BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CS 319

Object Oriented Software Engineering Project

e-Bola

Analysis Report

Group 2B

Can AVCI

Deniz SİPAHİOĞLU

Ergün Batuhan KAYNAK

Esra Nur AYAZ

Course Instructor: Bora Güngören

October 7, 2017

Table of Contents

1. Introduction

e-Bola is a real time strategy - puzzle game, which consists of 4-5 levels. The main idea of the project is improving the main character's attributes with items found around the map, and by solving puzzles, and becoming powerful enough to defeat the boss, the deadly ebola virus.

The game starts with a cell membrane, that doesn't have any organelles in it. In the first level, the player wanders around to improve the cell accordingly with organelles, and proceeds to the second level. The cell solves some puzzles to get the keys of locked treasure chests. Here, the cell reproduces, and becomes 2 cells. These two cells defeat others to gain nucleotides, which are the currency in the game. In the third level, there is a timeskip and the two cells make a complete organism. In this level, the organism collects more nucleotides by battling small bacteria, and solving some puzzles. After a while, the organism gathers enough nucleotides to mutate itself, and to face the deadly ebola virus. In the last level, the organism has to defeat the virus by surviving a fierce battle, and after defeating it, the organism continues its life, peacefully.

2. Overview

2.1. **Gameplay and controls**

The player will use the arrow keys in the keyboard to start the game, move around the map, access the menu, pause the game and resume the game. The player will have to use the space key to interact with the interactable entitites and to attack the enemies. An example gameplay:

"You start the game as Celly, a little, young "cell", who only has a cell membrane! In order to defeat the evil bacteria and virus, he must evolve.

The game starts in a room, where there's nothing to be seen. There are portals to other rooms, but you don't know where they lead, of what will be waiting for you... You go to the room at the right first, and see that there are some organelles waiting for you. You quickly get a mitochondria, knowing that you will need it later on, for your stamina. You realise that getting some ribosomes wouldn't be bad at all for your vitality. You take these organelles, and continue to the room below. On that room, there is a giant treasure chest. You go near it, but it's locked. You realise that you need a key for the chest to open, and you realise that you need to find a key. You go to the room on the left, and find a key, waiting to be picked up. You pick the key, and go to the room with the treasure chest. The key fits the keyhole perfectly, and the chest opens. In the chest, there are 3 possibilities to choose from: a greater mitochondria, a ribosome, or some nucleotides. You will take the one you choose, and the other 2 will be destroyed. The ribosomes are the same with the ones you have, so you pass on the ribosomes. You already have a mitochondria, but this mitochondria is better than the

one you have. Nucleotides are used to evolve, and to improve yourself furthermore, so you take the nucleotides. The mitochondria and ribosome are destroyed."

2.2. **Levelling**

There are 5 levels of e-Bola, and in each level, the game gets harder. In the first level, there are no enemies. The number of enemies increase throughout the game.

2.3. **List of Entities**

2.3.1. **Tiles**

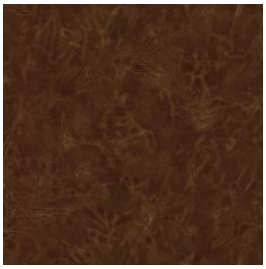
The game will take place at different environments, such as grass, rock and water. Each environment will have different effects on Celly.

2.3.1.1. **Grass**

When Celly moves in grass, his movement speed increases.

2.3.1.2. **Rock**

When Celly moves in rock, his movement and therefore dodging skills decrease.



2.3.1.3. **Water**

When Celly is in a body of water, his vitality slowly increases.

2.3.1.4. **Neutral**

Neutral tiles are the default tiles in the game, and have no effect on Celly.

2.3.2. **Interactable**

There are 3 types of interactables in the game. The chests, keys and portals. The player can use keys to open treasure chests, and portals to proceed to the next room.

2.3.2.1. **Chest**

The treasure chests might contain nucleotides, organelles or even enemies! Some of them require a specific key, and some of them don't.

2.3.2.2. **Key**

There is a specific key for the each one of the treasure chests, that require a key.

2.3.2.3. **Portal**

The portals are used to proceed to the next room. It is enough for Celly to run through the portals.

2.3.3. **Alive**

2.3.3.1. **Celly**

Celly is the name of the main character of e-Bola.

2.3.3.2. **Enemies**

There are 2 different virus that try to attack Celly. Some of them use ranged weapons and the others chase Celly, and try to kill him to steal his organelles.

2.3.3.3. **EBOLA**

EBOLA is also a virus, but is a special one. It is the final boss in the game, and uses ranged weapons to attack Celly.

3. Requirement Specification

3.1. Function Requirements

3.1.1. Main Menu

Ebola has a main menu that can be seen at the first initialization of the game. Following options can be found in main menu:

- Start game to initialize new game,
- Settings to customize the game,
- Save/Load the game.
- View Help
- Credits
- Exit

3.1.2. In-Game Menu

While playing the game there is another menu player can use. This menu can be reached with ESC button on the keyboard if the game is running. While this menu is on the screen game is paused. Player can use this menu for:

- Pausing the game
- Resuming the game

- Saving the game
- Loading game (This may cause the loss of current game)
- Settings
- Exit the game

3.1.3. Map

Map is a useful adjustment to the game to make it easier to play for the player. Player can see the position of the character and her surroundings with this menu. Map can be reached from “map button”. Initially it is ‘m’ key in the keyboard. It can be customized from the settings.

3.1.4. Save/Load

The game can be challenging or it may require more time than player has. Therefore we will add save/load option to the game. Player can save the game from the menu while the game is still running. Later on that save can be reached when player loads the game.

3.1.5. Settings

Here, the player can customize the game with the following settings:

- Enable/Disable music
- Enable/Disable game voice
- Adjust voice levels
- Change input keys
- Change brightness

3.1.6. Input Devices

We decided that the easiest input device for playability is keyboard. From keyboard player can use all the functionality in the game but mouse can still be used throughout the menu.

3.1.7. View Help

This function will help new players to understand the basics of the game. It will consist of few screenshots from the with and their explanations by words.

3.1.8. Credits

Credits option can be found in the menu. If clicked, user can see the list of developers and their contacts. It can be useful for further development of the project and bug reports.

3.2. Non-Functional Requirements

3.2.1. Smooth Gameplay

We will be doing our best to make the game playable at the same smoothness in every computer device being used nowadays. Our aim is having good graphics and animations while keeping the game as smooth as possible. It is possible to keep smoothness in most computers being used today since our program will use very small memory.

3.2.2. Extendibility

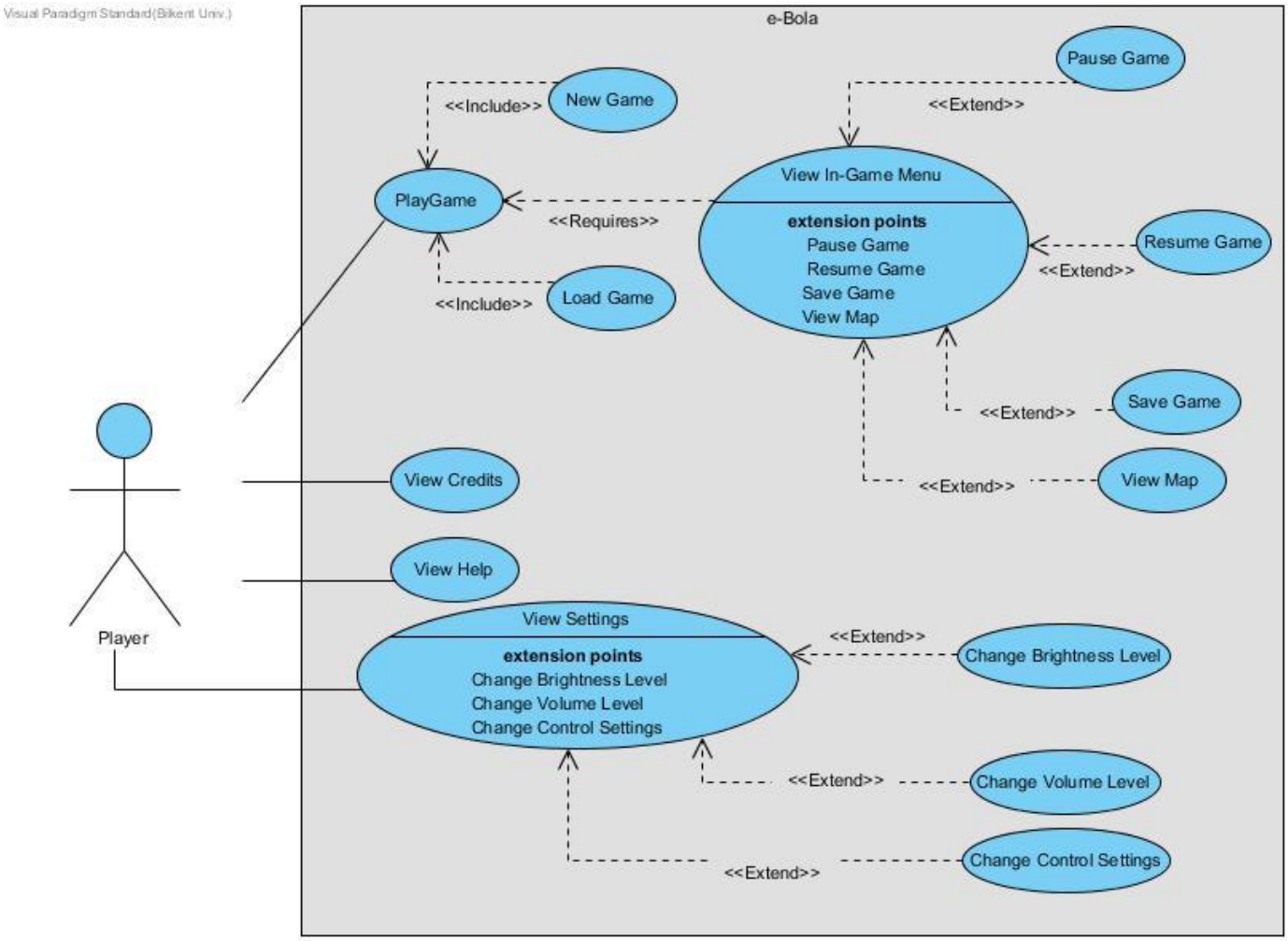
While designing the game one of our aim was to make game as extendable as possible for future works. New levels, items, enemies, backgrounds and even player characters can be added to the game without too much afford thanks to our design. We also added the credits button to get user opinion about further updates.

3.2.3. User Friendly Interface

Our graphical interface will be user-friendly that player can start the game with just one click and can change settings from our simple settings window. It is important for us to make player comfortable while playing the game that's why we tried to keep everything as simple as possible. There won't be any unnecessary setting or button in the game that will make it complicated.

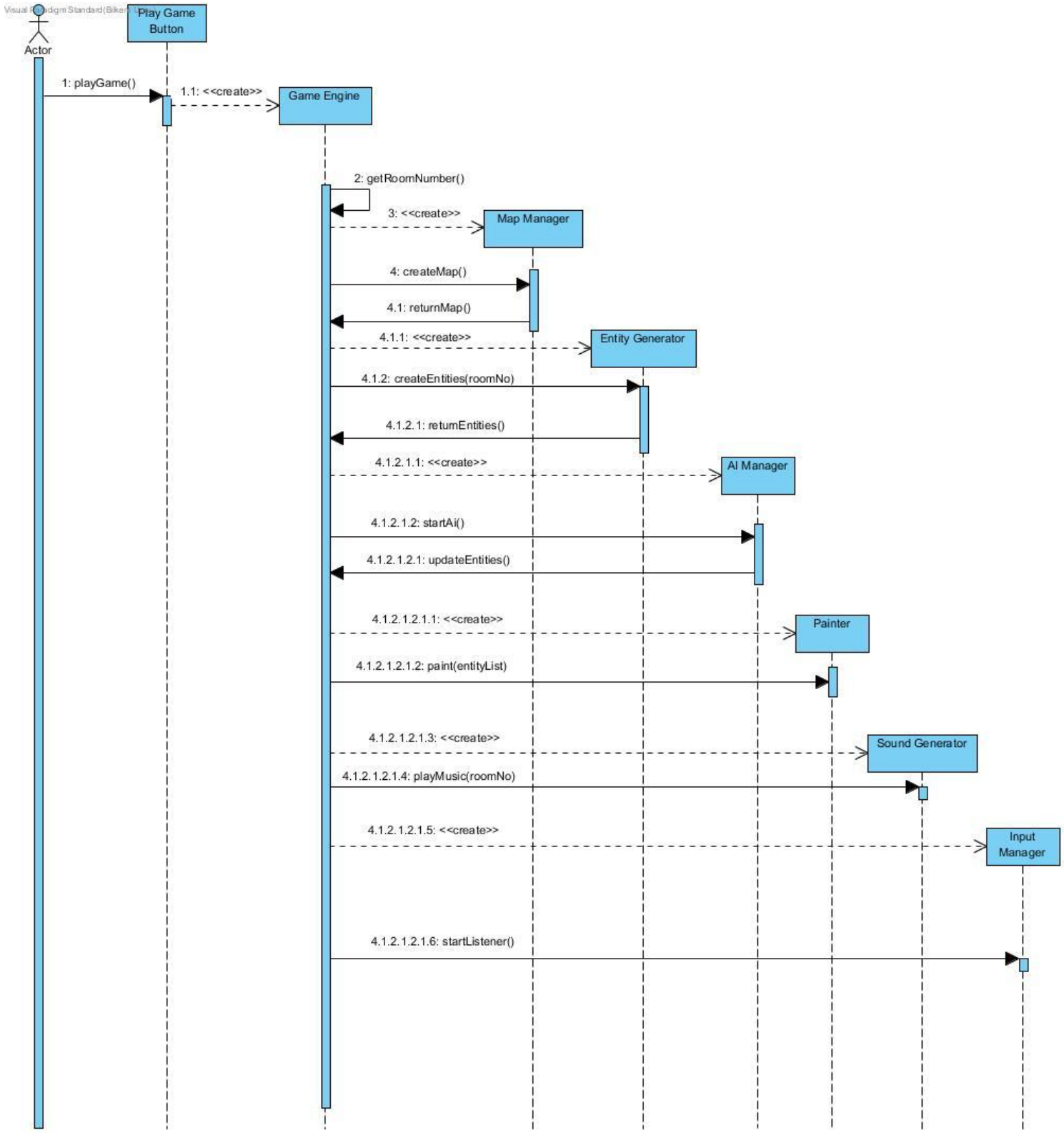
4. System Model

4.1. Use Case Diagram



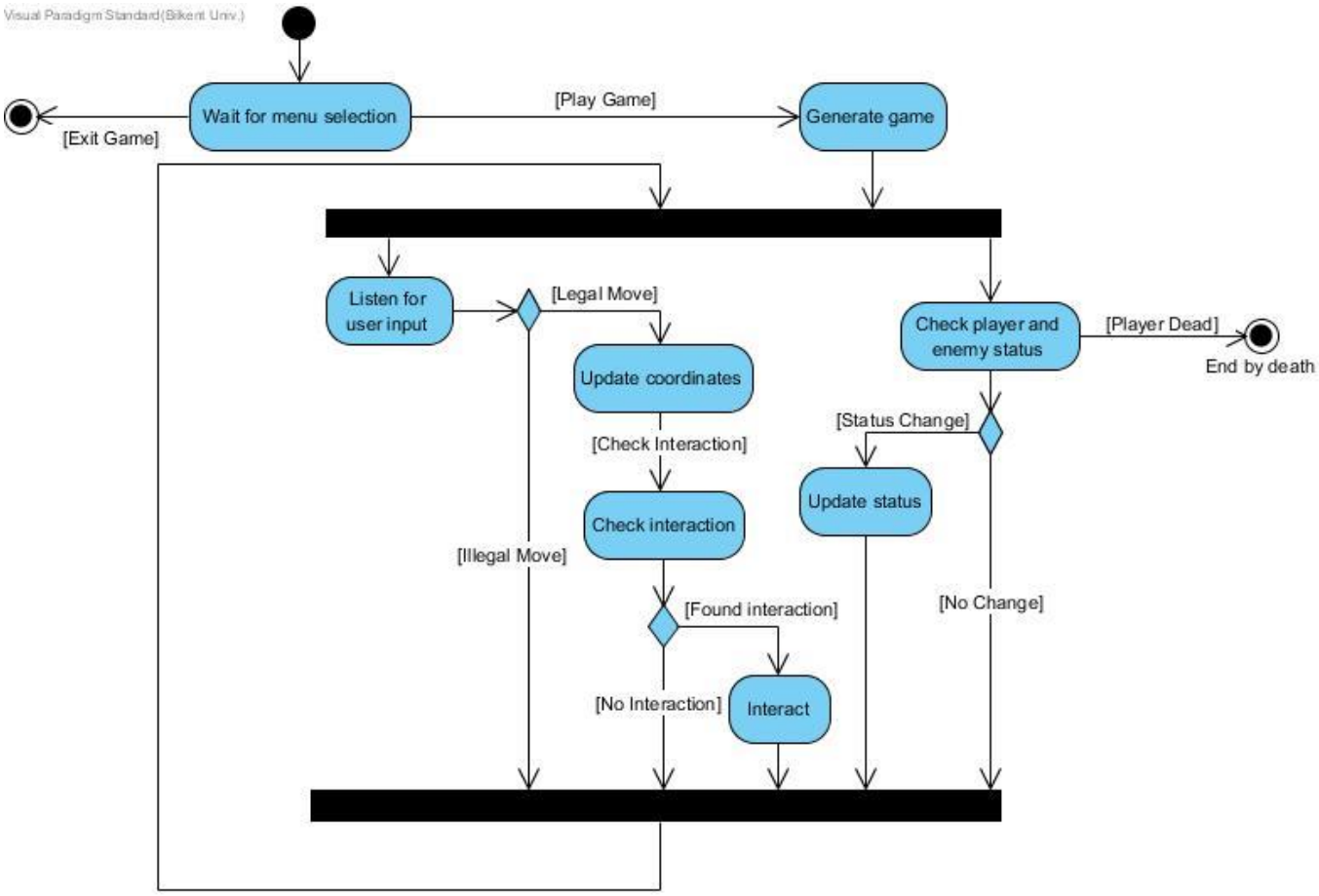
4.2. Dynamic Models

4.2.1. Sequence Diagram



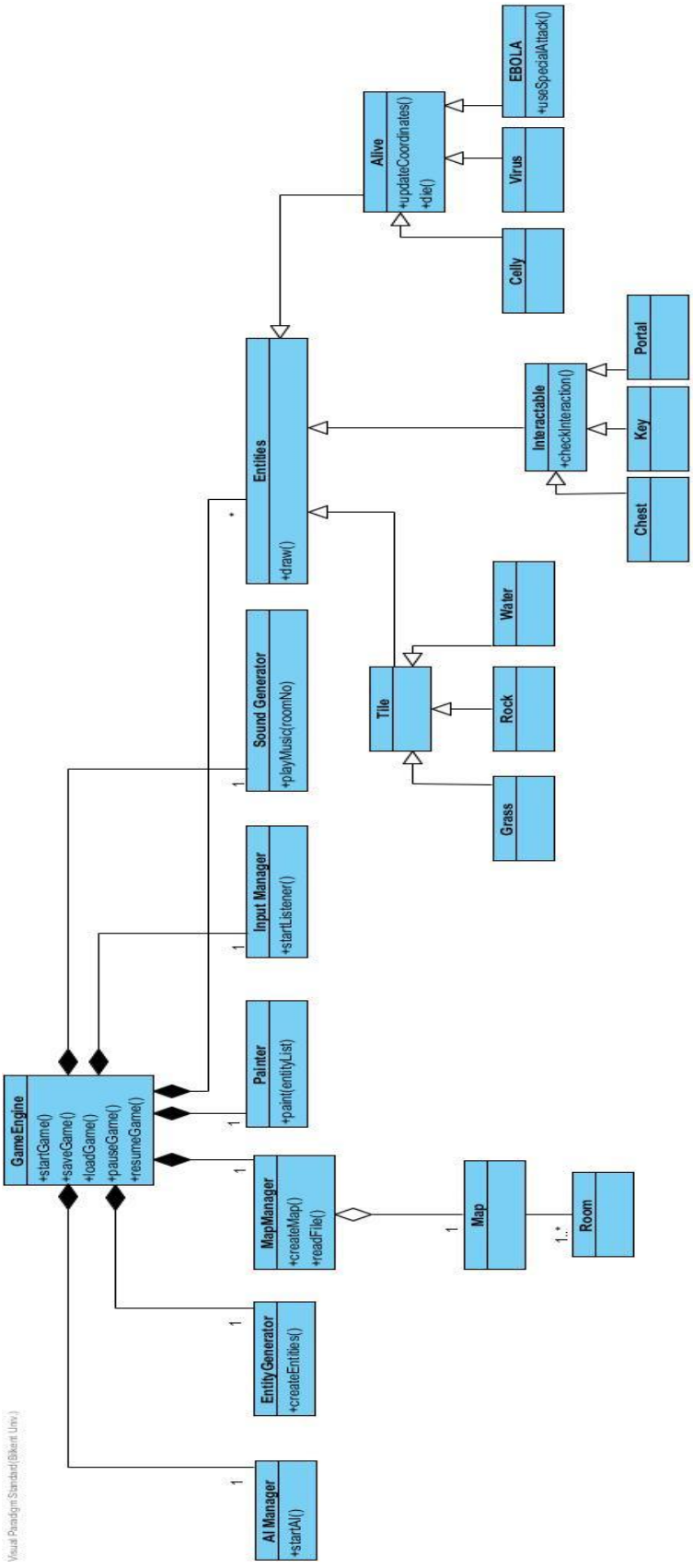
Scenario: The player starts playing the game using the play game button. Upon the request, game engine is created. Game engine creates map manager which creates and returns the current map. Using the room number returned by map generator, entity generator is called to create entities of the current room right after creation of entity generator. Then, the AI manager is created. AI manager controls movement of the virus and Ebola. Game engine is responsible of updating properties of all entities. Game engine sends command to repaint the screen accordingly with statuses of entities. Meanwhile, sound generator plays music specific to each room. Input manager manages the interaction between the player's keystrokes and movement of Celly. Game engine constantly checks if room is changed and updates corresponding objects.

4.2.2. Activity Diagram



After the actor (the player) selects the “play game” button, the main loop of the game starts. The game will concurrently listen for user inputs and check the status of the player and enemies. Status here refers to attributes that change as the game progresses, such as being dead or alive. When there are changes in statuses, the entity objects will be updated according to those changes. If these updates result in our player dying, the game ends. The status changes can occur even if the player does not give any inputs, so listening to user input and status checking have to be processed at the same time. If the user sends an input to the game, first the move has to be checked to see whether it is a legal move. An illegal move might be trying to go through an obstacle tile. If our move is legal, we update the coordinates of our player object. At this point, we check if our target location have interactable objects so we can register the interaction before the next game cycle. Now that all the updates and checks are made to the game, these are shown to the player and another game cycle begins.

4.3. Class Model



Game Engine: The main controller of the game, it orders other control classes to do their tasks when they are needed in the game.

AI manager: Controls the actions of non-user controlled entities and requests the game engine to update these entities if needed.

Entity Generator: Creates entity objects of the room the player is currently in.

Map Manager: Reads the text files that contain the rooms of the game and creates a Map object with them, that contain Room objects.

Map: Map of the current level, contains Room objects.

Room: Represents the rooms of the map.

Entities: A generalization for all the entities in game they share attributes such as: all of these entities can be drawn to the game screen, have a certain location.

Tile: Objects that represent the floor of the game. Consists of elements with different looks and modifiers.

Alive: Objects that can move, take damage and die. Consists of our enemies and main character, Celly.

Interactable: Objects that the player interacts to advance further into the game. Consists of elements such as gates to change rooms, or keys to open doors and chests.

Input Manager: Listens to user input and informs the game engine when the player causes an update to the game.

Sound Generator: Listens to events that could trigger any sound effects and plays them when needed.

Painter: Draws the entity objects to the screen on every game cycle.