Development Project Documentation

M.A. Thesis Submission titled:

Monetising Video Game Mods

The perspectives of the modding communities

Joshua Ryan Astbury

11136632

j.astbury@ymail.com

Winter Term 2020/2021

for

M.A. Digital Games at Cologne Game Lab / TH Köln

Supervised by André Czauderna & Roland Klemke

Submitted on XX/XX/2021

Contents

1. Introduction

This documentation aims to provide additional insight into the practical segment of the Master of Arts degree in Digital Games with the Cologne Game Lab faculty of TH Köln. Please refer to section 1.1 for multiple links where the development project files can be sourced.

Please bear in mind that the submitted project was only tested on an up-to-date Microsoft Windows PC. Performance on other operating systems is unknown and could lead to unexpected results. If that should occur, I hope the included YouTube videos provide sufficient evidence of the prototype itself, working.

1.1 Project links

To avoid any potential issues in accessing the project’s files, two uploads were created, one a ZIP file, the other a Git repository. Whichever is accessed is up to the individual. That said, it is recommended to clone the linked Git repository which includes the .exe file along with source files. For the game development tools used please refer to section 2. All links are included below.

**Prior warning:** The music was found to be on the loud side for myself, please keep this in mind when using headphones.

A Google ZIP document is available here: ##GoogledriveZiplink

The Git repository is available here: <https://github.com/Youg3/MA-DG_DevProject_JRAstbury>

Finally, a YouTube playthrough video is uploaded here: <https://youtu.be/qxko4ctrAVw> along with one from within the Unity Engine: <https://youtu.be/AMKQtZmjgVc>

* 1. Development Project Concept

The concept of this project was to provide in-game mod tools to the player allowing them to edit usually hidden game variables and modify the next playthrough experience. The idea was to open as many hidden variables as possible, allowing a “*full*” modification (sometimes called an “overhaul”) of the game and the game playing experience. Such modifications like player created maps/levels were desired however were deemed to be too time extensive to implement. As such, the potential scope of this project was reduced to the modding tools provided detailed in section 3.

It is important to note that the project itself has had no balancing to the predefined values. When playing, please do not feel like this is the intended experience, game balance was not seen as a necessity to highlight the *mod-ability* of this project.

Should the player ever get stuck during a run through, pressing the key: **t** will open the mod menu prior to the end of the level. Other keyboard shortcuts do still exist, though are irrelevant as they are remnants of the Udemy course this project used as a base.

As there is no in-built tutorial, it is recommended that the player reads the Controls prior to playing. Figure XX shows the pop-up Control information panel.

##SCREENSHOT OF CONTROLS

* 1. Playthrough Loop

The intended playthrough loop for this project is as follows:

1. Player begins a new game.
2. Explores Tutorial Safe Area, leaves when ready.
3. Navigates through Battle Area.
4. Battle area completed.
5. Mod tools become available, Player mods game values for next player.
6. Saves values and playthrough ends returning to main menu.

Figure 1 displays a visual representation of this loop.

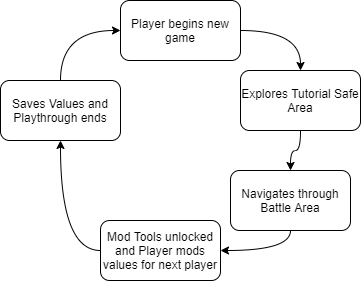


Figure . Concept Playthrough Loop

* 1. Game Areas

The project comprises of two playable areas: a safe area and a battle area. The safe area, shown in figure X, allows the player to familiarise themselves with the keyboard and mouse keys to be used while playing.

#SCREEN

The area itself is compact and includes some obvious interactable objects as well as ones that are hidden. For instance, the house the player character spawns in front of can be entered though this is unclear due to the intention of having hidden areas for the player to uncover. Once the player is ready to move on, they simply need to follow the road round until collision is triggered along the right side of the map.

The battle area, shown in figure X, contains two map modifiable features that are further explained in section 3.3.

#SCREEN

The player never has access to a mini map so must simply navigate across the areas intuitively. The level design aids the player with both areas progressing from left to right.

2. Tools Utilised

Being a solo development project, prior tool familiarity to enable a quicker development time was regarded highly. In this section the tools implemented and used for the development are detailed.

2.1 Unity Game Engine

The Unity game engine was chosen for the simple reason of prior familiarity. At time of project creation, the Unity version was the 2019 edition. Therefore, the version control allowed for Unity updates within the 2019 version and any further versions, even when prompt, were ignored.

At time of submission, the Unity version number for this project was: 2019.1.8f1. Whether this project can be successfully migrated to a newer version is unclear and recommended that the user avoids this potential issue.

2.2 Udemy course by James Doyle

To necessitate a more feature complete development project, an Udemy course was used and is linked below. The course provided by James Doyle provided a nice guide to help implement many features he saw as relevant for a 2D RPG styled video game. Not all were relevant for this development project but provided good practice and additional programming learnings despite lacking some much-needed code optimisations some of which were implemented and detailed in section 4. The vast majority of content, detailed in section ##, was provided through the course. Any additional resources referenced are also detailed in section ##.

Udemy Course Link: <https://www.udemy.com/course/unity2drpg/>

2.3 Github & the Fork client

A git repository hosted by Github was used as a version-control system which proved incredibly useful throughout the development cycle as errors and code breaks cropped up.

The 3rd party client called Fork was used to help manage the repository due to familiarity with the tool.

3. Mod Tools

This section details the mod tools implemented in this development project.

Upon reaching the battle area end trigger, the Mod Panel will activate and inform the player that they have completed their playthrough and can now mod the game values. Figure X shows the Mod Panel.

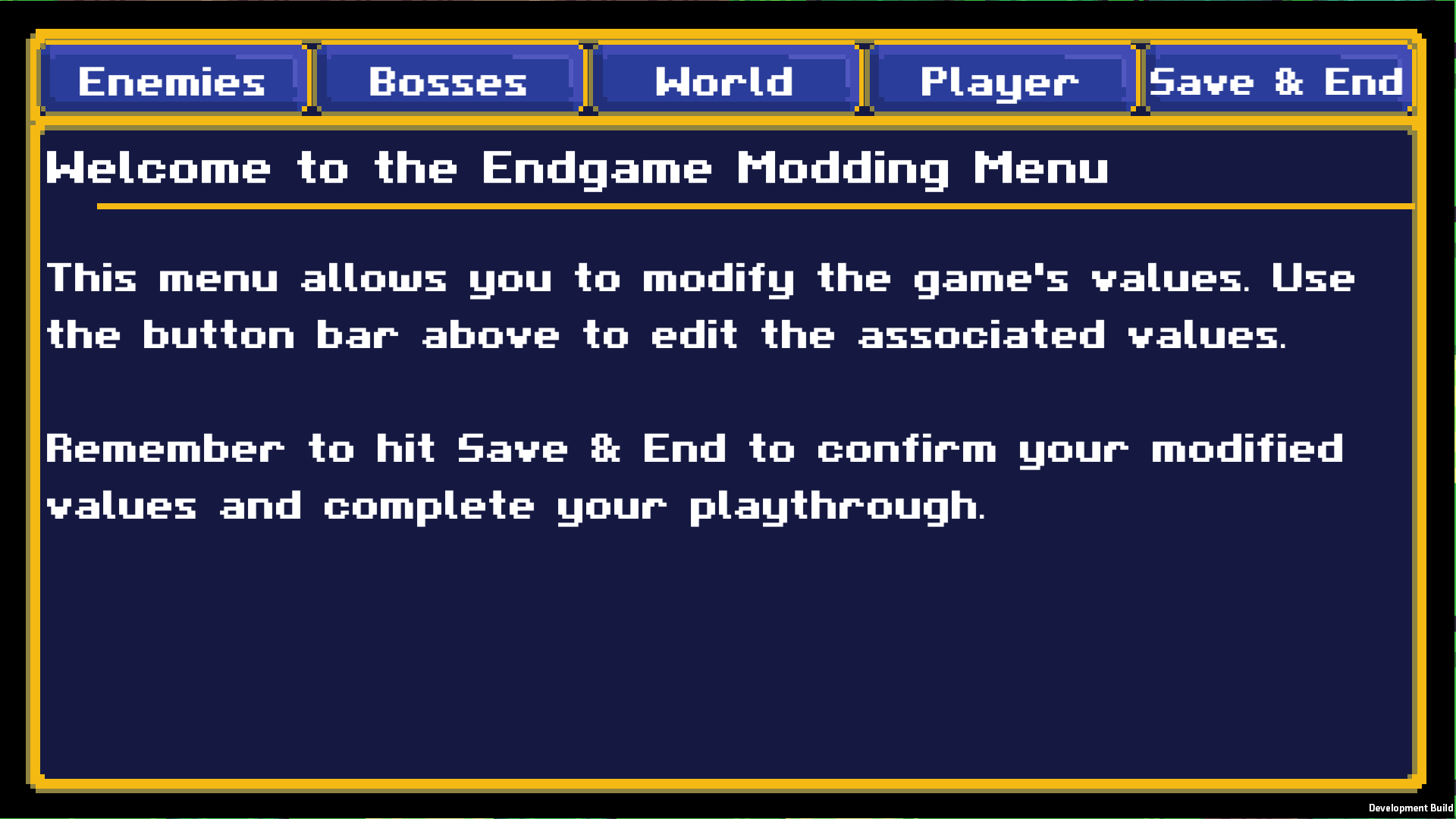


Figure 2. Mod Panel.

From here, the player can access the four sub-panels of:

1. Enemies (section 3.1)
2. Bosses (section 3.2)
3. World (section 3.3)
4. Player Characters (section 3.4)

Each of these sub-panels are detailed in the sections listed.

As the instructions state, once the player clicks the “Save & End” button from the top navigation bar, their playthrough will conclude and the game will cycle back to the main menu with the implemented new values.

3.1 Enemies Panel

The enemies can have their stat values modified from their dedicated section. Figure 3 shows the UI layout of this.



Figure 3. Enemies Mod Panel.

3.2 Bosses Panel

The Bosses panel has an additional “Enable” button which allows for the two bosses within the battle area to be either enabled/disabled by the player. When a boss is disabled, their icon will have its alpha values switched so that it is “faded out” while an enabled boss remains fully colourised. Because of this inclusion, a slight change to the UI was needed. Figure X displays the in-game Bosses panel.



Figure 4. Bosses Mod Panel

3.3 World Panel

The world panel enables the player to modify the values of the battle area in relation to two distinct elements. The first of which is the locking/unlocking of routes through the level. The second is the timer between potential battles within the six placed battle zones. Figure X shows the in-game panel.



Figure . World Mod Panel

Sadly, the composition of the battles themselves are not modifiable. This was considered and cut, with preference given to completing and refining the mod tools already included. Another consideration was the enabling and disabling of the six individual battle zones but was not implemented.

3.4 Player Character Panel

The Player Character Panel had a UI edit due to the cramped feel when trying to maintain a unison across all panels. This edit requires the player to click more though is felt to have a better readability when editing the values of each individual Player Character. Figure 6 shows the initial panel that is presented to the player.



Figure 6. Player Character Selection



Figure 7. Individual Player Character Mod Panel.

Figure 7 displays the change to the panel when the player has selected a character. Like the Bosses Panel (section 3.2), the additional two player characters can be “enabled” or “disabled” via the Enable button which switches the relevant icons alpha values to visually represent this. The main character, character 1, is the only character that cannot be disabled.

4. Issues and Fixes

Any issues and what the implemented solutions were, are outlined as well as any that are still present in the final submission. Potential workarounds are also detailed where required.

One major issue for the project base was the lack of updates to the course content from the course instructor, James Doyle. Luckily, the Udemy student base themselves conversed and suggested alternative solutions to these issues. Links to the student based suggested solutions are available in section XX.

2.4.1 Item pickup & Quest system bugs

Locking and unlocking unique items and quests never completely worked. The Udemy tutorial code would reset after each change of scene causing items to reappear and be collectable multiple times. This error was particularly confusing as the Game Manager which transitions between scenes in the same state, holds the Quest Manager and the Boolean log to which the items and quests themselves should reference prior to being accessed.

A separate unique item function similar to the Udemy quest system was attempted, though was ultimately scrapped as it never functioned correctly. A separate level/scene loader was believed to be the best solution to check what should be and should not be present. This was not implemented in the end as the quest system was essentially ignored for the submission.

Although the quest system was not used in the final concept, the unique items are in a very limited fashion via accessing the safe area house. Fixing this issue also allowed for greater insight into the enabling and disabling feature present for: player characters, enemy bosses, and battle area routes.

2.4.2 Combat bugs

A particularly interesting error arose with the combat damage calculation causing an unlimited amount of damage to be caused by the player characters. This was discovered as a mathematical error where the Udemy course code never accounted for the possibility of one of the stat values being zero. This was fixed by force setting the value to one (1) even if the player should set it to zero through the mod panel. This ultimately fixed the unlimited damage but does not prevent the player character from unleashing a massive amount of damage which will still one-hit everything.

2.4.3 Endless Sell bug

Although not a present function in the final prototype, a shop feature is ready though needed a tweak from the Udemy course code. The Udemy shop menu had a flaw where the player could endlessly sell any item. The student base solution proved to be valuable in figuring out this issue.

2.4.4 Modifying Character Stats

For the initial implementation, this seemed reasonably straightforward, successfully reading from defined prefabs, then writing to them upon save. For the Enemies and Bosses this seemed to work though the Player Characters were the issue. Upon start, the Game Manager holds a prefab of the Player Characters which is then used by the Battle Manager to reference from. This was overlooked and caused a few headaches but once found, a refactor of the Udemy code aided in understanding and assigning the correct directories for the mod tools to function correctly across all eight characters.

2.4.X Code Refactors

One of the major topics not covered by the tutorial was refactoring code into a more reusable state using inheritance. An example of this was, as the mod tools themselves began to form, a parent character class was coded for inheritance programming. This enabled greater reusability for the code and allowed the mod panels themselves to be coded once with only changes, done through the unity inspector, to which ID number each section should refer.

The Mod Panels and all the associated code essentially underwent massive refactors as own knowledge grew with continued practice. This makes many of the mod panel functions extremely reusable in comparison to the Udemy course content. A good example of this is the Load and Save functions which originally required individual input as to which character prefab it was reading and writing to.

Overall, refactoring own and Udemy code has greatly improved own understanding of key coding principles as well as the reusability of the important functions for this project.

#. Additional Reference Materials (Credits)

These materials are also detailed directly from the Credit panel within the development project.

|  |  |
| --- | --- |
| Material | Reference |
| Art/Battle Bosses | Stephen 'Redshrike' Challener, MrBeast, Surt, Blarumyrran, Sharm, Zabin |
| Art/Battle Enemies | Stephen 'Redshrike' Challener, hosted by OpenGameArt.org |

How to keep game states while loading different scenes - <https://answers.unity.com/questions/1457637/how-to-keep-game-states-while-loading-different-sc.html>

Unity Manual - <https://docs.unity3d.com/2019.1/Documentation/Manual/index.html>

Quest/Items Quest Manager - <https://www.udemy.com/course/unity2drpg/learn/lecture/12259482#questions/5953880>

Endless sell bug - <https://www.udemy.com/course/unity2drpg/learn/lecture/12259482#questions/6530700>