CSE 344: SOFTWARE ENGINEERING
YEDITEPE UNIVERSITY
TERM PROJECT
2021 SPRING

RUN BOYS
DESIGN REPORT

AHMET DEMIR
GAYE EMINE CELIK
GURKAN BEDEL
UGUR ORNEK
BURAK GÜNDÜZ
YAVUZ KARABIYIK

CONTENTS

1. Introduction
1.1 Aim of the Document.....3
1.2 Aim of the System.....3
1.3 Structure of the Document.....4

2. Detailed Design Class Diagram.....5
2.1 Game's Screen - Assets.....6

3. Dynamic Models
3.1 Sequence Diagrams.....X
3.2 State Diagrams.....X
3.3 Activity Diagrams.....X

4. Software Architecture
4.1 UML Package Diagram X
4.2 UML Component Diagram.....X

5. Entity Relationship Diagram.....X

6. Glossary & References.....X

1. INTRODUCTION

1.1 Aim of the Document

The aim of this document is to ensure detailed information about the structure of the 2D game, such as a mobile game called "Run Boys". It examines the class structure, system-wide sequences of action, the relationship between classes, the properties of the software in question, and all user functions through different sets of diagrams.

1.2 Aim of the System

"Run Boys" is a 2D pixel art games like game that has the pure addictiveness of today's pc games and the soul of old school arcade games. The aim of system is to amuse users with a game that is fun and challenging gameplay. Therefore, a domain analysis was done among similar games for the purposes of avoid some common mistakes and inherit fun features from them. The game has similar dynamics to the Mario game that we encountered before. Therefore, the player can easily adapt to the game and participate in the competitive system.

There are 2 different game modes in our game. The user can easily change the background of the game. Since the player changes the background and plays, he will adapt to the game within a theme he likes. We designed the leaderboard to include the player in a competitive system in our game. Along with this scoreboard, each player must write their name at the end of the game. The players who write their names are included in the leaderboard and thus join the competitive system. The list will change as the number of players increases. The main problem in the game is that more than one person cannot play online at the same time.

One of the biggest problems with pixel art games is the lack of story. No matter how good mechanics you have in pixel games, the lack of any story in these games can affect the tastes of the players. We thought that the leaderboard should be included in the game in order to reduce this and make the game even more fun. Thus, although the game has no story, it will have a mechanics based on competition. Thus, the players will be competing with each other. Besides, "how to play?" we saw that there is no page. In order to fix this problem, the game "How to play?" We decided to add the page. Thus, those who want to play will be able to adapt to the game very quick.

1.3 Structure of the Document

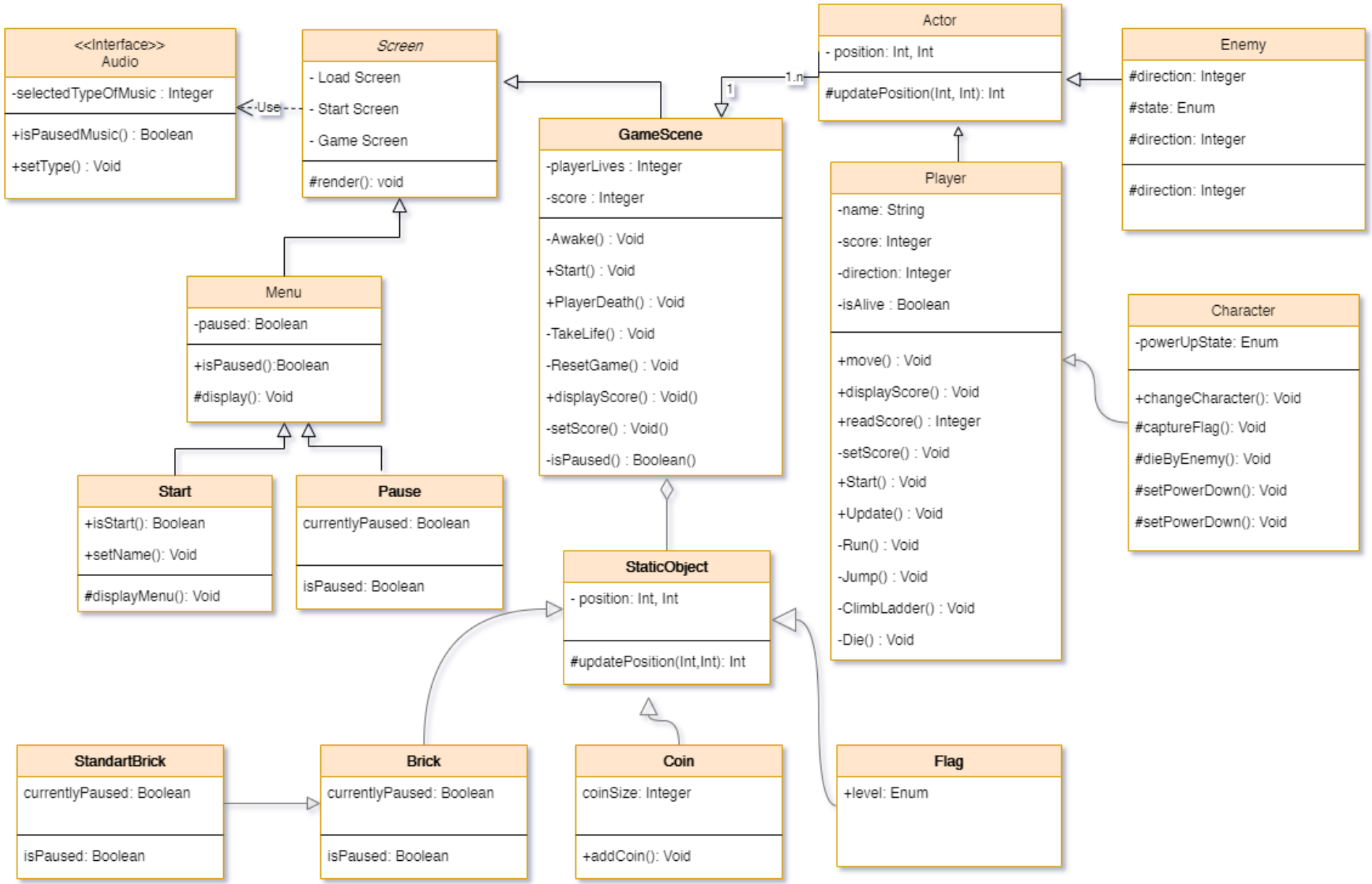
In this document, we aim to make the reader acquainted to our game, “Run Boys”. One can look at this document as three parts, Introduction, Diagrams and Run Boys and References.

The aim of the Introduction part of the document, is to make the reader understand what type of game “Run Boys” is and how it acts. We tried to explain how the game works on functional level and on motivational level. This section also provides explanations for how and why of our goals when we made the planning of the game, as well as the making of the game.

As the reader gets more acquainted with our game, one can move on with the diagrams of our project, which consists of sequence, activity, class and Entity-Relationship diagrams, as well as UML Package and UML Component diagrams. The goal of these diagrams is to show the reader the inside of our game without overwhelming with blocks of codes. The diagrams are there to show clearly how something happens and what the program consists of.

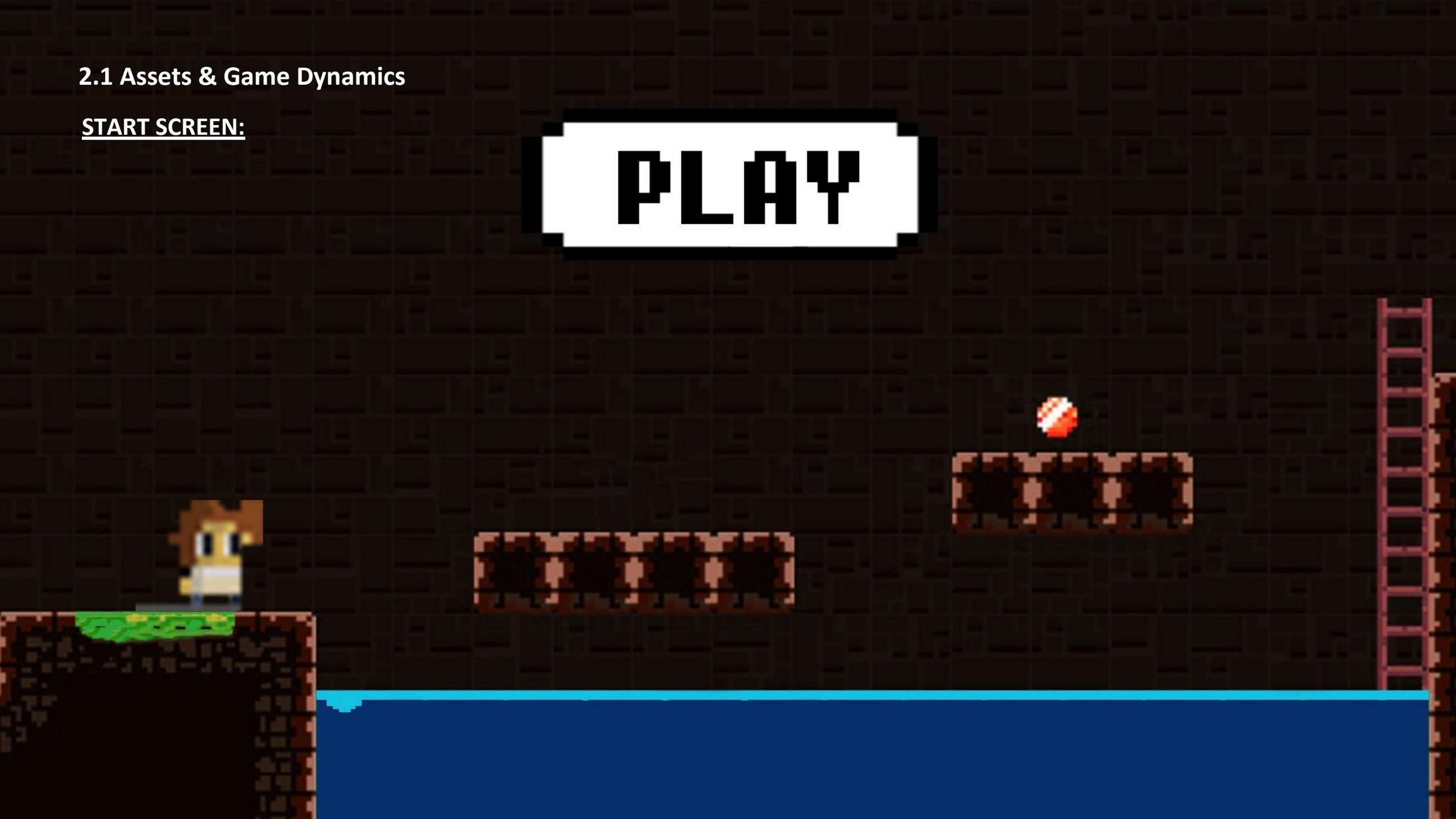
At the end of the document, one can find Run Boys&References for any unfamiliar term they found in the document or the sources of any quoted or referenced parts.

2. DETAILED DESIGN CLASS DIAGRAM

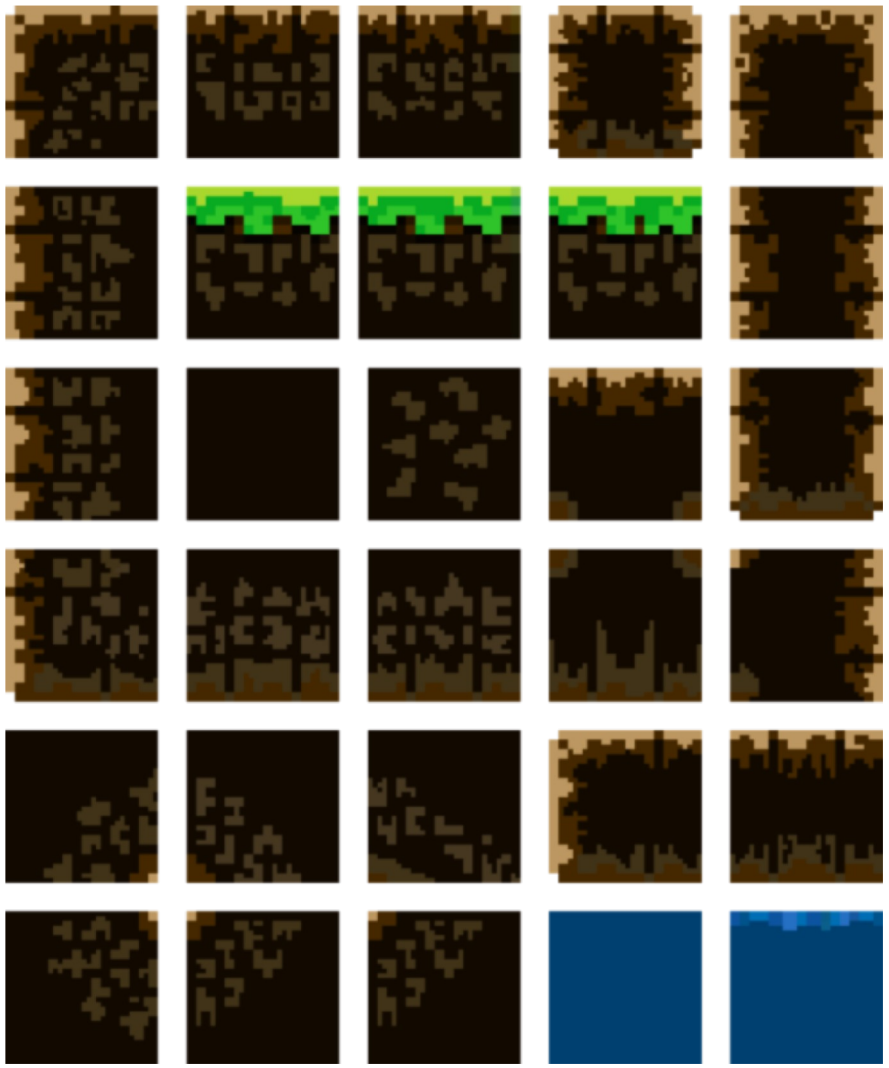


2.1 Assets & Game Dynamics

START SCREEN:



Background (Rock/Grass/Water Design):



Water

Rock

Grass

Characters:

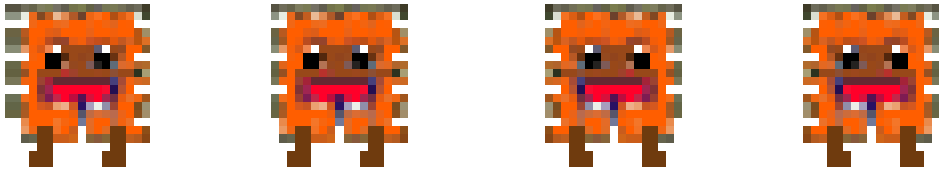
Character Example 1

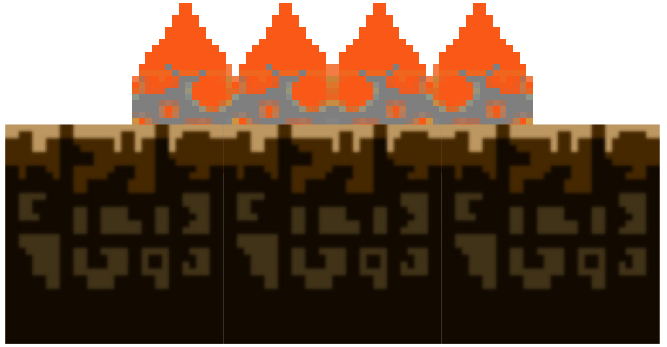
Character Example 2

Character Example 3

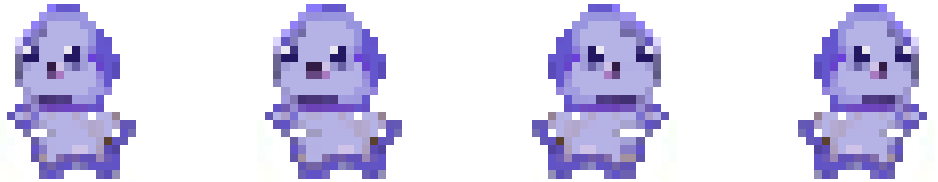
ENEMY and SPIKES:

Enemy Example 1:





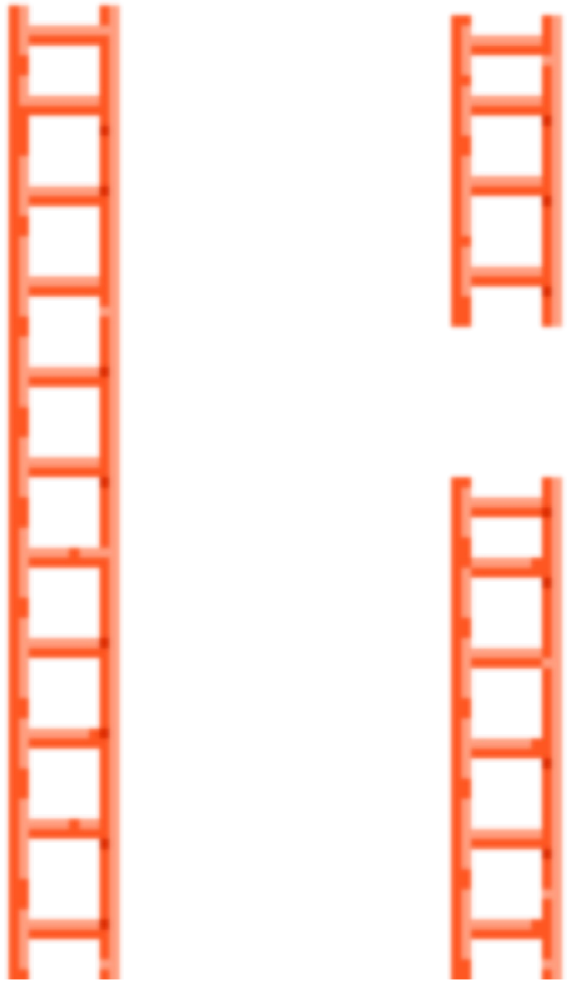
Enemy Example 2:



Spikes

COIN and LADDER:

Coin



Ladder


Game’s Background: Ex1

Game’s Background: Ex2

Game’s Dynamic：Ex1

Game Over Screen:

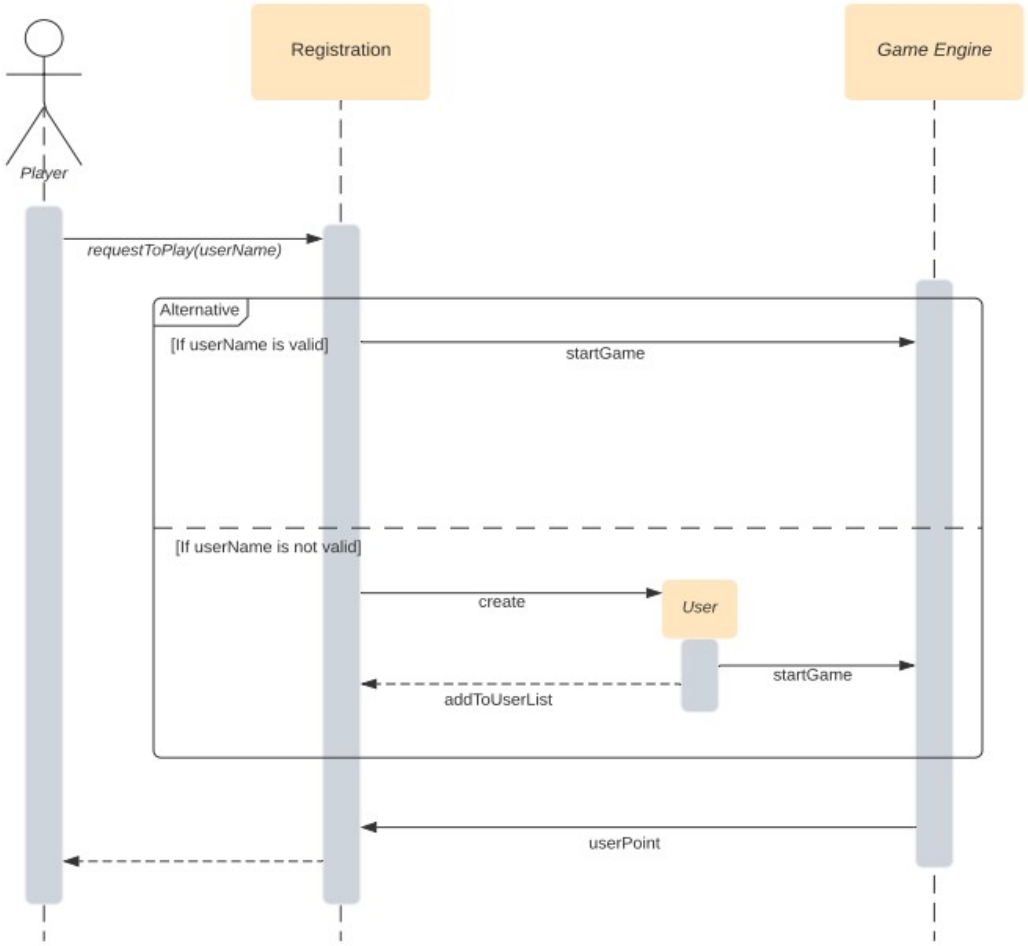
When you Win the game, You will this screen on game

FONT: Diary of an 8-bit mage

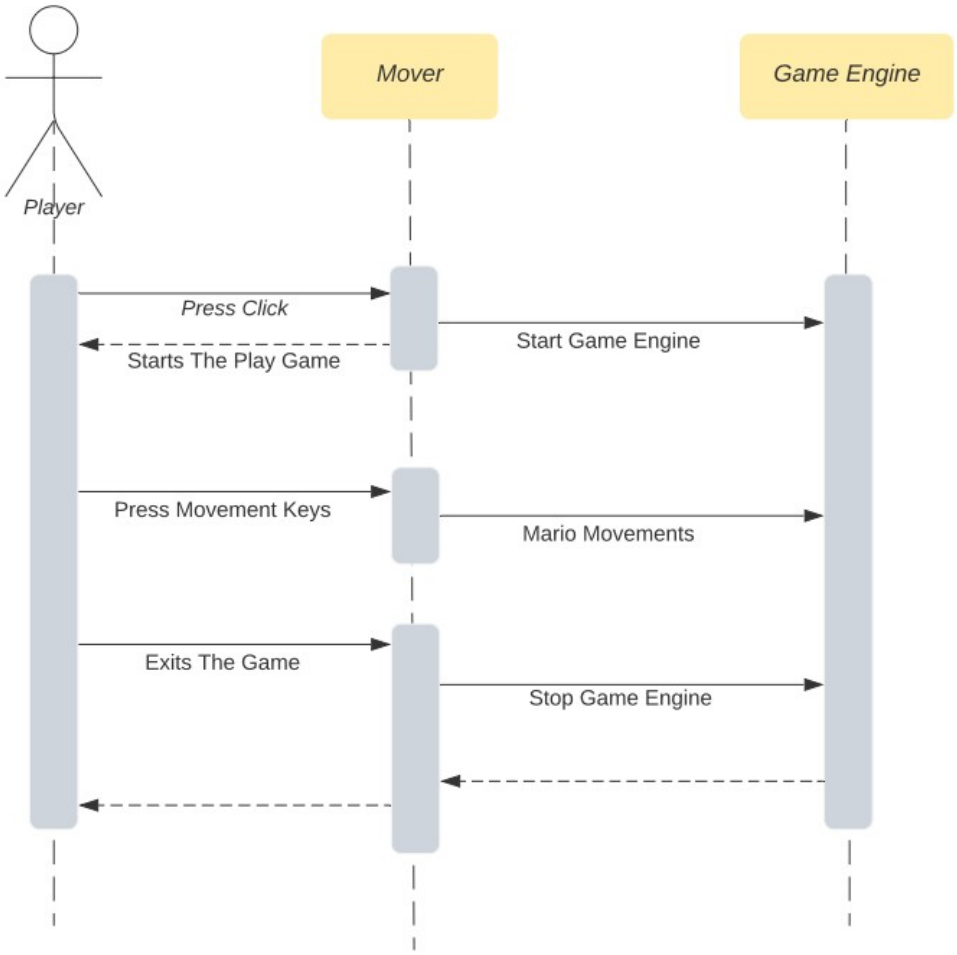
3. DYNAMIC MODELS

3.1 Sequence Diagrams

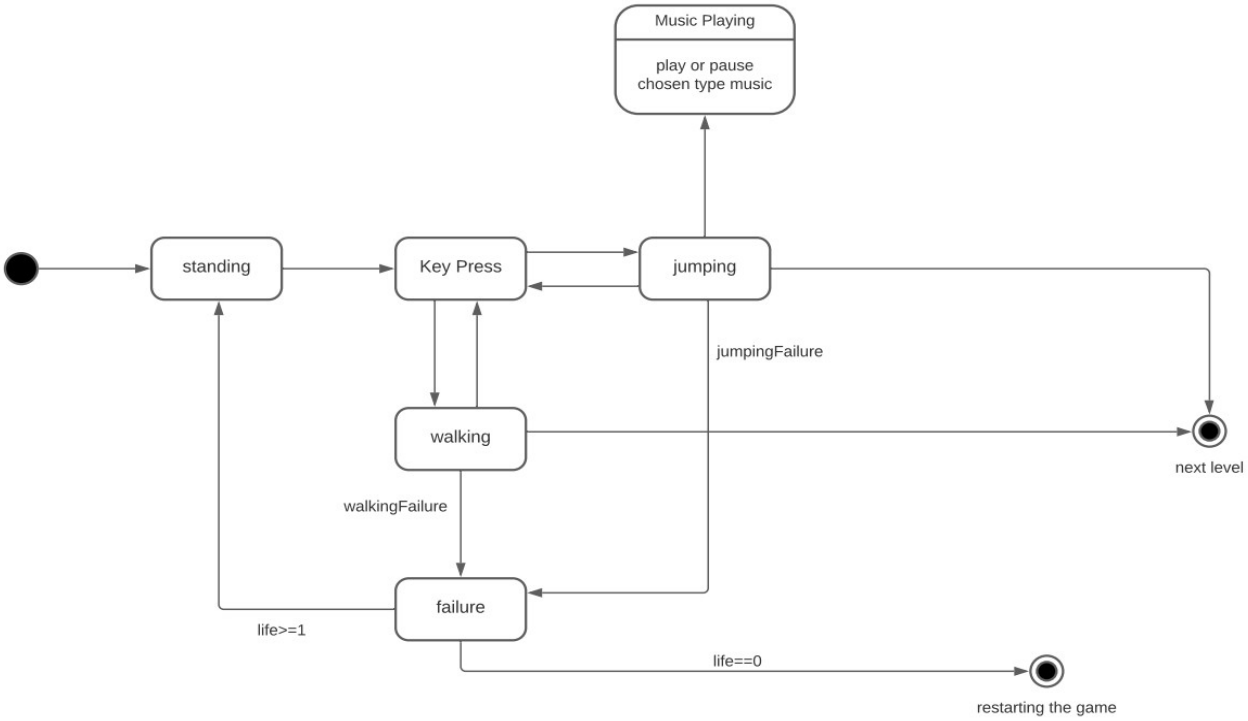
- Registration of Game



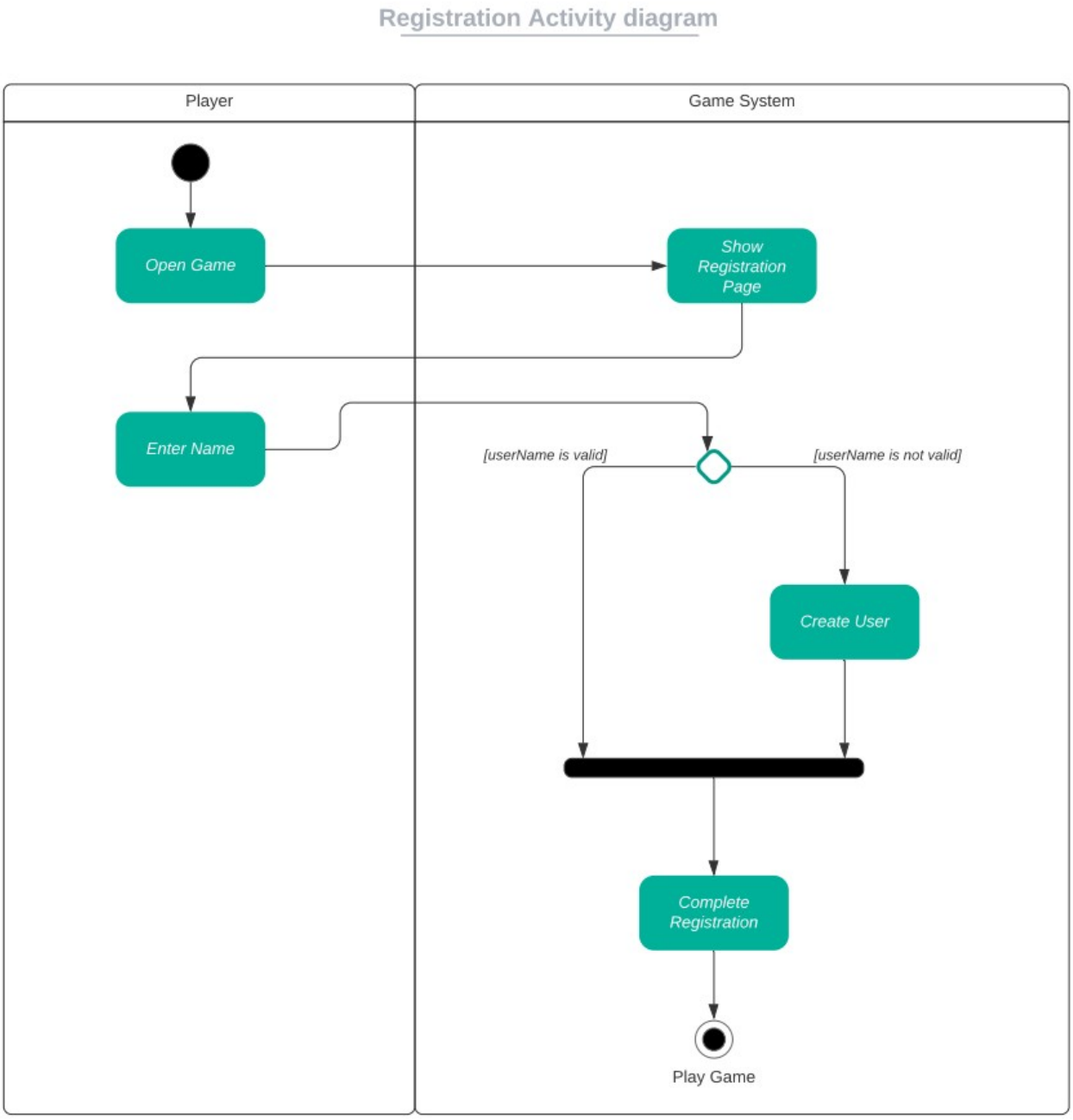
Sequence diagram - Actions



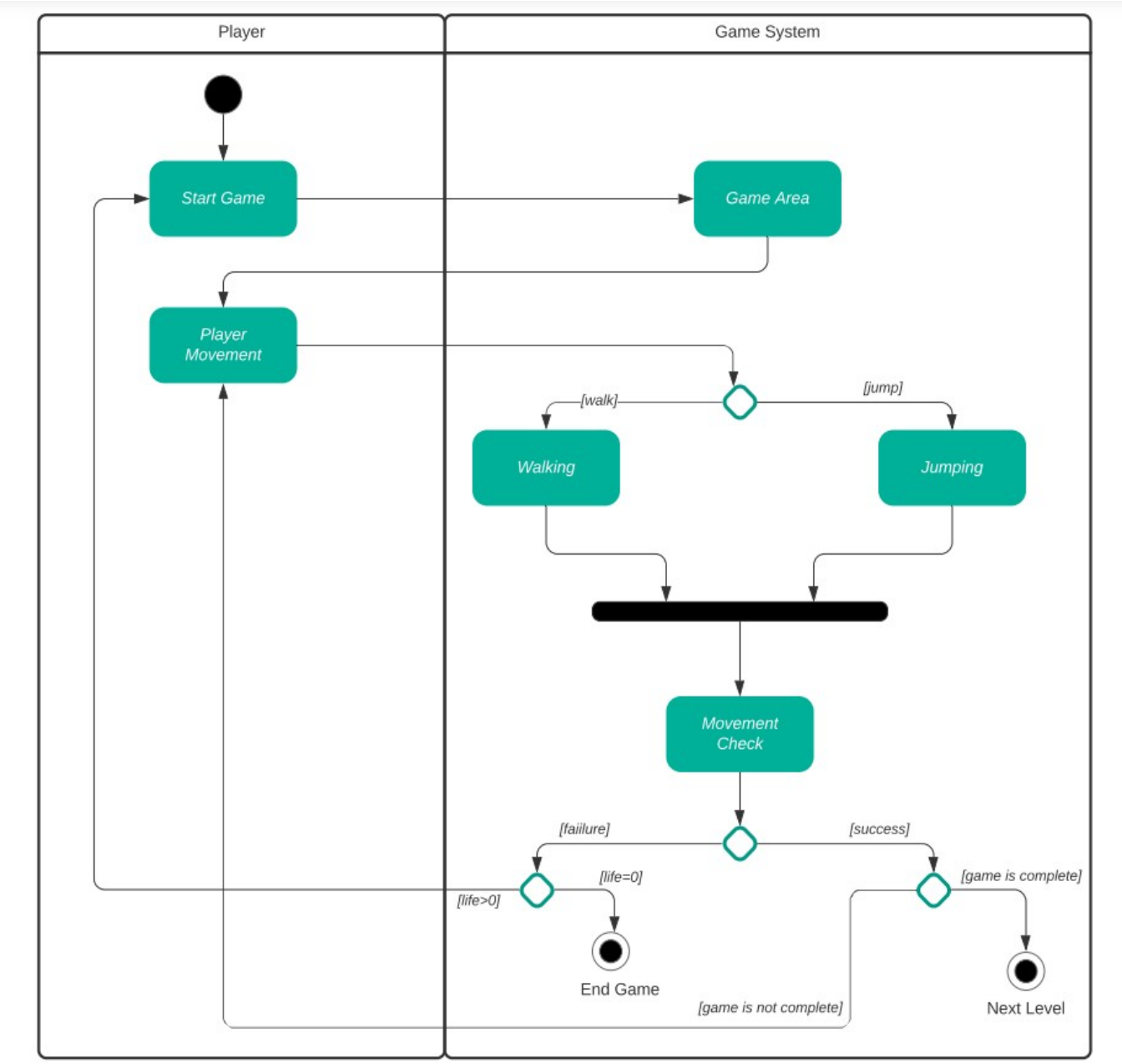
3.2 State Diagram



3.3 Activity Diagrams

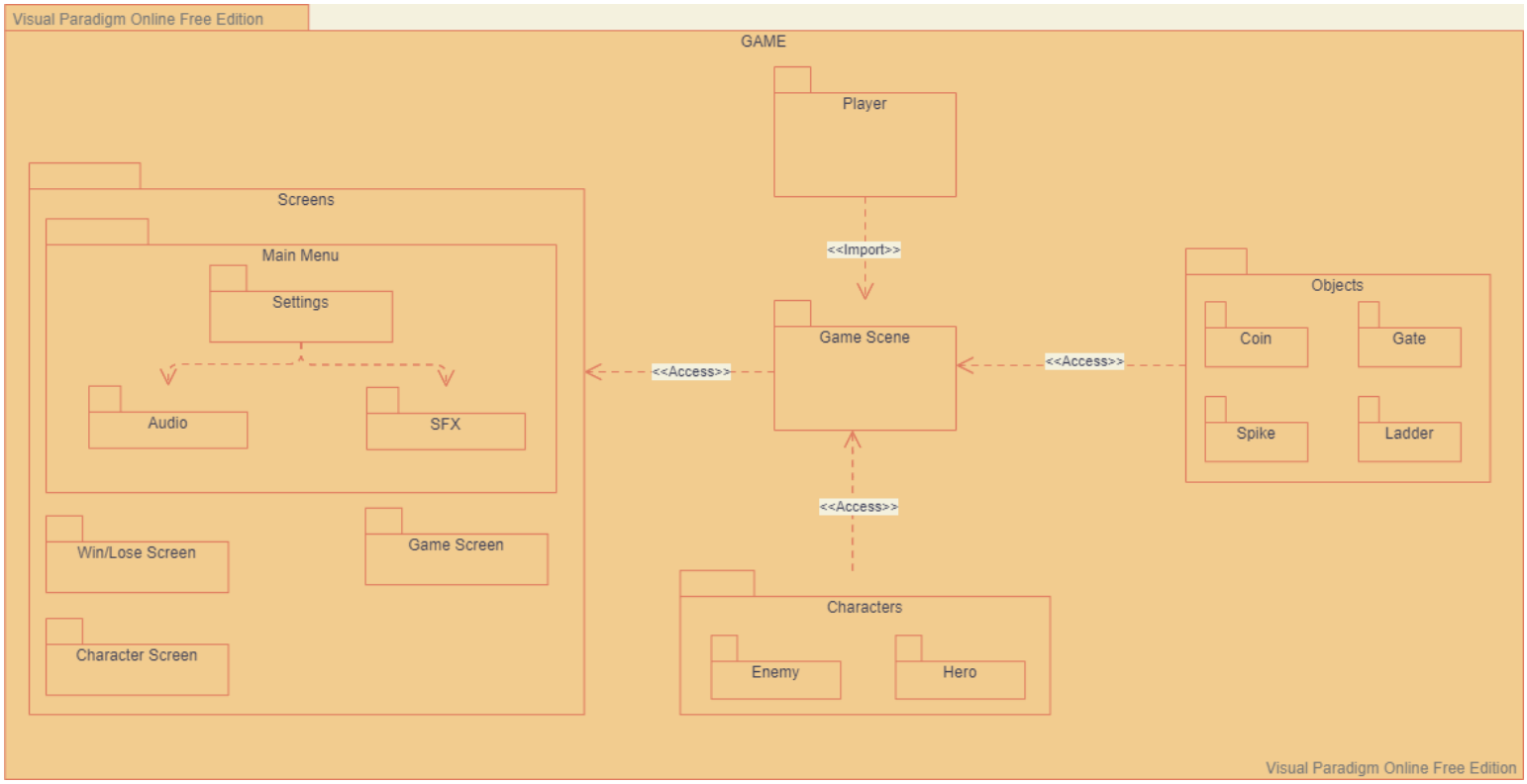


Game Engine Activity Diagram

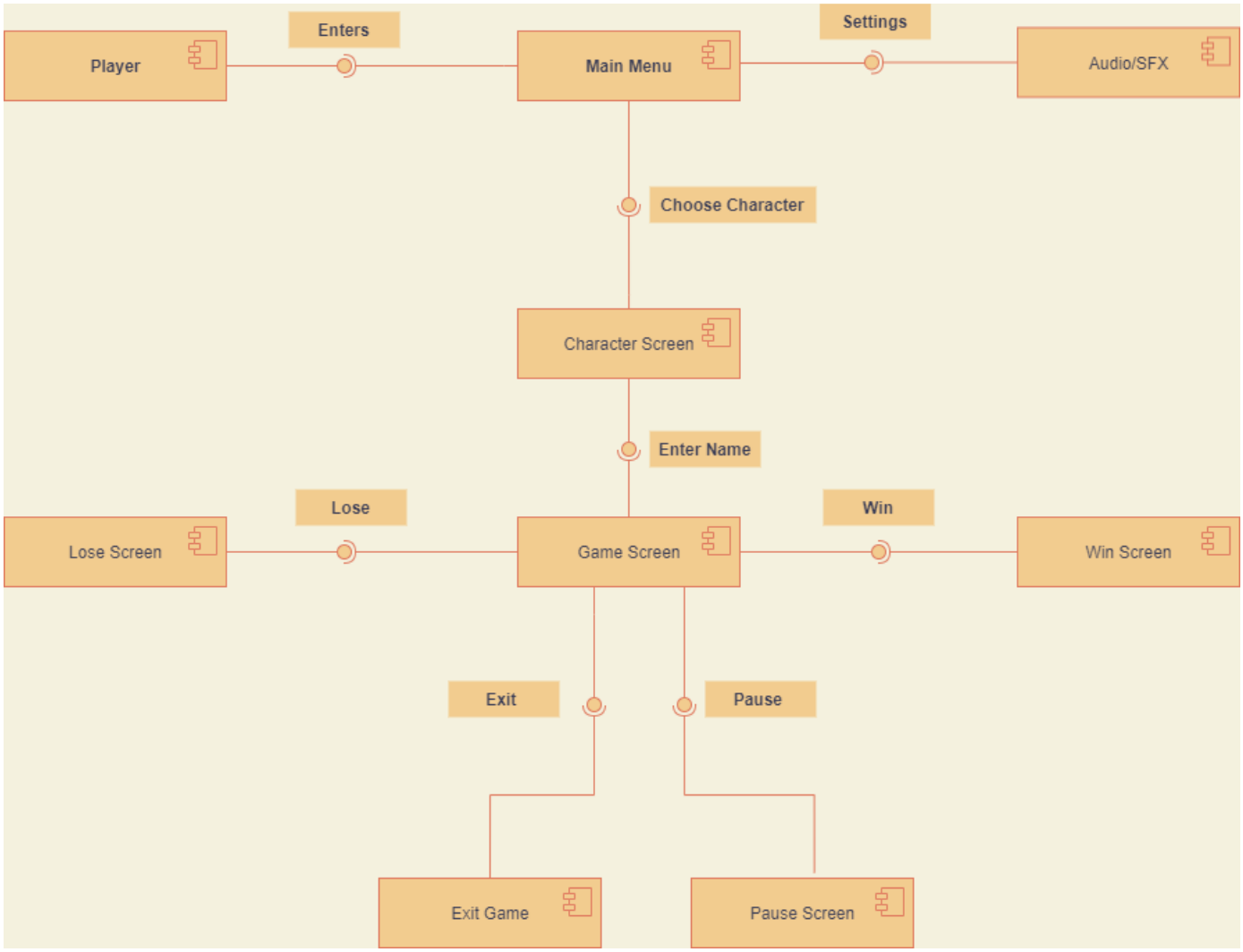


4. SOFTWARE ARCHITECTURE

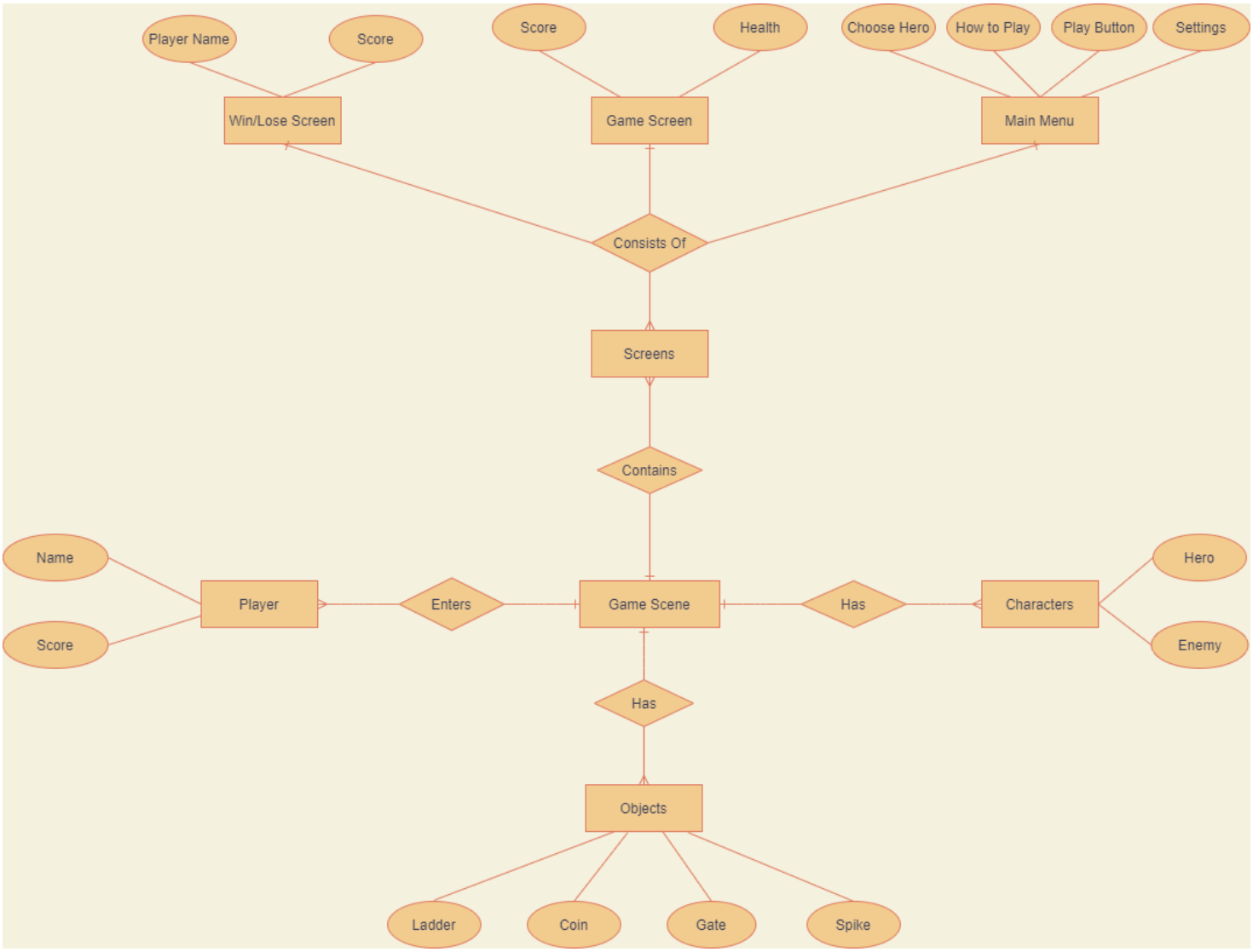
4.1 UML Package Diagram



4.2 UML Component Diagram



5. ENTITY RELATIONSHIP DIAGRAM



6. RUN BOYS & REFERENCES

UML: Unified Modeling Language

Arcade: Combination of old school video games

2D: Two dimensional

Pixel: Pixel Art Game