MOBILE GAME DEVELOPMENT

FINAL PROJECT REPORT

TOPIC : Romp of Mazes (maze adventure game)

Logo

Description automatically generated



GAME DESIGN DOCUMENT (GDD) TEMPLATE

# **Game Name: Romp Of Mazes**

## **Genre:** 2D adventure game

**Game Elements:**

There are 2D game objects like the walls, Doors, Keys, Players, Slimes etc.

## **Player:** Only one player can play the game at once.

# **TECHNICAL SPECS**

# **Technical Form:**

There are 2D game objects like the walls, Doors, Keys, Players, Slimes etc.

**View:**

Romp of Mazes is a 2D game where the user has the **top view** of the characters and all the game objects like walls and doors etc.

**Platform:**

Currently the game is built to run on PCs with the required specifications, but it can also easily be updated to run on Android, iOS, and a Mac device.

**Language:**

C# is used in the scripts for making the player move in the instructed direction and to perform certain actions when all criteria are met.

**Device:** PC

# **GAME PLAY**

**Romp of Mazes** is a single player game where the user is in control of player motions and governs its next movements. It has complicated mazes one after other all separated by levels, once the player passes through a level the scene shifts to the next level and the player will have to go through multiple levels before they reach their end goal i.e., save the princess. The game has gameobjects like walls, doors, keys and slimes(enemies).

In this game the player has to avoid all the slimes, collect the keys, open the doors and make his way through the maze to reach the end goal and save the princess. When the player reaches near the princess and touches her, the game ends and the scene shift to the end scene which displays a text as “You Win” and there is an option to play the game again from the starting. If the user wants to quit in between i.e., before completing the game, they can do so by pressing the escape key in between.

**Game Play Outline**

* Opening the game application
* Selecting the options to play
* Game objects load
* Player Moves
* Collect Keys
* Open Doors
* Pass each Level
* Save The Princess
* End
* Play again

**Key Features**

Some key features of the game include:

* The goal i.e., to save the princess after all the struggles.
* Challenge i.e., the difficulty increases with every level increase.
* Even a slightest of collisions will make the user restart the level from the start and there are no options to restart the whole game so once the game has started users will have to complete the whole game before they can stop.

# **DESIGN DOCUMENT**

**Walls** are rigid game objects that limits players' motion. There are multiple walls distributed throughout the levels of the game(prefabs). All the walls connected together make the maze.

**Doors** are another rigid game object that stops the player from passing through one level to another. They are destroyed when a certain condition is met.

**Keys** are the game object that the users have to collect in all the levels of the game. As soon as the player touches a key it vanishes immediately. Only when the player touches all the keys in a level the door gets destroyed and the player can now pass through to the next level.

**Slimes** are the enemies in this game. There are multiple slimes in every level of the game, each given their own animation i.e., the path they constantly loop around. When the player touches any of the slime at any point of the game the player game object is destroyed, and the level is started again from the starting.

**Player** is the user. The user controls the motion of the player and decides on its next path to follow. The player is a rigid body and is supposed to collect all the keys in all the levels before completion of the game.

**Design Guidelines**

All the above-mentioned game objects are supposed to have a rigidbody and a collider attached to them because the game heavily relies on the touch in between all the components of the game.

The player is supposed to touch all the keys and is supposed to not touch any of the slimes at all before he reaches the end goal.

**Game Design Definitions**

The player maneuvers through the maze avoiding the slimes, collecting the keys and finally reaching the princess.

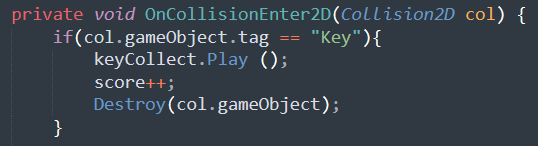
* The win state is defined as, when the player touches the princess, the scene shifts and the new scene now says You Win.
* There is no lose state as the game always restarts from the level start when the player collides with any of the slimes.
* The player moves from one level to another when he collects all the keys, opens the door and passes through it

**Game Flowchart**

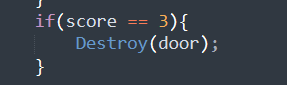
Diagram

Description automatically generated

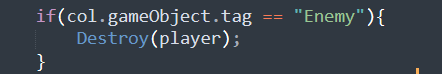
The script attached to the player game object has an on-collision function attached to it which governs the game flow of the whole game.

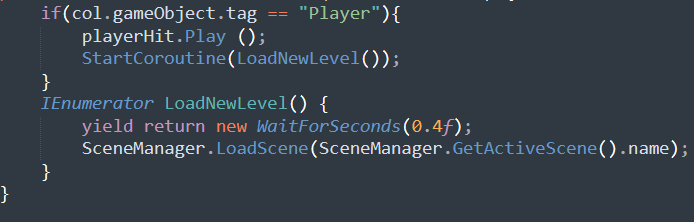


if the player touches the collider of the key, the score gets incremented and the touched key gets destroyed immediately and a sound is played.

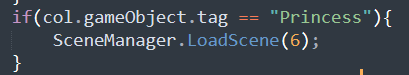


If all the keys are collected and the score is incremented to 3, the door also gets destroyed.

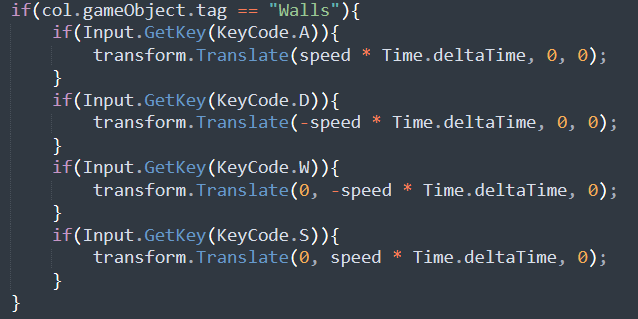




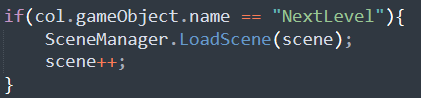
If the player touches the slime that is tagged as enemy the player gameobject is destroyed and the current scene is loaded again from the start after waiting for 0.4 seconds.



When the player touches the princess, the last scene is loaded that displays a text as YOU Win.



If the player touches the wall, he is repelled by the same force it struck the wall.



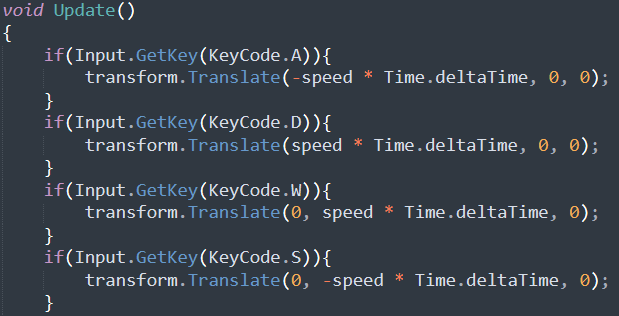
If the player passes through the door the next scene is loaded and a new level starts.

Graphical user interface, text

Description automatically generated

If user wants to quit, they can do so by pressing the Escape key which displays a quit to home screen scene.

**Player Definition**



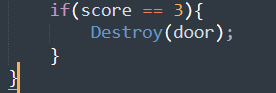
The player moves in the specified direction upon the correct click of the key. A,D,W,S are the key codes that define the movement of the player game object.

The player has other properties like:



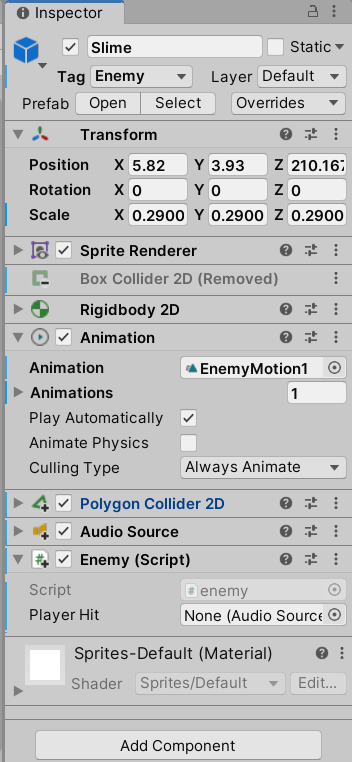
* A tag name of player, which is used in the code to relate with the game objects.
* Unity’s inbuilt feature to add a sprite renderer to any new sprite that is added to the game.
* Box collider, that is used in collision detection and is important for the flow of the game.
* Rigid body gives the player object physical properties similar to that of an actual real-life object.
* Audio source it is triggered when the player collides with any key or with any slime.
* Player Move is the script attached to it that defines the motion of the player.

### **Player Definitions**

* + 

If the player touches all the 3 keys in one level the door is destroyed, and the player can now pass through to the next level.

### **Player Properties**



Along with all the normal properties that a game object should have like rigid body and collider, The slime game object is also attached with an animation tag. This animation tag has a defined animation attached to it. This animation works in a loop and alters the slime position within the game at every instance.

# **Player Rewards (power-ups and pick-ups)**

# Player has to pick up 3 keys in each level avoiding slimes in its way to unlock the door to the next level.

**User Interface (UI)**

The UI of this game is very basic. In this game we have used the ‘A’, ‘D’ keys of a keyboard so that the player can move right and left respectively and ‘W’, ‘S’ keys so that the player can move in upward and downward directions respectively too. Furthermore, we have added buttons at the start and end scenes of the game from where the user can decide whether to play or not. Once after the player has started the game they can still stop or pause the game in between by pressing the escape key.

**Future Scope**

The game can be further improved by adding more levels to the game and refining the whole UI part of the game and giving the user more control.