Nimble RPG

Documentaion

Adrian J.

2023

Content

[The Project 3](#_Toc129112266)

[Brainstorming 3](#_Toc129112267)

[Tasks (unsorted) 4](#_Toc129112268)

[The base 4](#_Toc129112269)

[Utilities 4](#_Toc129112270)

[Game features 4](#_Toc129112271)

[Game finish 4](#_Toc129112272)

[PSP 5](#_Toc129112273)

[Tasks (sorted) 6](#_Toc129112274)

[The base 6](#_Toc129112275)

[Utilities 6](#_Toc129112276)

[Game features 6](#_Toc129112277)

[Game finish 6](#_Toc129112278)

[Documentation 7](#_Toc129112279)

[The base 7](#_Toc129112280)

[Creating Framework 7](#_Toc129112281)

[Creating main loop 7](#_Toc129112282)

[Handle drawing 7](#_Toc129112283)

[Handle UIs 7](#_Toc129112284)

[Update game objects 7](#_Toc129112285)

[Manage window events 7](#_Toc129112286)

[Create camera 7](#_Toc129112287)

[Conclusion 7](#_Toc129112288)

[Utilities 8](#_Toc129112289)

[Create UI 8](#_Toc129112290)

[Create game object 8](#_Toc129112291)

[Create player 8](#_Toc129112292)

[Create entity 8](#_Toc129112293)

[Create animations 8](#_Toc129112294)

[Create item 8](#_Toc129112295)

[Create inventory 8](#_Toc129112296)

[Conclusion 8](#_Toc129112297)

[Game Features 9](#_Toc129112298)

[Base game loop 9](#_Toc129112299)

[Implement looting system 9](#_Toc129112300)

[Create concepts of crafting ideas 9](#_Toc129112301)

[Implement inventory crafting 9](#_Toc129112302)

[Implement alchemy 9](#_Toc129112303)

[Implement smithing 9](#_Toc129112304)

[Implement enchanting 9](#_Toc129112305)

[Create skill tree 9](#_Toc129112306)

[Test different leveling systems 9](#_Toc129112307)

[Create final leveling system 9](#_Toc129112308)

[Conclusion 9](#_Toc129112309)

[Game finish 10](#_Toc129112310)

[Implement saving and loading system 10](#_Toc129112311)

[Create graphics 10](#_Toc129112312)

[Create sounds 10](#_Toc129112313)

[Make music 10](#_Toc129112314)

[Create map editor 10](#_Toc129112315)

[Create world map 10](#_Toc129112316)

[Test the game in public 10](#_Toc129112317)

[Fix found bugs 10](#_Toc129112318)

[Conclusion 10](#_Toc129112319)

# The Project

The project “Nimble RPG” is a 2D-Top-Down RPG with focus on open world and skill systems.  
It is created and built by Adrian Jahraus in 2023 and serves as start to get into game development.  
The game is written in C++ using the [Raylib](https://www.raylib.com/) library for rendering, handling collisions and other useful stuff coming with the library.  
There is also a [YouTube Channel](https://www.youtube.com/channel/UCKWhMp1kpjG3eCN6Cihf9AA) cowering a lot of game updates as well as updates for the game, it serves as part of the documentary.

# Brainstorming

The following points serve as total overview on what should be achieved by the project and how:

* Framework must be created that handles window events, game loop, updating game objects,  
  handling UIs as well as rendering stuff to the screen each frame.
* Create main loop that loops the framework as well as calculates delta time for each frame.
* UI class that handles drawing UIs with buttons and info on screen and handle button pressed by mouse.
* Create a game object class that has a struct for saving purpose as well as game related functions.
* The game object class must contain a virtual function for handling inputs done this frame, a virtual function that updates the game object every frame and takes delta time as argument, a virtual function that checks if any movement should be calculated this frame taking a pointer to a vector containing all game objects that should be checked for collision and also a virtual function that draws the game object, if it should be drawn, to the screen at its position.
* The functions for input, updating, moving and drawing shouldn’t return anything.
* A player class must be created that inherits from the game object class. The class must handle its own saving struct containing player skills and an inventory.
* Inventory class that contains a vector of item class and has a function for drawing and calculating the stats of given game object.
* Create an item class that contains its own stats.
* Camera class that inherits from Raylibs Camera2D that controls the viewport of the game, it should track the current actor that is given by the framework.
* Open world single player RPG with crafting mechanics, skill trees, leveling system, magic combat, ranged combat, melee combat.
* Crafting must contain alchemy, smithing and enchanting.
* Skill trees can improve the players skills in crafting and combat by spending points from leveling up.
* Animation class that handles the speed and current frame of animations with indexing and delta time.
* Entity class inheriting from game object that contains routs that can be moved as well as following points on the grid.
* Saving and loading system implemented.
* Map editor for generating the game map.

# Tasks (unsorted)

Following tasks were created after the brainstorming and are a short overview of what must be done. The list serves as base for creating a sorted list with time management and dependencies.

## The base

* Creating Framework
* Creating main loop
* Handle UIs
* Handle drawing
* Update game objects
* Manage window events
* Create camera

## Utilities

* Create game object
* Create UI
* Create player
* Create entity
* Create animations
* Create inventory
* Create item

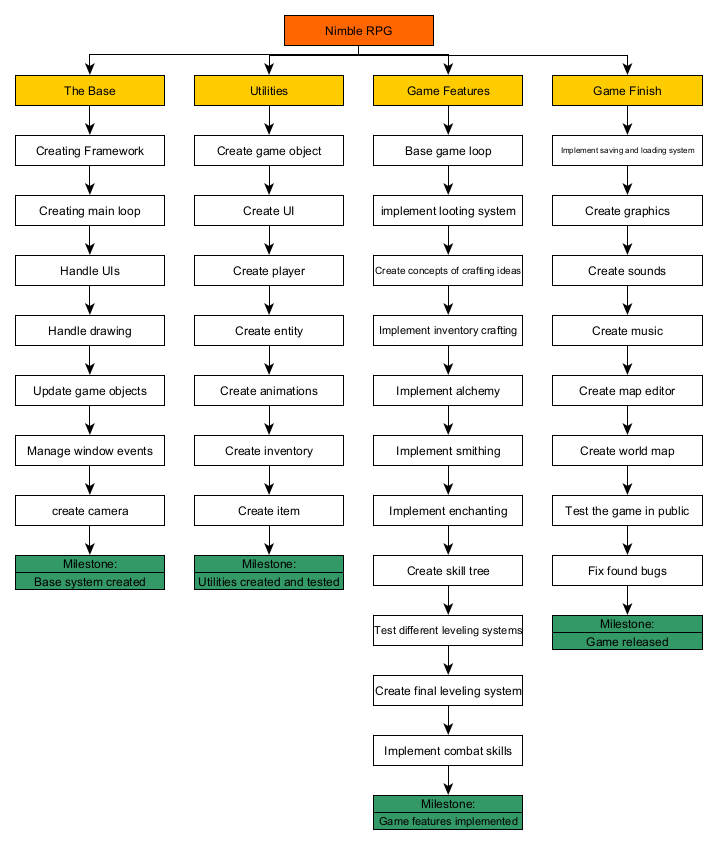
## Game features

* Base game loop
* Implement looting system
* Create concepts of crafting ideas
* Implement inventory crafting
* Implement alchemy
* Implement smithing
* Implement enchanting
* Create skill tree
* Test different leveling systems
* Create final leveling system
* Implement combat skills

## Game finish

* Implement saving and loading system
* Create graphics
* Create sounds
* Make music
* Create map editor
* Create world map
* Test the game in public
* Fix found bugs

# PSP



# Tasks (sorted)

Sorted task list with dependencies. Every milestone is mandatory for next step.

## The base

* 1. Creating Framework (D: 2)
  2. Creating main loop (D: 1 ; VG: 1.1)
  3. Handle drawing (D: 1 ; VG: 1.2)
  4. Handle UIs (D: 2 ; VG: 1.3)
  5. Update game objects (D: 1 ; VG: 1.1)
  6. Manage window events (D: 1 ; VG: 1.1)
  7. Create camera (D: 1 ; VG: 1.3)
  8. Milestone: Base system created (VG: 1.7, 1.6, 1.5, 1.3)

## Utilities

* 1. Create UI (D: 2)
  2. Create game object (D: 2)
  3. Create player (D: 2 ; VG: 2.2)
  4. Create entity (D: 2 ; VG: 2.2)
  5. Create animations (D: 1)
  6. Create item (D: 1 ; VG: 2.2)
  7. Create inventory (D: 2 ; VG 2.6, 2.1
  8. Milestone: Utilities created and tested (VG: 2.7, 2.5, 2.4, 2.3)

## Game features

* 1. Base game loop (D: 5)
  2. Implement looting system (D: 2 ; VG: 3.1)
  3. Create concepts of crafting ideas (D: 2 ; VG: 3.2)
  4. Implement inventory crafting (D: 3 ; VG: 3.3)
  5. Implement alchemy (D: 4 ; VG: 3.4)
  6. Implement smithing (D: 4 ; VG: 3.4)
  7. Implement enchanting (D: 4 ; VG: 3.4)
  8. Create skill tree (D: 3 ; VG: 3.7, 3.6, 3.5)
  9. Test different leveling systems (D: 2)
  10. Create final leveling system (D: 3 ; VG: 3.9)
  11. Implement combat skills (D: 6 ; VG: 3.1)
  12. Game features implemented (VG: 3.11, 3.10, 3.8)

## Game finish

* 1. Implement saving and loading system (D: 2)
  2. Create graphics (D: 5)
  3. Create sounds (D: 6)
  4. Make music (D: 4)
  5. Create map editor (D: 2)
  6. Create world map (D: 6 ; VG: 4.5, 4.2)
  7. Test the game in public (D: 10 ; VG: 4.6)
  8. Fix found bugs (D: 10 ; VG: 4.7)
  9. Game released (VG: 4.8)

# Documentation

The documentation explains how every task is being handled and how the goal of each is achieved.

## The base

### Creating Framework

### Creating main loop

### Handle drawing

### Handle UIs

### Update game objects

### Manage window events

### Create camera

### Conclusion

## Utilities

### Create UI

### Create game object

### Create player

### Create entity

### Create animations

### Create item

### Create inventory

### Conclusion

## Game Features

### Base game loop

### Implement looting system

### Create concepts of crafting ideas

### Implement inventory crafting

### Implement alchemy

### Implement smithing

### Implement enchanting

### Create skill tree

### Test different leveling systems

### Create final leveling system

### Conclusion

## Game finish

### Implement saving and loading system

### Create graphics

### Create sounds

### Make music

### Create map editor

### Create world map

### Test the game in public

### Fix found bugs

### Conclusion