Game Design Document

Top Down action RPG

Document version 0.1

Overview

This document is a combination of the Function Design, Technical design, Technical Analysis and Plan of Approach. I suggest reading these files separately since it would make more sense and give you the information you are looking for faster.

Objective

A 16/8 bit prototype game with mouse + keyboard and controller support. With minimum 3 levels and a boss level. Needed powerups, health, money and at least 1 puzzle. At least 1 enemy that follows the player when he gets to close. There needed to be a main menu, overworld and levels with pause menu.

Chosen solution

This concept is all about a combination of solution #3 and #1 as the client wished for. We'll create a little overworld with 4 dungeon entrances where the village is taken by mystic creatures. Behind the last door is a big creature(the boss). You want to save the world but you are not strong enough. The player needs to gather strength and items in the first 3 levels to be able to slaughter the boss. By going through doors that are open you can get in a level.

Possibilities

- Mobility / Walking system.
- Inventory system (Picking up items and use items).
- Shop system (Buy items).
- Minimap system (Show player, maria)
- Interacting system with A.I. (Maria and enemies).
- Teleportation system for entering levels.
- Village and forest in 1 map.
- Fighting system with the enemies.
- 8/16 bit art style.

Impossibilities

- None

Requirements

Software required for this project:

- Unity 2018.3.7f1 (Game engine: used to combine scripts with assets making the game and compile it to a .exe).
- Photoshop (Used to create art).
- FL Studio (Used to engineer sound).
- Visual Studio (Used to write code for the game).
- Sourcetree (Used for our GitHub repository).
- Chrome (Used to communicate with our team and to distribute the game).
- Windows (Used to run Unity, Visual Studio, Chrome, FL Studio, Photoshop The game).

Hardware required for this project:

- 3 computers or laptops that can handle Unity, FL Studio, Photoshop

  Minimum requirements

  - processor: i7 6700k
  - Graphics card: GTX 960m
  - Storage: 512GB SSD/HDD
  - Ram: 8GB

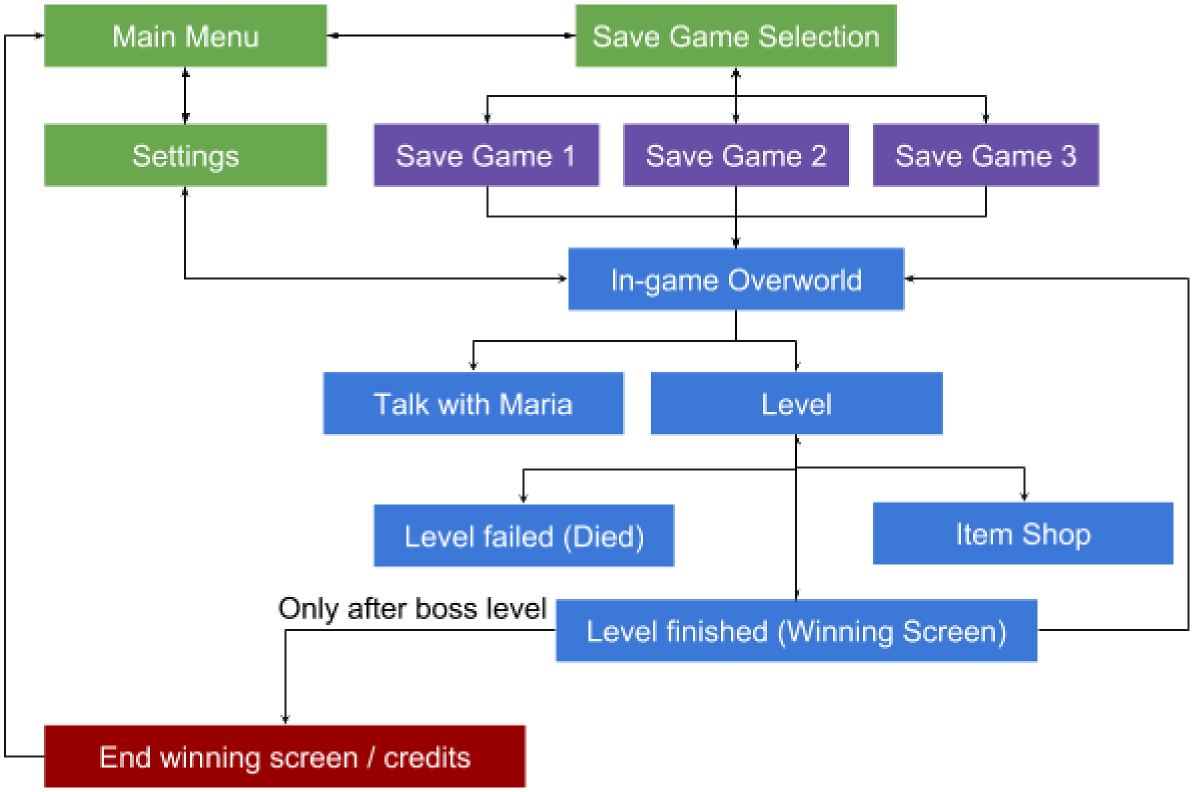
Story and Background

The game will be played in a 16 bit world. Everything should be made with 16 bit art. The world you live in is taken by strange enemies who came from the forest. The boss of these creatures is in the 4th dungeon slaughter him and every creature will die with it. If you are not strong enough you should go to the other levels to gather strength and items. The dungeon exists out of randomly generated mazes. The dungeon has a slightly dark tint.

Characters

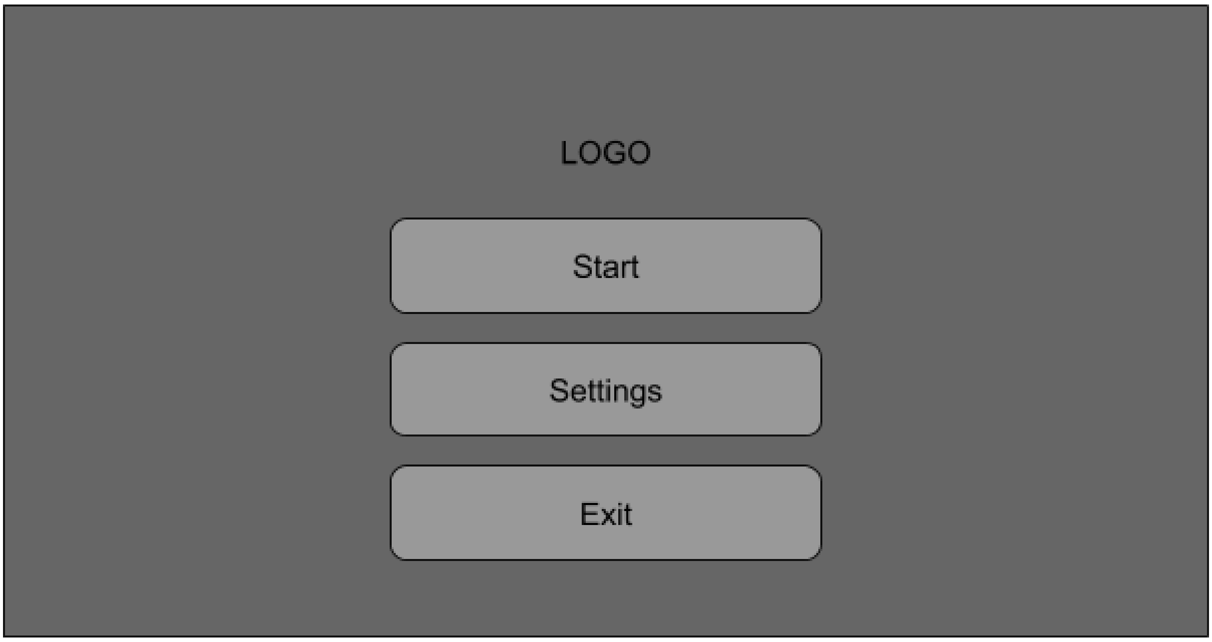
Who	**Goal / Purpose**
You (the Player)	Liberate the village! Slaughter the monsters and solve puzzles. The final goal for the player is to kill the boss to free the village.
Maria	Maria is your best friend. She doesn’t want you to get out there but since your mother got killed by these creatures she knows she can’t stop you. Maria is very intelligent and will try to help you when possible.
Creature(s)	Creatures try to kill the player and protect the items the player need to gether. They won’t let you win that easily.
Boss	The leader and biggest creature from them all. He is really strong, way stronger than the normal creatures. When the boss gets killed all the creatures will die because they are connected through their leader.

Screenflow

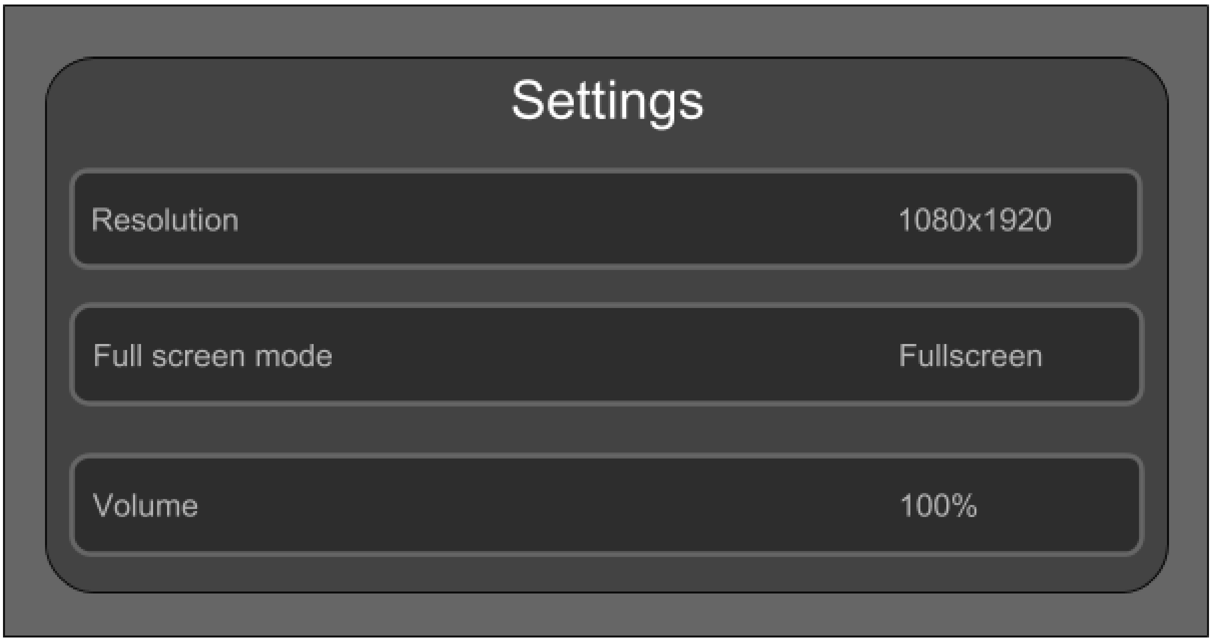

Main Menu
Save Game Selection
Settings
Save Game 1
Save Game 2
Save Game 3
In-game Overworld
Talk with Maria
Level
Level failed (Died)
Item Shop
Only after boss level
Level finished (Winning Screen)
End winning screen / credits

Wireframes

Main Menu



Settings

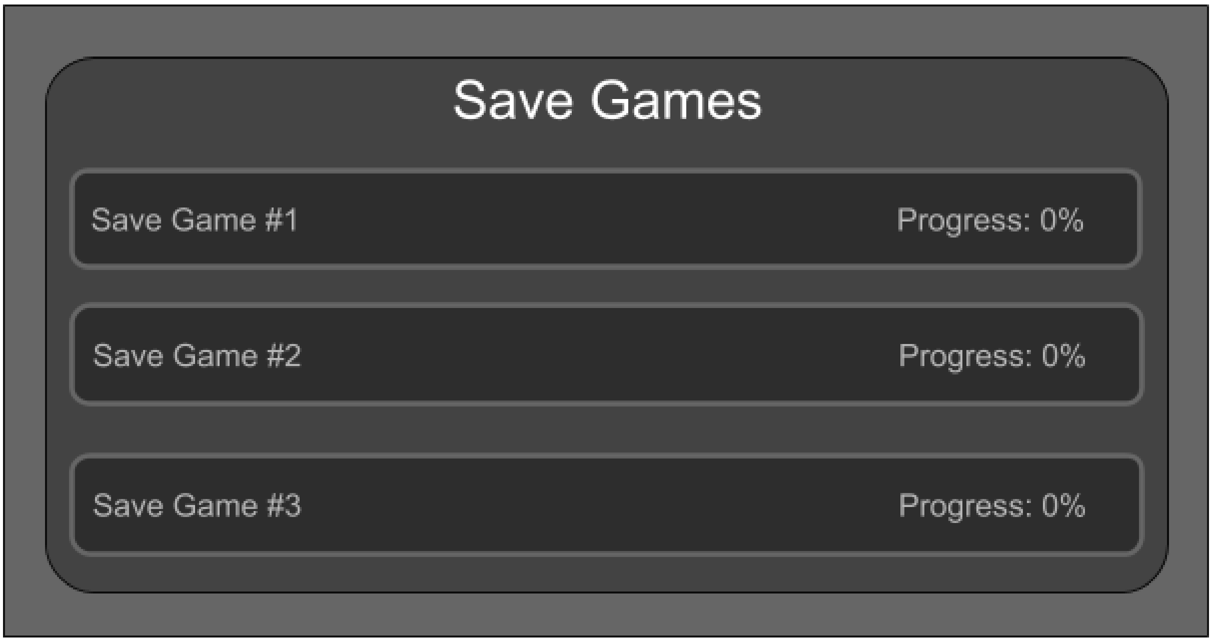

Settings

Resolution 1080x1920

Full screen mode Fullscreen

Volume 100%

Save Game Selection



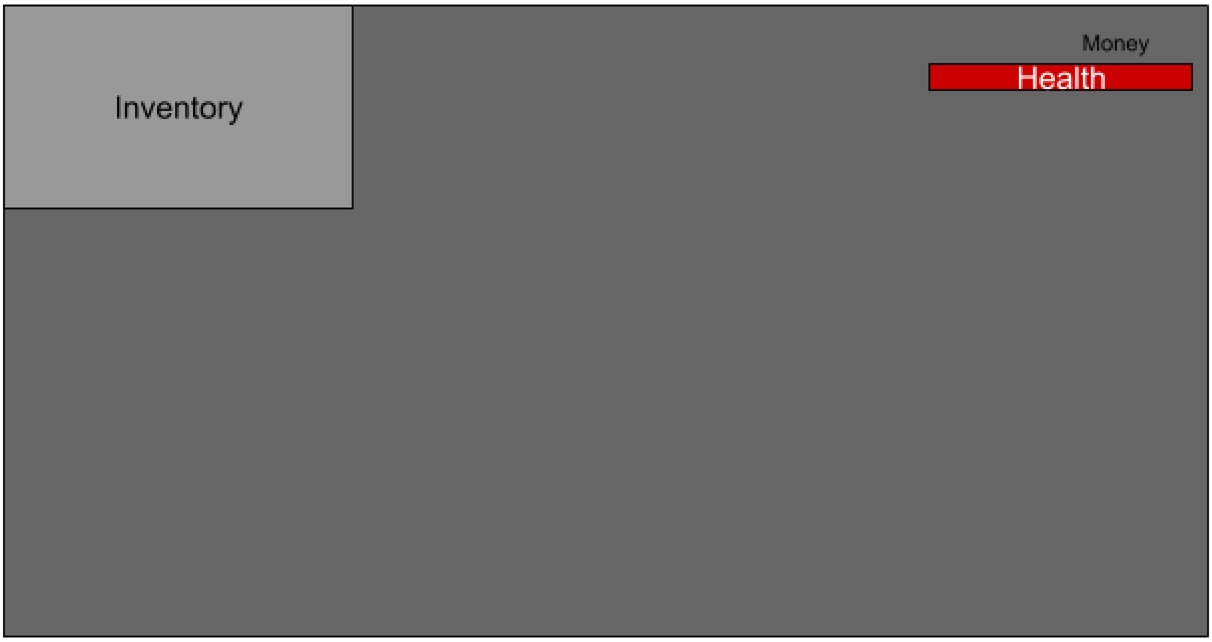
Save Games

Save Game #1 Progress: 0%

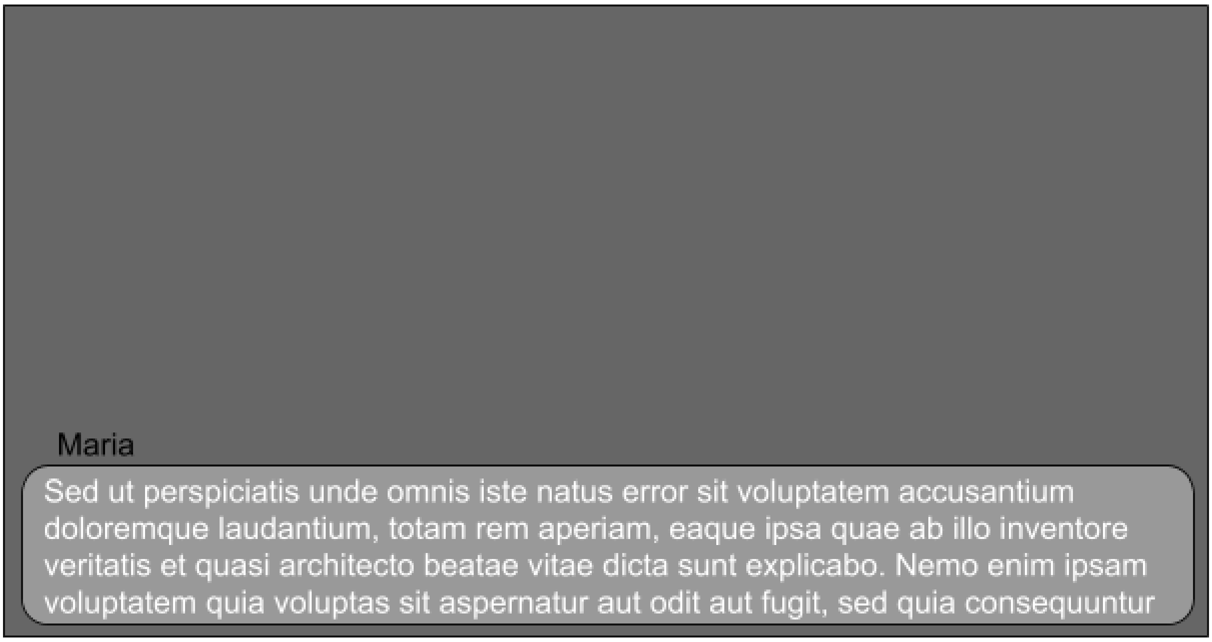
Save Game #2 Progress: 0%

Save Game #3 Progress: 0%

In-Game HUD

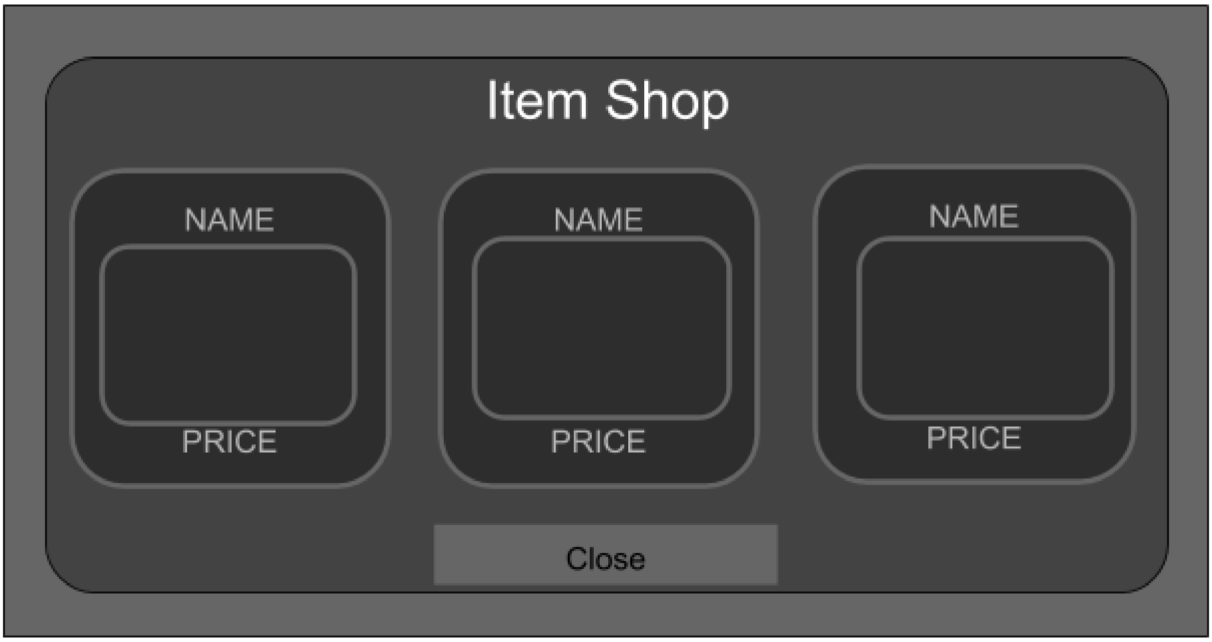


Dialog with Maria

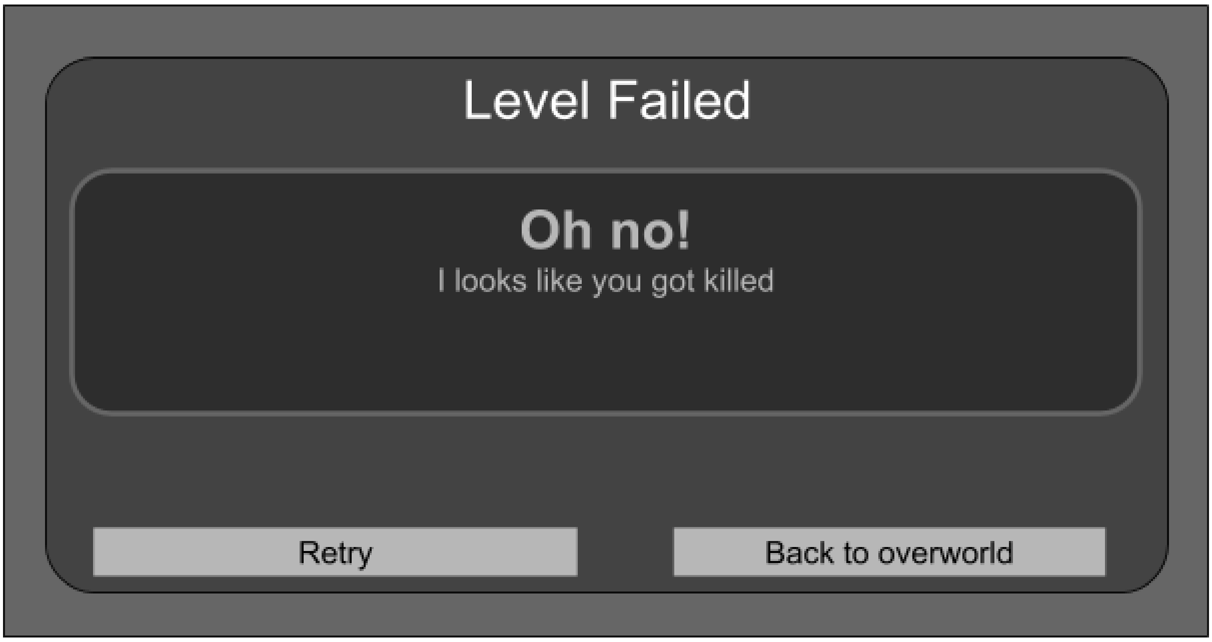


Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur

Item shop



Level Failed

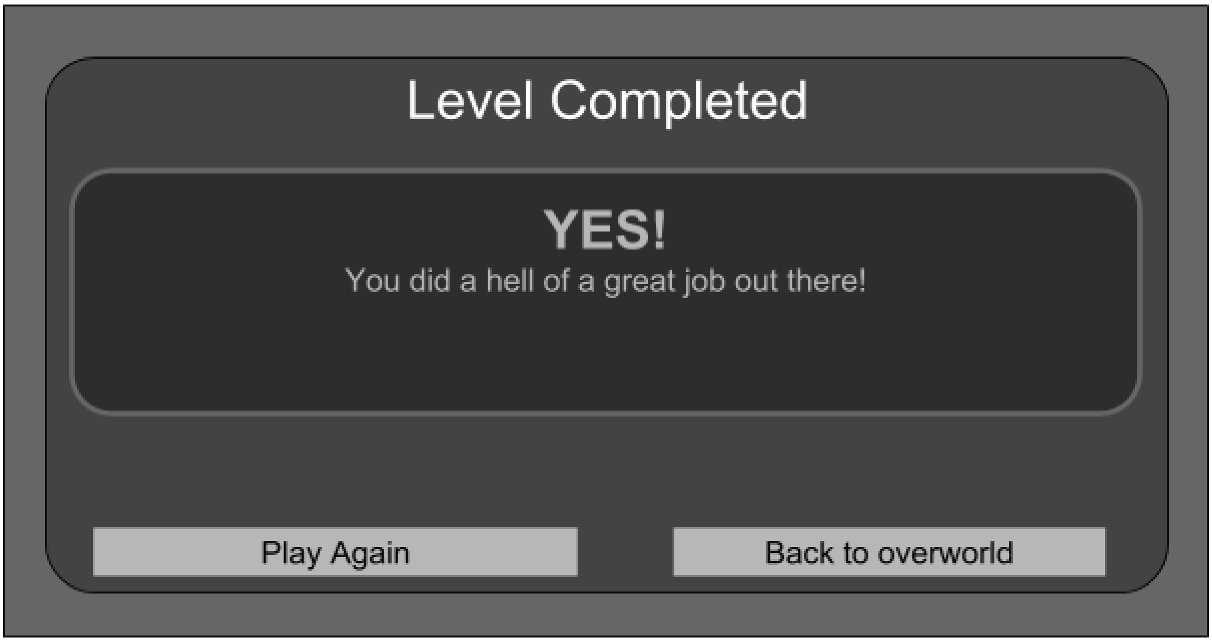


Level Failed

Oh no!

I looks like you got killed

Level Finished

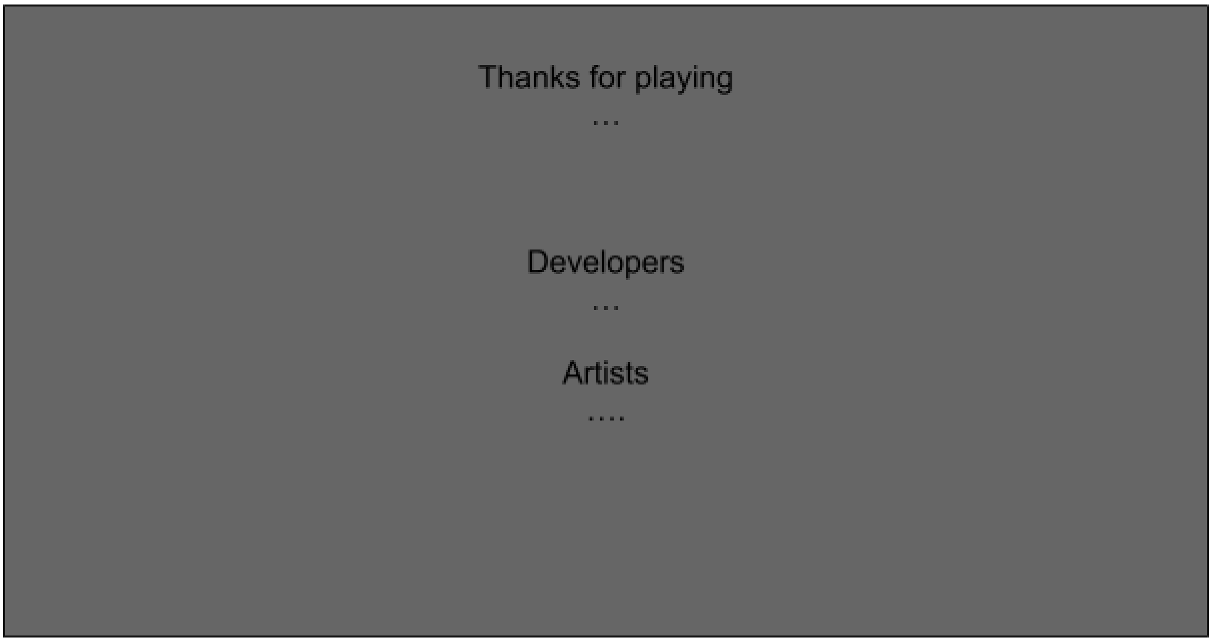


Level Completed

YES!

You did a hell of a great job out there!

End screen / Credits



Thanks for playing
…

Developers
…

Artists
….

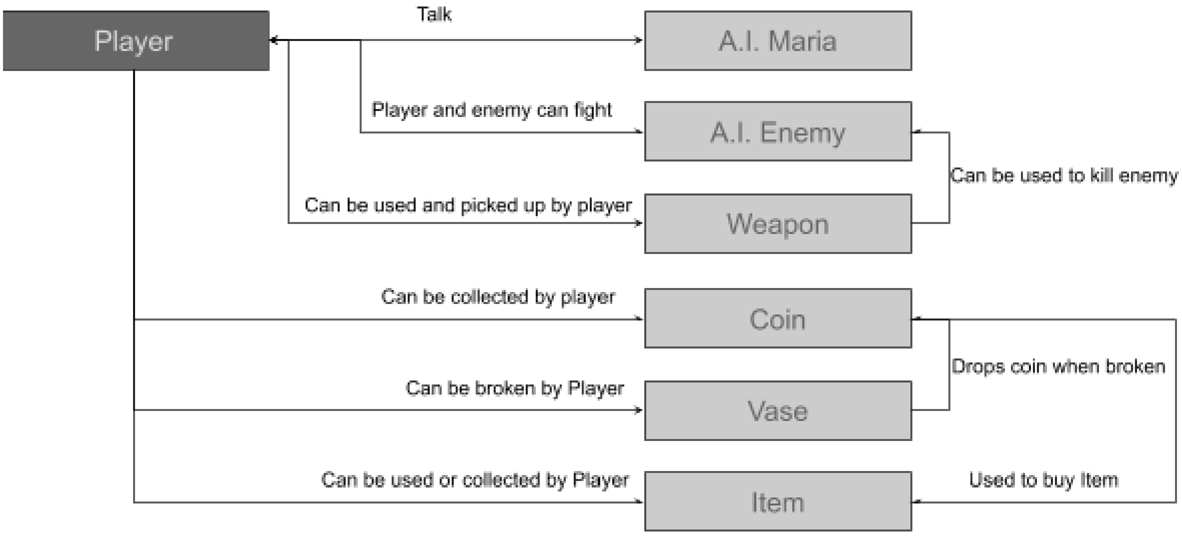
Interactive Objects

Name	**Type**	**Purpose / Description**
Coin	Item	Player can pick this up to collect money.
Health potion	Power Up	Heals the player
Shop	Shop	Used to buy upgrades or power ups for the player.
Maria	A.I.	The player can talk with Maria.
Enemy	A.I.	Can be killed by player or kill the player
Boss	A.I.	Can be killed by player or kill the player
Suicide fly	A.I.	Can be killed by player or kill the player
Dungeon door	Entrance	Used to enter levels and continue within a level.
Spike	Trap	Deals damage to the player.
Crate	Item	Can be used to hold a pressure plate.
Pressure plate	Button	Used to deactivate a deadly environment.

Behaviours

Type	**Purpose / Description**	**Expected Situations**
Weapon	Shoots or throwable	Can be used in combat (in levels only)
Items	Collectable	Can mostly be found inside levels but can also be used in overworld.
Power Ups	Give a boost to the player	Inside levels when the players health is low for example. The player could use a health power up.
A.I. Maria	Talks with player	Only inside overworld near the starting house
A.I. Enemy	Fights with player	Only inside levels where the player is in combat.
Entrance	Enter a level or continue	Expected in overworld and within levels.
Button	Deactive a certain trap.	Only expected within levels.

Object Relationships


Player
Talk
A.I. Maria
Player and enemy can fight
A.I. Enemy
Can be used to kill enemy
Can be used and picked up by player
Weapon
Can be collected by player
Coin
Drops coin when broken
Can be broken by Player
Vase
Can be used or collected by Player
Item
Used to buy Item

Core Game Mechanics

Mechanic	**Purpose / Description**
Lives	Keeping track of players health and make it possible to die.
Top down	The game is played top down.
Random level generation	Generating random levels so levels are never the same.
Buying	Buying items to become better/stronger.
4 directional movement	Let the player move around in 4 directions.
Teleporting (Entering level)	Throws the player inside a level.
Attacking	Make sure the player can kill the enemies.
A.I. Enemy System behavior	Enemies walking around and/or attacking the player.

Target

Platform	Epic Games store (Windows)
Recommended device	A computer or laptop with the following specs: - Processor: 2 GHZ dual core - Dedicated graphics card with 2GB vRAM - 10GB of free space - 4GB of RAM
Minimal device	A computer or laptop with the following specs: - Processor: 2 GHZ dual core - Dedicated graphics card with 1GB vRAM - 10GB of free space - 2GB of RAM
Engine	We'll use Unity. Thanks to unity's 2D advantages and tile system we can create levels easily also the LWRP is perfect for the the platform we're targeting.

Project rules

Folder structure

- xxxxx_rpg (Root directory)
  - From client (Files we got from client)
  - Research (Moodboards or other research files)
  - Builds (Here we collect our builds)
    - Windows (Builds for windows .zip files)
  - Assets (Here we collect our assets: Art, Sounds)
    - Art (Here we collect our sprites/art)
    - Sounds (Here we collect our sounds and our music)
  - Documentation (Here we collect our documentation for this project)
  - Production (Here we keep our source files for Photoshop, Unity, FL Studio)
    - FL Studio (This is our FL Studio project)
    - Photoshop (Here are our .psd files)
      - Workspace (Here we store our .PSD that we are working on)
      - Master (Here we store our finished .PSD's)
    - Unity (This is our Unity project **Sub folders should only contain production ready assets**)
      - Scripts (Here we store our scripts)
      - Assets (Here we store our assets)
        - Audio (Audio files)
          - Music (Audio music files)
          - FX (Audio FX files)
          - Other (Other Audio files)
        - Art (Here we store our art files)
          - Environment (Environment art files)
          - Items (Item art files)
          - UI (UI art files)
          - Characters (Character art files)
          - Other (Other art files)
      - Materials (Here we store our Unity materials)
        - Environment (Environment materials)
        - Items (Item materials)
        - UI (UI materials)
        - Characters (Character materials)
        - Other (Other materials)
      - Shaders (Here we store our Unity shaders)
        - Environment (Environment shaders)
        - Items (Item shaders)
        - UI (UI shaders)
        - Characters (Character shaders)
        - Other (Other shaders)
  - Delivered to client (Here we paste all our files that have been sent to the client)

File types

- .jpg (textures we don’t need transparency)
- .png (textures that use transparency)
- .psd (Photoshop files for creating and changing textures)
- .mp3 (Sounds)
- .exe (Windows builds)
- .zip (Compressed builds)
- .cs (C# scripts)

Naming

- Don’t use spaces we use under squares ex: police car.fbx should be police_car.fbx
- Textures should be named after their models ex: police_car_door.jpg
- Name everything logical
- Make sure to use the correct file extension
- Don’t forget the name your nodes right in Maya/Unity

Version control

- We use a GitHub repository with sourcetree as software.

Technical choices

Subject	**Explanation**
Rendering and View	Lightweight render pipeline since it is a pixel art 2D game. This will make sure the game works on low end pc's
2D top down	This is a requirement from the client.
Scene management	We'll use a scene per level because we need to randomly generate the level ones the player starts one. This is a bit easier if we do this in seperate scenes.
A.I.	Maria will be standing still on a spot near the main house of the player. Enemy A.I.'s will be randomly walking around in the room inside the levels. The game needs to be completed in 5 weeks so we can't make really advanced A.I.'s
Manager objects	We'll use multiple manager object for the player to keep track of he's inventory, health and money. transition system, level, audio etc.
Data storage	We'll only store save games on the PC of our players and nothing online. Because it is a single player game there is no need to save files online.
Unity's new UI system	We'll be using Unity's new UI system since it doesn't require programming to create a fairly nice UI. This can be faster done with the new UI system.
Player	The player can be controlled with mouse and keyboard or controller. Since split screen isn't a requirement from the client we don't want to loose to much time trying to add it.
4 Directional movement	We'll use 4 directional movement since it requires less art so will be faster done. And it is a pixel art game so it fits in perfectly.

Resources

Role	Tasks	Who	Critical role
Artist	Create the art listed on the assets list.	John (Contract)	Yes
Developer	Write the gameplay and game mechanics and integrate the art/sound assets in the game.	Menno (Contract)	Yes
Sound engineer	Create the sound effects and musics listed on the assets list.	Do (Freelance)	Yes
Project manager	Make sure everyone is motivated and does their jobs. Keep track of the deadlines and make sure we make them. Keep the client up2date.	Menno (Contract)	Yes

Boundaries

Boundary	Flexible
Budget 18.000 euro	No
Start date 25 February 2019	Yes
Deadline 29 March 2019	No
3 employees	yes

Costs

Role	Hourly costs	Software costs per hour	Total costs per hour	Hours	Total
Developer	€ 40	€ 1,10	€ 41,10	200	€ 8.220
Artist	€ 30	€ 0.20	€ 30,20	200	€ 6.040
Sound engineer	€ 30	€ 0,-	€ 30,-	100	€ 3.000
Total costs per hour for project:			**€ 101,30**		
Total sum costs for project:					**€ 17.260**

Software	Role	Costs (5 weeks)
Unity pro	Developer	€ 220
Photoshop	Artist	€ 40
FL Studio	Sound engineer	€ 0,- (Since he’s a freelancer)