**Jungle Angry Birds System Engineering Document**

Contents

[Game Mechanics 2](#_Toc510636297)

[Game Scenes 2](#_Toc510636298)

[Game Flow 2](#_Toc510636299)

[Classes and scripts 3](#_Toc510636300)

[Classes 3](#_Toc510636301)

[Methods: 3](#_Toc510636302)

[Flow Chart 5](#_Toc510636303)

# Game Mechanics

## Game Scenes

This game will have three basic scene schemas, the main menu, the level scenes and the win and lose screens.

* Menu Screen: here the game will be started, being able to begin the first level or quit the game
* Level scenes: Will be comprised of the player, the different enemies, the different kind of rewards available to the player and the health recovery items. All in a platform dungeon. The health and score of the player will always be displayed
* Win/Lose Screen: Here the game will notice the player If it either won or lost the game and show his score. Buttons to the main menu and quit are available.

## Game Flow

1. Menu scene will lead to first level
2. Losing in any level will lead to the lose screen
3. Finishing the last level will lead to the win screen.
4. The losing or finishing screen will be able to lead back to the menu.
5. The lose, winning and menu scenes can send quit requests.

# Classes and scripts

## Classes

In these games several objects need to be handled

* A level manager to keep the scene sequence in order
* Camera control: to keep the camera focused on the player
* Enemy: The enemies will deal damage to the player and push him away. They will move in a set route.
* Healing: these are objects that restore health to the player
* Reward: these objects increase the player score
* LoseCollider: This is set at the bottom of the levels in order to make the player lose when falling.
* Moving Plat: some platforms are able to move
* PlayerController: This will be the player, a rigid body that is able to walk on platforms, has health and score. Can jump and is controlled with the arrows.

CameraControl methods:

* Start() in here the position is located in the player
* LateUpdate() will follow the player around

Enemy has as a property the damage it does to the player and the target that limits its route. Other properties that help it move.

## Methods:

* Start() it locates the start position of the enemy and saves it in a variable
* Update() here is moves the object to the target position and flips the animation when it changes direction.
* Flip() transform the scale to the negative of the current.

Healing only has the amount it heals as a property

Reward only has the amount it increases the score as a property

Level Manager methods:

* loadLevel(name) it receives the name of the scene to be loaded and loads it
* QuitRequest() sends a quit request to the game
* LoadNextLevel() it loads the next level on the build index

LoseCollider methods:

* OnTriggerEnter2D(Collision) calls the level manager and makes it load the lose scene

MovingPlat methods:

* Start() the starting position is stored
* Update() The object is moved to the target position using the MoveUp Boolean, transforming the position of the current object

PlayerController, has the score and health properties, its methods are:

* Start() here it saves the rigidbody properties, the renderer and the animator.
* FixedUpdate() it sets the animations to the buttons pressed and check if the player is standing on Ground.
* OnCollisionEnter2D(collision): checks if the player is grounded, checks if the colliding object is an enemy and takes damage and bounces.
* OnCollisionExit2D(Collision): sets to false the animation flags of isOnGround and Idle
* OnCollisionStay2D(Collision): activates the isOnGround animation flag
* OnTriggerEnter2D(Collider): destroy the object it collides with and depending if it is a reward or a health item, it increases the health or score. The health can never go above 100 though.

# Flow Chart

menu

Level Scene

Lose Screen

Click=star

win

Win Screen

no

si