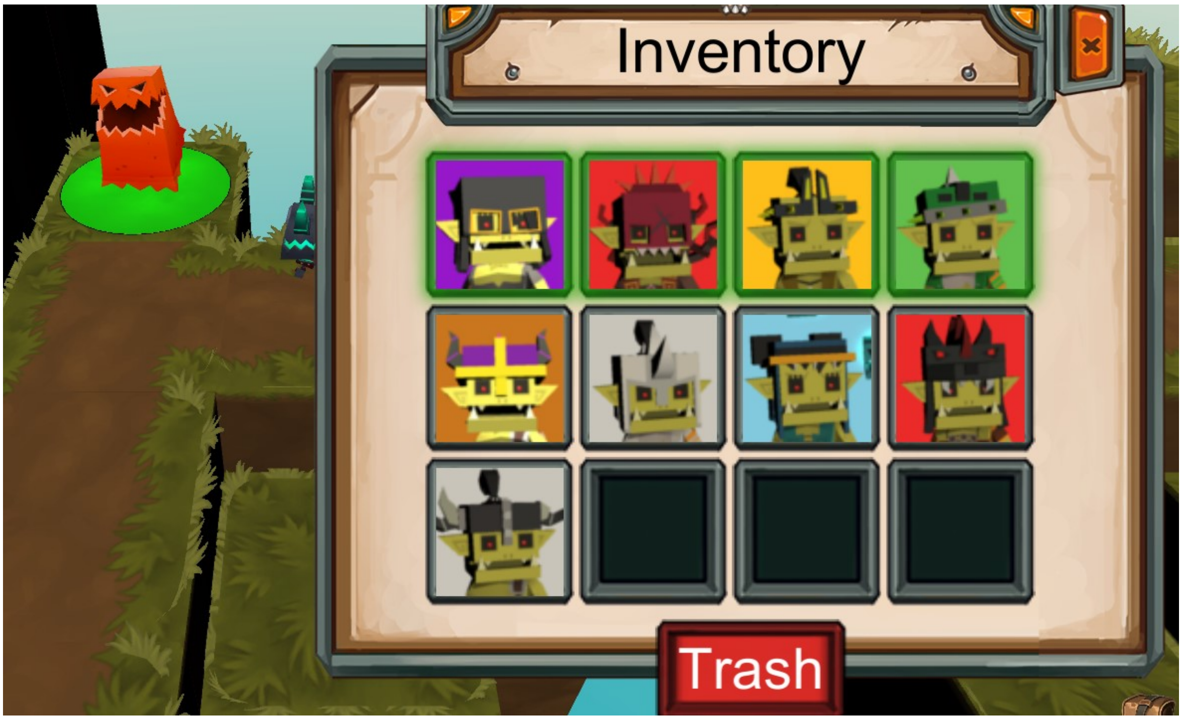
30. If the lightning unit attacks the nature unit the following will happen: * Mark only one oval.

○ The lightning unit will deal less damage to the nature unit, and will survive

○ The lightning unit will deal less damage to the nature unit and die from retaliation The

○ lightning unit will kill the nature unit in one attack

Question 5



31. The type of units the player will bring to this battle are: * Tick all that apply.

□ Water

□ Nature

□ Stone

□ Normal

□ Lightning

□ Fire

□ Shadow

Skip to question 32.

32. Was the information intrusive to your play? * Mark only one oval.

○ Yes, very

○ Yes, somewhat

○ No, not really

○ No, not at all

33. Was the information understandable? * Mark only one oval.

○ Yes, completely

○ Yes, most of the time

○ No, it felt out of place some times

○ No, I did not understand any of it

34. Do you consider this to be a good tutorial? * Mark only one oval.

○ Yes, very

○ Yes, at times

○ Not really

○ No, not at all

35. What makes you like it/not like it? *

36. Is there any additional information you would have liked to have gotten through the tutorial? *

37. Is there any information in the tutorial you feel did not need to be told? *

38. Did you have fun playing the game? *

Mark only one oval.

○ Yes, alot

○ Yes, at times

○ Not really ○ Not at all

39. What made the game fun/not fun? *

40. General comments

Please just continue through this section without making any changes, the giveaway will be presented once you have pressed submit!

41. ID - Should fill automatically DONT TOUCH *

Powered by
Google Forms