Turn-Based RPG

Daniel Nokes

2023 - 2024

**DID YOU SPELL CHECK? Are screenshots readable and clear? Did you attach and sign the CRF? Checked the video link? Checked each section has a relevant part in the mark scheme? Hole punched with a SINGLE hole in the top-left corner?**

Table of Contents

[Introduction 3](#_Toc139853909)

[Analysis 4](#_Toc139853910)

[Comparable Products 4](#_Toc139853911)

[Shin Megamin Tensei: Nocturne 4](#_Toc139853912)

[For A Vast Future 4](#_Toc139853913)

[Interviews and summary of results 4](#_Toc139853915)

[Initial Modelling 4](#_Toc139853916)

[Prototyping key/critical areas 4](#_Toc139853917)

[Mock-up of the User Interface 4](#_Toc139853918)

[Strategies for Design 4](#_Toc139853919)

[Simple Project Plan with Milestones and Critical Path 4](#_Toc139853920)

[Summary of SMART objectives 4](#_Toc139853921)

[Design 5](#_Toc139853922)

[Technical Solution 6](#_Toc139853923)

[AQA NEA Spec Table as a table of contents identifying key structures in the code. 6](#_Toc139853924)

[**SINGLE** Video showing the working solution demonstrating the completeness aspect. 6](#_Toc139853925)

[Source-Code repository and how to install/bootstrap your application 6](#_Toc139853926)

[Code Contents 6](#_Toc139853927)

[Testing 7](#_Toc139853928)

[Executable Unit Tests for all functions in the code 7](#_Toc139853929)

[Testing that the program can do what it does – demonstrate the core functionality. 7](#_Toc139853930)

[Testing the effectiveness of specific algorithms 7](#_Toc139853931)

[Evaluation 8](#_Toc139853932)

[Evaluation against SMART objectives 8](#_Toc139853933)

[External validation / Independent feedback that’s useful/realistic 8](#_Toc139853934)

[Evaluating the Interface 8](#_Toc139853935)

[Evaluating the Code 8](#_Toc139853936)

[Improvements by discussing the feedback in meaningful ways. 8](#_Toc139853937)

[References: 9](#_Toc139853938)

[Appendices: 9](#_Toc139853939)

# Introduction

For my project I have decided to create a turn-based RPG in style of classical RPGs like final fantasy, Zelda, and Pokémon. This game will be made in python using pygame. The game will focus on a main character that the user controls and by talking to npcs, fighting certain enemies, and using quest items to progress in the story. The game will be presented with a user-friendly GUI that should be able to be used without complications. The combat in the game will focus on the user choosing out of a couple options what attack/defence/item they would like to use and depending on the type of enemy they are facing it will determine the effectiveness of their choice. During combat the game will be able to process which order the characters / enemies fight by using the pre determinised speeds of each character. The game will include will contain many different types of enemies that each will each have a defining trait, these traits will change the way the enemy acts both inside and outside of combat. While many enemies will be abundant within the game there will be certain enemies that will act as bosses that will be activated by progressing through the main quest.

A key element in the game will be its monetary system, the user will be able to earn money by doing quests or winning fights, this will simply be saved as a variable that is added or subtracted from. The money earned will then be able to be used to buy new weapons, armour, and items, this will help the user power up while going through the game. Weapons that can be bought or found can be upgraded by fusing multiple weapons or items together, this will be done by each weapon and item having a sharable trait that, when used in fusion, will be applied to the chosen weapon. There will be three types of items; items used in combat, items used to upgrade weapons/armour and items used to progress through quests.

The map for the game will be traversed by the player where when the user moves the background scrolls in the direction of the character. This will be done by only allowing the player to see a zoomed in part of the background and depending on the direction of which the user moves the map with move as well in the determined direction. However, the map must only move when the user moves as if it moves by itself, it can cause the user to see areas they are not intended to see yet. When the user reaches the edge of the map the background shouldn’t slide off into nothingness instead it should stop moving, only moving again if the user moves in a direction that isn’t the edge.

**End Users:**

The end user for my game will be my friends that play games and people from online forums that have an interest in turn-based RPGs. Talking with people interested and reading reviews on related projects there a couple of common problems that limit the turn based RPGS:

* Difficulty system: Many RPGs lack a difficulty system; this means that the game isn’t balanced for different types of consumers. This problem could cause a large part of the consumer base to drop the game as its either too hard or easy.
* Poorly communicated directions: In some games it is difficult to understand what you’re supposed to do or where you’re supposed to go, this causes a lot of complications and can cause people to stop playing as they can’t progress with the game.

**Criteria:**

These first criteria are what must be implemented into the game, these are generalised aspects that will be implemented in:

* M1: Be able to fight enemies in a turn-based combat environment.
* M2: Be able traverse a map that has enemies, locations, and items on it.
* M3: Must have a collision system so the user and npcs/enemies can’t walk through walls.
* M4: Must have music and sounds for different parts of the game.
* M5: Must have a monetary system that the user is able to gain and use their money.
* M6: Must have different type of enemies.
* M7: Must have multiple different items that the user can use and fuse together.
* M8: Must have a levelling system where the user can grow stronger.
* M9: Must have a story quest that allows the user to fight different things and interact with npcs to progress in it.
* M10: Must be presented with a user-friendly GUI.

These criteria are more specific that should be implemented but if needed can be avoided if proven too difficult to code:

* S1: Each enemy should have a different element to increase variety in combat.
* S2: The main character should have a choice in different attacks that are chosen pre-battle and are obtained through levelling up, chests and buying them.
* S3: The enemies should walk on the map even when not in view of the main character.
* S4: The main character should have allies that can be controlled during battle.
* S5: The combat should be based around speed determining who attacks first.
* S6: During combat the user can use items, like potions, to aid them.
* S7: Should have boss enemies that will drop strong items to make progressing through the story worth it.
* S8: The user should be able to equip new weapons and armour to increase their strength.
* S9: Enemies should become stronger when the user progresses through the story to ensure the game doesn’t become too easy.
* S10: Items sold in shops should change when the user becomes stronger, so the monetary system is seen as useful.
* S11: Different attacks should cause different ailments, e.g., bleed, poison, freeze.
* S12: Certain areas should be locked until the user has progressed through the story far enough.

# Analysis

## Comparable Products

As the turn-based RPG genre is one of the oldest video game genres there are a vast number of games that all fall under this category. This allows me to analyse why these games are loved and what is frequently praised and what isn’t and lets me adapt my game to its best.

### Shin Megami Tensei: Nocturne



Shin Megami Tensei is a jrpg that uses turn based combat that is aimed for more skilled gamers instead of casual. During the combat each ally and enemy has a type (fire, air, ice, etc), this allows for different attacks to be stronger or weaker based on who they attack. The order in which characters go in works by having each ally and enemy having different speeds, this speed effects who attacks first and the complete order of each character in combat. After winning a battle the user is rewarded with exp, a form of currency and a chance of an item which depending on the difficulty of the enemy will determine the amount you gain and if the user loses the battles, they are penalised by losing money and having the chance for an item to be stolen. Shin Megami Tensei uses the classic HP and MP system to portray the statuses of allies, this is a simple and effective way of showing this as it is not too complicated to understand.

I intend to implement many aspects of the gameplay of Shin Megami Tensei as the combat is one of the strongest praises of the game. I will be adding speed stats into my RPG as it creates another variable that the user must think about, this allows for a vaster variety of strategies and creates more dynamic battles while not being too complicated so the user can’t understand it. The reward system after battles will also be implemented as it gives a sense of accomplishment for each battle the user wins, this will also help move the story along as the better rewards dropped by stronger enemies create an incentive for the user to progress. To create more dynamic gameplay enemy and ally types will be a must, this forces the player to think more when fighting as it stops the possibility of spamming the same attack multiple times to beat every enemy. The GUI will be using a HP and MP bar when in combat so the user can see if they are winning or losing.

Shin Megami Tensei also includes fusion to help increase the power of the user. This is done by the user fusing two or more of the demons that are part of their team into one stronger one. Shown above is the fusion chart used, when looked at there are no specific demon names written instead their types are written and depending on what two are fused together will determine the type that will be produced. This will influence the item fusing idea where only attributes are fused instead of two specific items, this helps negate the need to hard code every item combination possibility.

### For a Vast Future



For a Vast Future is another turn based rpg, while the main concept is very similar to Shin Megami Tensei instead the execution is very different. This is due to Vast Future being presented in a 2d pixeled setting rather than being in a 3d environment. My game will be presented in the same format as creating and executing a rpg in 3d would take more time then available.

In For a Vast Future when moving the background moves along with the character, this works by having the character stay in the middle of the screen with the background moving depending in which direction the character is going. When the character reaches the edge of the map instead of the background scrolling off screen going to black it locks in place, this means the when the character moves the background stays while their character moves, this allows for further exploration while presenting it in a nice format.

My game will most likely follow this map design as it can be implemented by having a bigger map that zooms into a select area while leaving the rest of the map still active, this allows for a more dynamic game as enemies will be able to traverse the map even when they’re not on the viewable screen.

## Initial Modelling

All images shown here are rough simulations of what is to be expected in the final product.

**Initial Menu Model**

A screenshot of a video game

Description automatically generated

This menu will be the first screen the user will see when they boot up the game, it will have four different choices the user can chose from. First selecting new game will allows the user to begin the game from the start. Secondly selecting the second load button will allow the user to continue with the game if they have a save file. Next the settings button will open a new list that has different settings the user will be able to change if they want. Lastly the exit button will close the game if selected.

**Battle Model**



This model simulates what the battle system will look like, on screen you will be able to see both the enemies and the heroes. The menu at the bottom of the screen serves two purposes first is to present the hp and mana of allies, and the other is to allow the user to attack/use items/ run in combat.

At the beginning of combat the user will be able to select what they want to do; if attack is chosen the user will then be able to chose which enemy they want to attack (shown in model by using an underline), if skills is chosen a new list will appear allowing the user to choose from the skills they have previously equipped then they can select the enemy to use it on, if defend is used to character which used it will receive a percent less damage that they would usually take, if items is selected a new list will come up showing all useable items that the user has and run lets you attempt to leave a fight without attacking.

The user will be able to select an action to do with all allies on their team, depending on the ally they will each have different skills and will deal different amounts of damage depending on their stats.

To make the game appear more polished the background of the battle will change depending on the area that the fight began in.

**Overworld/Map Model**

A video game screen with a cartoon character and a sword

Description automatically generated

This model shows how the user will traverse the map and how the user can interact with the world, as shown the main character will be in the middle of the screen with the background moving according to the users’ inputs. To interact with objects/npcs/enemies the user will need to move their character to whatever they want to interact with.

In the image on the right an enemy can be seen, while the game is active enemies will be able to walk around the map with most having some sort of radius where if the user stands in it, they will begin to walk towards them. Different types of enemies will have different walking patterns and different reactions to when the user gets near them. Once the user touches an enemy a fight will begin showing the battle screen.

On the left is a chest, many objects like this chest will be on the map, when the user interacts with an object a unique interaction will happen, for example if an item that can be picked up is on the floor the user can choose whether they want to take it.

**Menu Model**

**A video game screen with a cartoon character

Description automatically generated**

While in the menu the user will still be able to see the map and what was there however everything will be paused, this means while the user is in the menu, they are safe.

The menu has 6 different options that the user can chose from, first being inventory, this allows the user to see all the items and equipment that they have in their inventory and if the user wants, they can use an item, for example if they select a health potion, they can use it on someone in their party. The next option is skills, this allows the user to view the skills each party member has. While most skills will be useless outside of battle some can be used in the overworld, for example if a party member has a healing skill, they can use it at the cost of mana. The third option is equipment, this will allow the user to see what each ally has equipped and allows them the change it if necessary. The status choice will be used to view the party’s health, mana, and equipment. Saving the game will save the exact location the user is standing in, what items they have, all their characters stats including equipment and what they have and haven’t unlocked, this is so the user can continue the game from where they finished last time. The final button will close the game down, before closing the user will be asked if they want to save.

Using the menu, the user will also be able to see how much gold they have gathered.

## Prototyping key/critical areas

**Movement**

As the movement is essential in my rpg having a working prototype is critical as it will help me build the game from almost nothing. Using the pygame website they provide a basic setup of how to use pygame, the example they provide shows a backscreen with a circle that moves whenever the user presses wasd. While the example does provide a way in which a character can move, I would prefer my game to have movement where the character appears to be moving while the background is actually moving.



A screen shot of a computer program

Description automatically generated

Shown above is how when w, a, s or d is pressed the coordinates of the map moves in the opposite direction pressed as this causes the illusion the character is moving accordingly. Scree.blit() is used so that each time a button is pressed the screen updates the coordinates of the map, this allows the map to appear as it is showing. While this did do what I want it caused problems as when the character got to the edge of the map the camera would follow them into the void, as this would break the game, I had to fix it.

A black screen with white text

Description automatically generated

As a basic fix I would check if the character was next to the edge of the map and if so the character will move instead of the map. This allowed to keep the immersion of the character moving without the camera going into the void. However, while this worked for moving left and up as the edge position is 0 for both the right and bottom sides I had to create an equation to figure out where the edge was.

A screen shot of a computer program

Description automatically generated

First using the get\_width() and get\_height() I could get the resolution of the background image, this value would then be multiplied by -1 and then the values of the games screen size (1280 x 720) would be added to the coordinates. This fixed the issue of the camera moving off the map however if the character moved, they wouldn’t return to the middle if they tried to move in the opposite direction.

A black screen with white text

Description automatically generated

Finally, to fully finish off the movement system I made it so it checked whether the character where at the coordinates of the middle of the screen, if so, it would let either the map or player move but if not if the player tried to move in the opposite direction when they reached the middle of the screen the character would stop moving and the map would instead.

## Simple Project Plan with Milestones and Critical Path

# Design

Overview of how the system works and soft/hardware requirements.

HCI/Screen designs and transitions modelled.

Diagram showing the entire system (structure/class diagram explaining key relationships.)

ER diagrams if using databases.

Pick each high-Level SMART objective as title.

Show key algorithms and functionality.

# Technical Solution

## AQA NEA Spec Table as a table of contents identifying key structures in the code.

## **SINGLE** Video showing the working solution demonstrating the completeness aspect.

Remember to use a MONOSPACE font like consolas – do NOT use Arial! Preferably shorten your URLs and insert a QR Code graphic to make it easy to open.

## Source-Code repository and how to install/bootstrap your application

## Code Contents

Use a monospaced font. Use a Syntax highlighting to make it easy to read code. Remember no “Widows and orphans”.

# Testing

Link to Video in previous section that showed the functionality and testing of the app.

## Executable Unit Tests for all functions in the code

## Testing that the program can do what it does – demonstrate the core functionality.

## Testing the effectiveness of specific algorithms

# Evaluation

## Evaluation against SMART objectives

## External validation / Independent feedback that’s useful/realistic

## Evaluating the Interface

## Evaluating the Code

## Improvements by discussing the feedback in meaningful ways.

# References:

(Citation, 2023)

# Appendices:

All code listings here, any extras e.g. HTML, CSS, etc