“Toll and Error”

CS39440 Web Based Gameification Project Report

Author: Damien Xavier Phillips ([bep36@aber.ac.uk](mailto:bep36@aber.ac.uk))

Supervisor: Dr Edore Akpokodje (eta@aber.ac.uk)

25th March 2023

Version 1.0 (Draft)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Graphics, Vision and Games (G450)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

Declaration of originality

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work, I understand and agree to abide by the University’s regulations governing these issues.

Name: Damien Xavier Phillips

Date: 25/04/2023

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name …………………………………………

Date ……………………………………………

Acknowledgements

I am grateful to my supervisor, Dr Edore Akpokodje, for all of his help in completing this project. I couldn’t have done any of this without his support.

I’d like to thank everyone at the Ceredigion Museum for meeting with me on multiple occasions to discuss the project and for providing further information on the historical events in the game.

I would also like to thank my boyfriend, Nathan, for his continued support throughout this project, and beyond it, as well as my furry and scaly friends, Poppy, Molly, Wing Commander the Lord Flashheart, and Mara, back at home for their companionship.

Abstract

“Toll and Error” is an application that aims to educate visitors to the Ceredigion Museum, and to visitors on the website, about the historical events of the Rebecca Riots in a fun and engaging way. The Rebecca Riots, which were a series of riots that took place in Wales in the 19th century, specifically in 1839-1845 these riots were in response to the toll fees that were charged to residents for use of their own local roads. These riots involved the rioters all dressing in traditional Welsh women’s wear and refusing to pay the tolls at the gates, and in some cases tearing them down. These rioters, who were predominantly men, called themselves “Rebecca”, or, more commonly, the “Daughters of Rebecca.” These riots were an important moment in Welsh history, and their story is still told to this day.

Interactive museum displays have become very popular, as they are, in the experiences of many, an excellent way to display information and convey it in a manner that is engaging, memorable, and that makes the museum experience fun and accessible to all. “Toll and Error” is an excellent example of such an exhibit, as it allows visitors to “step in to the story,” so to speak, and take on the role of a toll booth operator in an exaggerated, over the top, “Horrible Histories” style universe.

Once fully implemented, the “Toll and Error” game will be available on the museum’s website, allowing visitors to play and learn about the Rebecca Riots from the comfort of their own homes. In addition, the game may also be displayed in the museum itself, allowing visitors to engage with the exhibit in a more immersive way, as they will be able to see the toll board in question standing in front of them as they play.

Contents

[1. Background, Analysis & Process 6](#_Toc133591710)

[1.1. Background 6](#_Toc133591711)

[1.1.1. The Ceredigion Museum 6](#_Toc133591712)

[1.1.2. The History of the Rebecca Riots, and other miscellaneous research. 6](#_Toc133591713)

[1.2. Analysis 7](#_Toc133591714)

[1.3. Process 7](#_Toc133591715)

[2. Design 8](#_Toc133591716)

[2.1. Overall Architecture 8](#_Toc133591717)

[2.2. Detailed Design 8](#_Toc133591718)

[2.2.1. Even More Detail 8](#_Toc133591719)

[2.3. User Interface Design 8](#_Toc133591720)

[2.4. Other Relevant Sections 8](#_Toc133591721)

[3. Implementation 9](#_Toc133591722)

[4. Testing 10](#_Toc133591723)

[4.1. Overall Approach to Testing 10](#_Toc133591724)

[4.2. Automated Testing 10](#_Toc133591725)

[4.2.1. Unit Tests 10](#_Toc133591726)

[4.2.2. User Interface Testing 10](#_Toc133591727)

[4.2.3. Stress Testing 10](#_Toc133591728)

[4.2.4. Other Types of Testing 11](#_Toc133591729)

[4.3. Integration Testing 11](#_Toc133591730)

[4.4. User Testing 11](#_Toc133591731)

[5. Critical Evaluation 12](#_Toc133591732)

[6. References 13](#_Toc133591733)

[7. Appendices 14](#_Toc133591734)

[A. Third-Party Code and Libraries 15](#_Toc133591735)

[B. Code Samples 16](#_Toc133591736)

# Background, Analysis & Process

## Background

To prepare for the project, research was put into the following areas:

* The Ceredigion Museum
* The History of the Rebecca Riots
* GameMaker 2

Similar systems were also researched, such as other interactive museum displays in the Eureka! Science and Discovery Centre [1] and the Xplore! Science Discovery Centre [2], as well as games and shows of similar styles. “Horrible Histories” [3] was researched to assist with the playful, silly tone of the game, while “Papers, Please” [4] was researched to help with gameplay, as it is a critically acclaimed example of the gameplay features of the type of game that was to be created.

### The Ceredigion Museum

At the beginning of this project, it was essential to research thoroughly and gather as much information as possible. It was important for those involved to visit the Ceredigion Museum and view their displays in person. This allowed us to gain a more thorough understanding of the exhibits there, and provided a lot of inspiration for the final project.

During the visit to the museum, many “star exhibits” were shown as options for the project. This provided a wide variety of ideas to consider. After much discussion, it was decided that the final game would be a narrative-heavy game that took after the gameplay of “Papers, Please” and the tone of “Horrible Histories” called “Toll and Error” that focused on two star exhibits: The traditional Welsh attire and the toll board.

The decision was made to focus on these two exhibits was based on their local significance, due to the fact that the board is from nearby, and the fact that the story of the Rebecca riots piqued the interest of the team. The Welsh clothing exhibit showed beautiful traditional attire worn by women in Wales during the time period, and the toll board shows the toll fees charged during the 19th century, prompting the Rebecca Riots.

A second visit to the museum was then made to showcase the progress made on the games concept and storyline. This allowed discussion regarding a bilingual script, which then led to the English script being sent to the museum staff for translation.

### The History of the Rebecca Riots, and other miscellaneous research.

After a decision was made regarding the game’s topic, research then had to be done on the history of the riots. A lot of time was spent researching the event via articles online, podcasts, and discussion with the museum staff. This was quite an enjoyable part of the project due to the very interesting nature of the event and its prevalence in Welsh history.

In addition to researching the history of the riots, this phase also involved looking at the social and cultural context of the time. Learning about this allowed moving forward to look into how to make the tone of the script fun for all audiences. This led to looking into specific sketches from the BBC’s educational sketch comedy “Horrible Histories”. The specific sketch that inspired many parts of the script was the “Measly Middle Ages – Pay Rise” sketch that involved a medieval peasant approaching the knight he worked for to ask for a pay rise due to the plague wiping out many of them.

The team also worked closely with museum staff to ensure that the game accurately represented the exhibits and historical events depicted in the game. The staff provided valuable insight and expertise, as well as the aforementioned Welsh translation of the script.

### GameMaker2

When developing a game, one of the critical decisions that need to be made is selecting the right game engine. For the “Toll and Error” game, the team researched different game engines to determine which one would be the best fit for the project. After careful consideration, the team decided to use GameMaker, as it looked easy to use and had the features that would allow them to create an engaging and interactive game.

To test that the software was fit for purpose, a tester game was made. This enabled exploration of the games features, and finalised the decision to use GameMaker2 over Unity to create the game.

There were challenges encountered in the process of the game, such as the more advanced features of GameMaker, and trying to code things in GML, GameMaker’s text based programming language, that weren’t possible with GML Visual, GameMaker’s “code block” style programming language.

Deciding on, and subsequently learning to use GameMaker was critical in the development of the game.

## Analysis

The issue presented was that of a museum looking for a web-based gamification of their exhibits. To solve this problem, the task had to be broken down into smaller parts to make the whole thing more manageable. There was only one proposed approach to this, which was to first visit the museum, gain knowledge on the exhibits, decide on a topic for the project, research the topic extensively to provide historical accuracy and a compelling, entertaining game that’s tonally appropriate for all ages and abilities. The next step was to research how the game was actually going to be created. The game engine had to be decided, and then had to be researched, as despite being somewhat familiar with the game engine Unity from the “Computer Graphics and Games” module from the previous semester, prior experience with the software was still very limited, as before September 2022, the user had not been exposed to any game development software whatsoever, so was still very much a beginner in that regard.

Security is a very important issue when faced with a computing project, especially one that interfaces with the public and works with people directly, such as in the case of “Toll and Error”. However, “Toll and Error” does not take any data from users whatsoever, nor does it save anything between playthroughs of the game. Each run is completely independent from the last, with the only differences coming from decisions made within the game itself, such as language choice and decisions made by the player character. So, while still very important, security wasn’t much of a concern for this project.

The goal of this project was to present an original script written by the team via the means of a narrative game published via GameMaker’s own hosting service, GX.games. The smaller goals of this project to reach the final, ultimate end goal are:

* To gain an understanding of the Rebecca Riots for the purpose of writing a sensible script.
* To create a simple, mock-up framework for the game, for the purposes of showing both developers and museum staff how the game will look at the end of development.
* To create original graphics for the game based on the mock-up design.
* To write an original script for the game, that will then be approved and subsequently translated into Welsh by museum staff.
* To create a first, beta version of the game where there are no user decisions, and only one ending.
* To create a final version of the game where there are user decisions and two endings.

## Process

An Agile development methodology was utilised for this project. There wasn’t a rigid plan followed for this project which aligned very well with the flexible nature of Agile.

Basic principles of Agile were followed throughout the project, such as frequent communication with stakeholders (in this case, museum staff) to ensure that the project was meeting their needs, as well as adapting to changes as and when they arose. Tasks were also prioritised based on efficiency, which also aligns with the principles of Agile development, as does the clear idea of what features were needed.

An agile framework, such as Scrum or Kanban, was not used, as the development of this project was adapted very specifically to suit the work style of the developer. Choosing and then sticking strictly to a specific framework was not possible, not only due to the unknowns presented by the project, but the changing needs of the developer.

# Design

As previously mentioned, “Toll and Error” is a historical comedy game in the point-and-click adventure style of “Papers, Please”. The game follows the story of a toll collector from the 1830s-40s watching the Rebecca Riots play out in real time. The player is encouraged to make their own decisions regarding what to do about the rioters.

The game mechanics include dialogue options, and a point and click interactable objects within the world. The UI design had to facilitate this, and such, a significant amount of time was dedicated to creating a user interface that facilitates this. The user interface had to be both immersive and intuitive, and one that complemented both the game’s theme and setting.

In terms of the programming language, the programming language that was built into GameMaker2, GML, was used. GML visual was also used as it is very intuitive and beginner friendly. Many tutorials are available for both, enabling quick learning, and easy exploration of the software, especially during early phases of development.

Throughout the design process, multiple alternative designs and plot lines for the game were considered and rejected. These alternate designs included a cooking simulator style game based on the mining and processing of lead, A “What is that thing?” style of game using stome age artefacts, and a talent show rhythm game based on the Eisteddfod. While all of these ideas had merit, the team chose to focus on “Toll and Error” due to its story and local relevance.

## Detailed Design

Due to the narrative nature of the game, there was never many coded features. However, Coding has been required to implement those few features. Firstly, the game has multiple endings, and which ending the player is going to get is decided based on a score system. Certain decisions will raise the score, and certain decisions will lower it. Depending on the players score at the end of the game, the player will be shown one of two endings. This score is not visible to the player and is purely behind the scenes.

Another feature that had to be coded into the game was that of making the game bilingual and available in both English and Welsh. At the beginning of the game, there is a decision regarding language. Depending on which button the player presses, English or Cymraeg, that corresponding script is then loaded into the game using a Boolean variable that is altered depending on the button pressed. This decision regarding how to implement the language system was created to avoid having to hard code the dialogue, as this would have made making any edits very tedious and require sifting through code to do so. This could, in the future, also enable for further translations of the game in the future, as all that would have to be done is change the Boolean into another type of variable, and use that to load up more scripts in other languages.

## User Interface Design

The goal of the user interface design was to emulate the desk of a toll booth worker, and give various windows to show different areas of the booth, and a window for dialogue bubbles to appear inside. Everything on-screen is point-and-click, and is activated by the left mouse button. The layout is heavily inspired by that of “Papers, Please”.



Figure 1. The User interface of "Papers, Please"

Diagram

Description automatically generated

Figure 2. The User Interface of "Toll and Error"

Diagram, schematic

Description automatically generated

Figure 3. The User Interface of "Toll and Error", showing a customer as well as the dialogue.

The colour scheme of the user interface was based on several reference images of toll booths from Wales that existed in the 19th century. Muted, realistic colours were used to match the images found. A key image from the organisation History Daily [5] is shown below, alongside a mock-up of the final user interface design made with royalty free stock images. The sources of these images are credited in the appendices.

A small white building

Description automatically generated with low confidence

Figure 4. Aberystwyth Southgate Tollhouse



Figure 5. The mock-up of the final games User Interface

The font used for the project is Calibri. This is because, during talks with museum staff, the team were told that it is one of the best fonts for accessibility, and that the museum is required to use it for signage in order to make the museum as accessible as possible. The team also wanted to make the user interface as intuitive as possible for users who are, perhaps, visually impaired, young, or otherwise have trouble understanding user interfaces where too much is going on at once. Accessibility was a top priority of the team, as the developer has a lot of personal experience, both with themselves and their friends and family, with inaccessible software.

The user interface overall supports the design of the game, as, with the aid of the tutorial, it provides an intuitive and immersive experience when playing.

## Script Writing

As a narrative-heavy game, the script was very important to get right, and one of the projects greatest strengths. However, as this is a computer science project, it won’t be discussed in depth. As previously mentioned, much inspiration was taken for the story from other age-appropriate historical comedy media. A lot of time and consideration was taken to ensure that the script would be entertaining for all ages, appropriate for all ages, and understandable for all ages as well. The script also had to be translated to Welsh by museum staff. Both the English and Welsh scripts are included in the final project file.

# Implementation

Implementing this project provided many significant obstacles, and was very challenging to complete, and, despite the best efforts of the team, many of them were unable to overcome within the designated timeframe. No appropriate third-party code or libraries were able to be found to assist with this. However, many important aspects of the game were still complete, and, while there is no completed game, very important groundwork has been done. Due to the amount of work already completed, it could easily be continued by the current developer, or a more experienced developer could easily pick up and continue the project.

Loading the dialogue was perhaps the most significant hurdle encountered, and it is still not fully implemented. A lot of time had to be dedicated to trying to fix the script failing to load, but it was never done. Common bug fixes, such as checking the file path, and utilising the console to check various points in the code, but, unfortunately, nothing ever worked. This obstacle took a lot of time away from other tasks where time would have been better spent, simply due to the fact that the developer was not stuck on those tasks, such as turning the mock-up User Interface into hand-drawn graphics and character sprites. This, as well as other issues with the GameMaker2 program not opening on the available hardware, caused significant delay and prevented the game from being completed.

In conclusion, the implementation is visibly unfinished, and not all requirements have been met. While an understanding of the Rebecca Riots was gained, a framework for the game and its features was made, scripts were written and some original graphics were created, the beta of the game is unfinished, and, needless to say, the final version of the game was never made within the timeframe of the project.

DONE UP TO HERE

# Testing

Due to the very limited development of the software, there was no comprehensive testing needed. However, initially, simple tests were conducted. The plan was to have play testing take place, with several colleagues able to play the game through, and report any bugs that they found. There was also plans for the developers to play the game and go through a list of features to check them, such as dialogue progression, dialogue options, correct language, etc.

## Overall Approach to Testing

As previously mentioned, a play-testing method would have been employed for this project as well as testing by developers against a checklist of features.

This form of testing would address the design, as it would put the software into a situation where someone who had never interacted with the game before would play it, and a developer would get to see what these users, who were going in completely blind would do with the information given. This is an excellent testing method for many forms of software, but especially games. Many times game developers who have not ran user testing as part of their process are confronted upon release with the fact that the users will attempt to do things never before thought of by developers, and often end up breaking or crashing the software.

Due to the interactive nature of the system, it relies heavily on human input. This means that automated testing would not be appropriate given the situation. Unit tests were, and would be in the future, performed manually.

## Integration Testing

Integration testing was, and would have been if the software was complete, done with every new asset or code block that was brought together in the final piece of software. Every new asset or code block added was tested as it was added to ensure the software continued to run smoothly, and so that if it did break the system, it would be easy to narrow down the issue. During integration testing was where the issue with the dialogue was discovered.

## User Testing

As previously mentioned, in order to ensure that the software is user-friendly and could not be broken easily by end users, the software was planned to be tested on a group of users who had no previous interaction with the software behind the scenes. This was planned to be done using friends of the development team, museum staff, and the project supervisor. This would have allowed for a broad range of feedback, from finding criticisms based on historical context to criticisms and help based on technical knowledge. This wide range of individuals with various levels of knowledge on different aspects of the project would have rigorously evaluated the project.

This testing would have provided valuable insights into the software’s performance and usability to ensure that it was the best possible quality. Arguably, this is the best form of testing for games, although it certainly cannot exist in a vacuum.

# Critical Evaluation

The requirements for this project were extensively explored, and correctly identified according to staff at the museum. The design was also very robust and able to be displayed in all contexts in which it was needed, those being on a display in the museum itself, and on the museum’s website.

It is unclear if using Unity for this project would have made it easier to complete. It would have to be remade in Unity for this to be explored fully, as the developers would have to see for themselves if the bugs encountered with GameMaker2 would also appear in Unity.

Unfortunately, the software did not meet the needs of those expecting to use it. As it is not functional at this point, it would need a lot more work done to get it to a functional state. However, it is not for a lack of effort that the software was not finished. A lot of work has been done on the project which could be useful to the museum in other ways. The developers would also be happy to continue working on the project outside of the context of the major project if needed.

While the software is not finished, the process of getting to this current point shows valuable skills when it comes to working collaboratively with others and incorporating their requirements and feedback into the project. Through engagement with museum staff and the project supervisor, important insights were gained into the user needs and expectations.

A strength of the project’s design is its adaptability, and the fact that new parts of the game could be added very easily. If, for example, a new part of the script with a new character were to be added, it could be very easily added to the script with no bugs.

In regards to future projects, the strengths from this project could be applied to any other project, computer science or not. This project shows many desirable skills to have when creating any project that is needed, such as ability to research effectively, the ability to interface with stakeholders and to apply their needs to the project being developed.

If the project was started again from the beginning, prior research on the topic in question, in this instance, the Rebecca Riots, would have been done before the time constraint began would have been ideal. Issues in the developers personal life also delayed the project significantly, some were unavoidable and some were completely unexpected. Unfortunately, it is unclear what could have been done differently in these situations.

# References

|  |  |
| --- | --- |
| [1] | “Eureka! Science and Disovery Centre,” [Online]. Available: https://www.eurekadiscovery.org.uk/. [Accessed 2023]. |
| [2] | “Xplore! Science Discovery Centre,” [Online]. Available: https://www.xplorescience.co.uk/. [Accessed 2023]. |
| [3] | “BBC - Horrible Histories,” [Online]. Available: https://www.bbc.co.uk/iplayer/episodes/b00sp0l8/horrible-histories. [Accessed 2023]. |
| [4] | “Papers, Please,” 2023. [Online]. Available: https://store.steampowered.com/app/239030/Papers\_Please/. |
| [5] | “History Daily,” [Online]. Available: https://historydaily.org/rebecca-riots-when-welsh-peasants-revolted-tolls/3. |

# Appendices

DONE UP TO HERE

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

If you have used any 3rd party code, i.e. code that you have not written yourself such as libraries, then you must include Appendix A. In that appendix, you will provide details of the 3rd party code that you have used.

For most other items, it would be better to include them in your technical submission instead of including them as an appendix. For example:

* If you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document in the technical work. In this report, you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.
* If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification at the start of the project. Perhaps you used stories to keep track of the functionality and the ‘future conversations.’ If it isn’t relevant to include all those stories in the body of your report, you could detail those stores in a document in the technical work.
* If you have used manual testing, then include a document in the technical work that records the tests that have been done. In this report, you would talk about the use of those tests.

Documents included in the technical work or in the appendices are supporting evidence of the work done. Where you include documents, this report should refer to the documents. You should not be relying on detailed study of those documents in order to understand what is written in this report.

Speak to your supervisor or the module coordinator if you have questions about this.

* 1. Third-Party Code and Libraries

If you have made use of any third-party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third-party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what your original work is and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

The following is an example of what you might say.

**Apache POI library** – The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client’s existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [5]. The library is released using the Apache License [6]. This library was used without modification.

Include as many declarations as appropriate for your work. The specific wording is less important than the fact that you are declaring the relevant work.

* 1. Code Samples

This is an example appendix. Include as many appendices as you need. The appendices do not count towards the overall word count for the report.

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Project Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Random Number Generator

The Bayes Durham Shuffle ensures that the pseudo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs.

// Some example code here…