



Bilkent University

Department of Computer Engineering

CS319 Object-Oriented Software Engineering Project

Siege

Final Report

Project Group 26: İrem Yüksel, Gökhan Şimşek, Emir Acımtaş, Furkan Küçükbay

Course Instructor: Uğur Doğrusöz

December 24, 2016

Table of Contents

1. Changes in the Design.....	3
2. Complications during the Implementation	9
3. User's Guide	
3.1. Introduction.....	9
3.2. System Requirements and Installation Information	
3.2.1. System Requirements.....	9
3.2.2. Installation Information.....	10
3.3. Inside Siege	
3.3.1. Game Overview.....	10
3.3.2. Game Objects.....	11
3.3.3. Score and Remaining Lives.....	13
3.3.4. Controls.....	13
4. Current Status of the System.....	13

1. Changes in the Design

The implementation is done with respect to the 3 Facade Design and the MVC design as stated in the design report.

- A constructor for the Tank object is added.
- To PlayerTank class, endPowerUp() method is added in order to make the power up's effect disappear after 10 seconds.
- Place class is removed since it was much easier to keep the location of each object in the map as integers.
- getType() method is added to the PowerUp class, since it was useful when a player picks a power up in the game.
- shoot() method is deleted from the Tank class since it was unnecessary to implement it there, but it is added to the Game class and handled in there.
- getScore() and increaseScore() methods are added to the Tank class
- addScore(int x) methods is added to the GameEngine.
- The attribute int shield is added to the PlayerTank class. If the shield power up is taken, this integer value becomes 3.
- createGameBody(GameBody gb, int x, int y) method is added to the GameEngine class and it places a new GameBody to the map.

- In MainFrame class, ArrayList<JPanel> list was unnecessary, so it is removed.
- A new panel named GameOverPanel is added.
- In GameEngine class, getEnemyTanksLeft() method is added which return the number of the enemy tanks left.

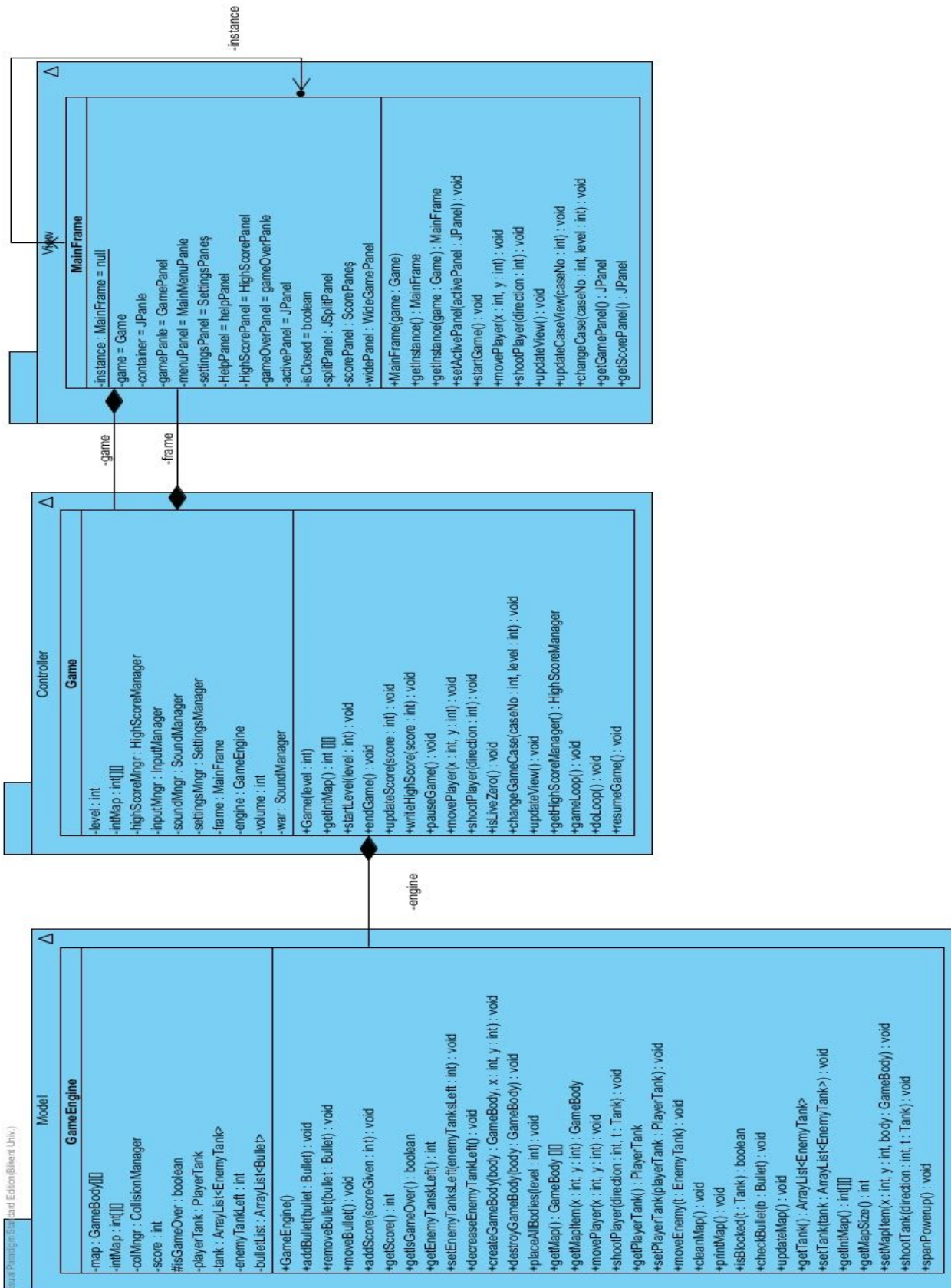


Figure 1. The 3 Subsystems (MVC), 3 Facade Classes and their Associations

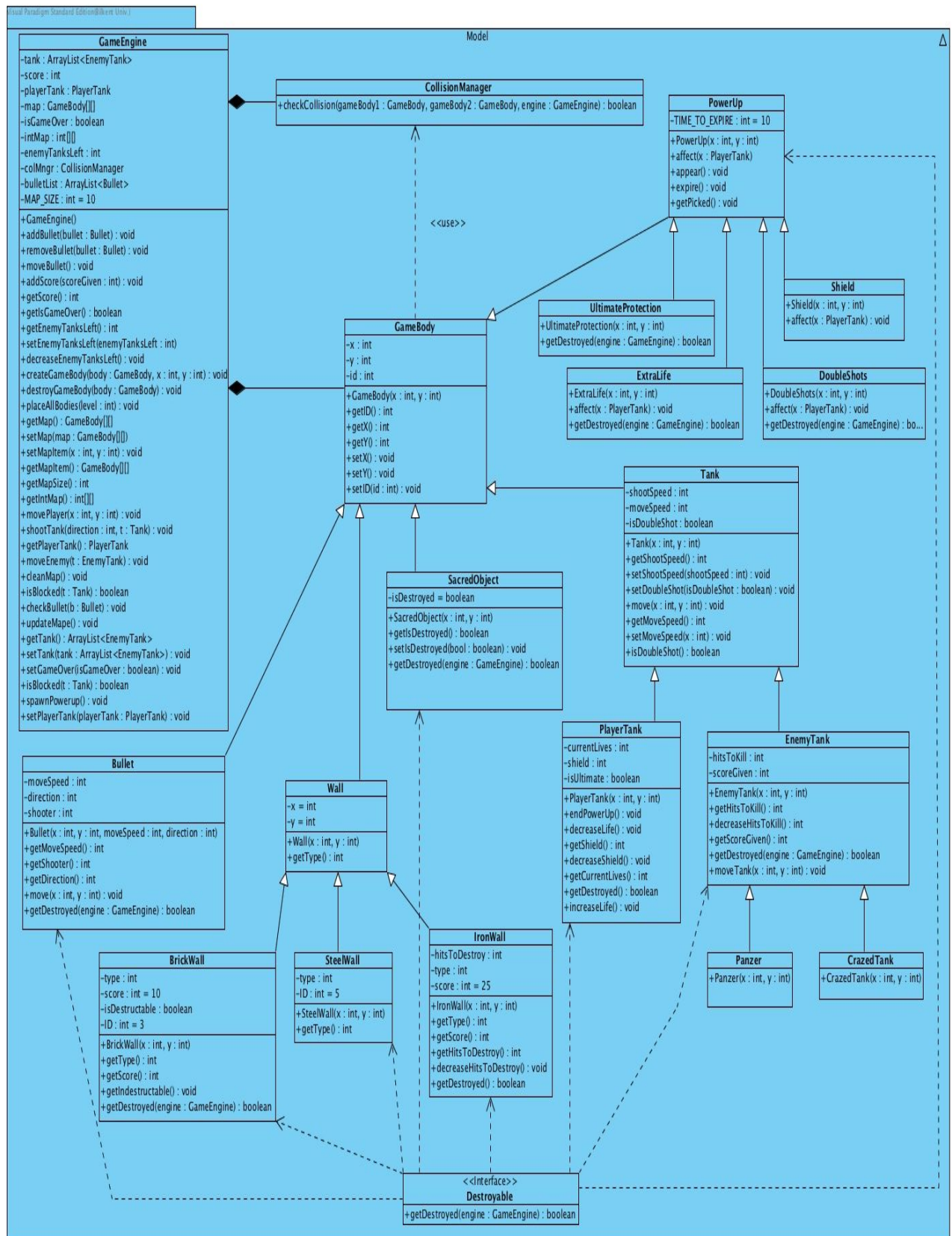


Figure 2. Detailed Model Subsystem

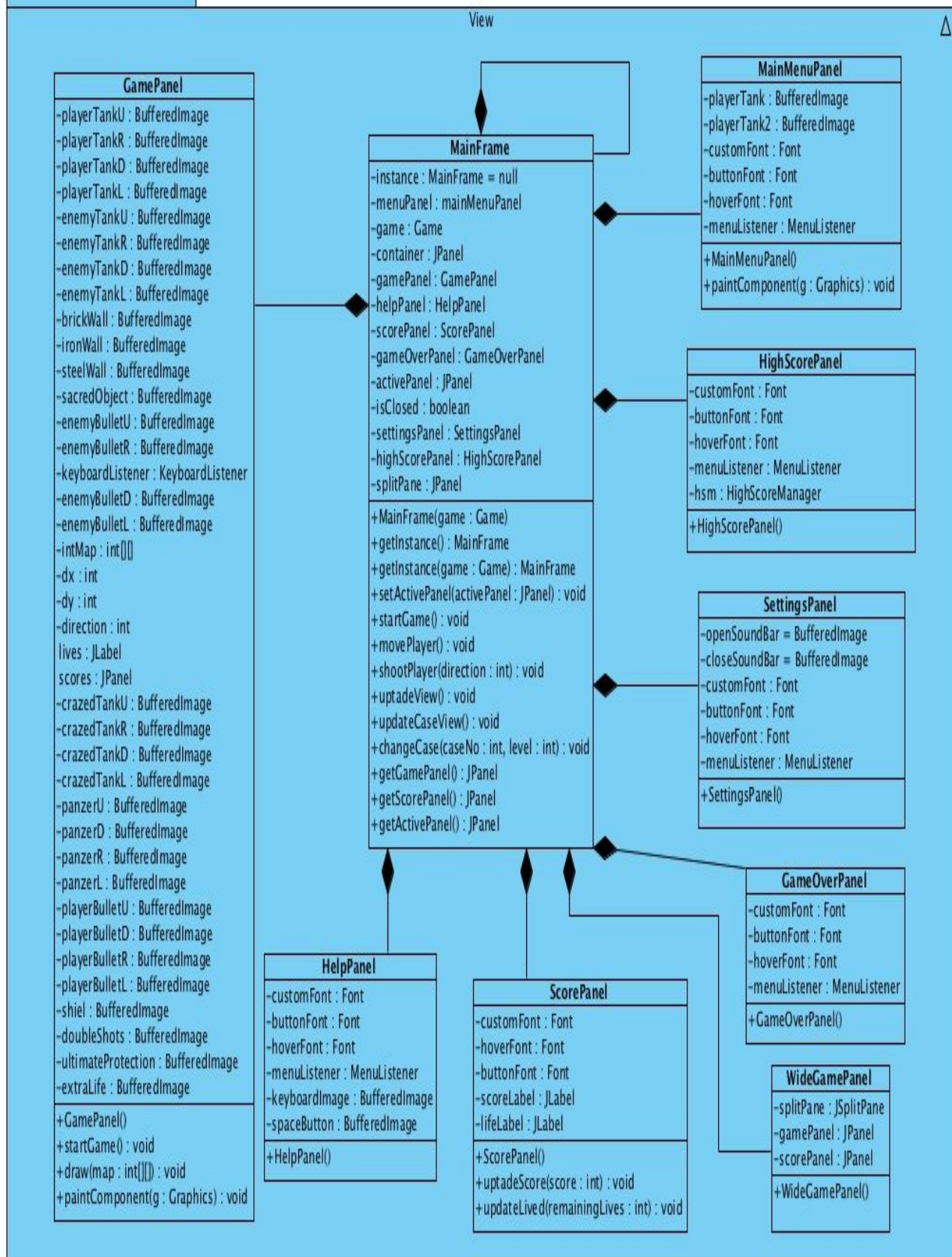


Figure 3. Detailed View Subsystem

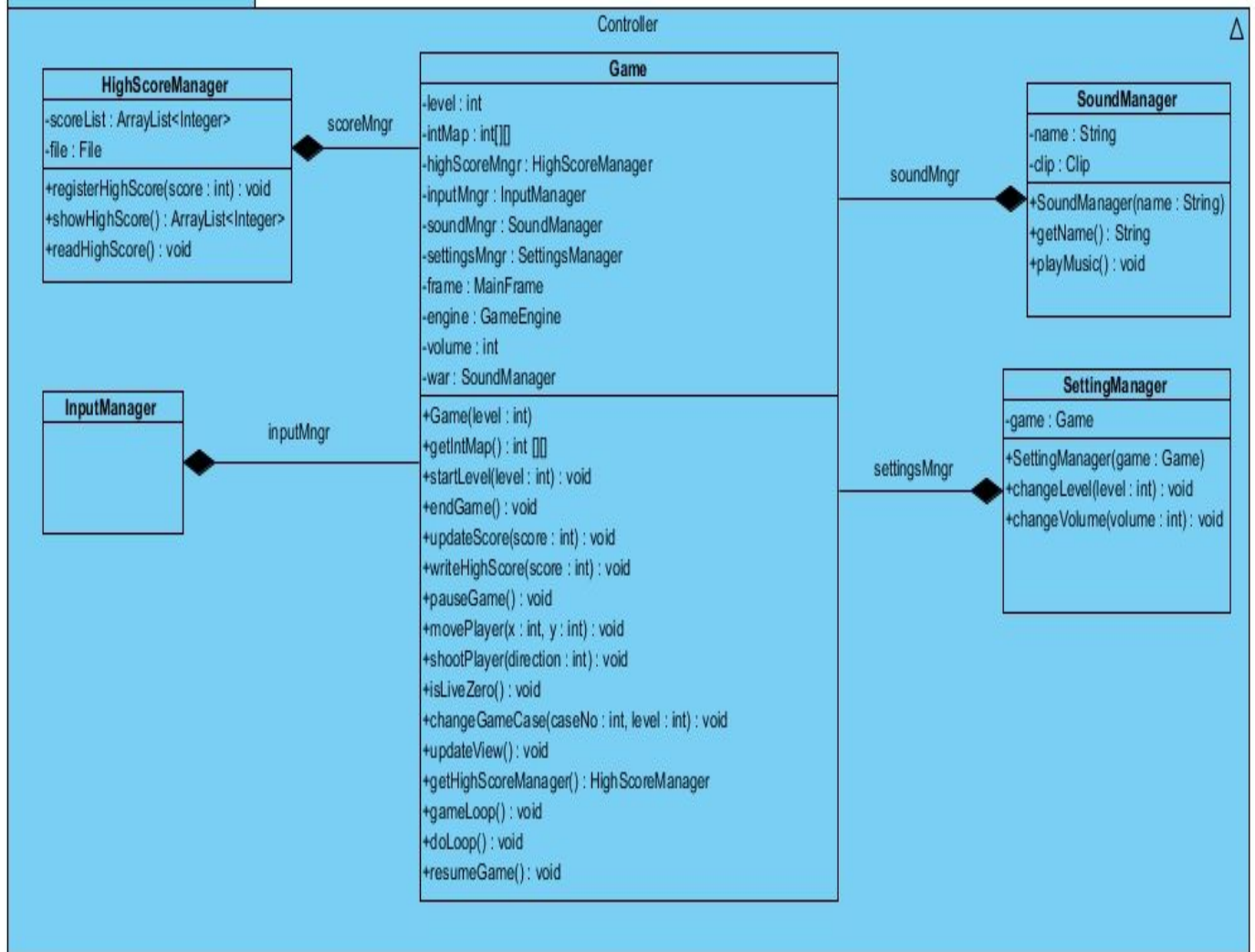


Figure 4. Detailed Controller Subsystem

2. Complications during the Implementation

One of the challenging part of the game is the implements the interface of the game because there are five level of the game and each level has a different difficulty. This is the reason why each level needs different types of structure and design.

Another challenging part of the project is Enemy tanks random move and shooting the bullet to the player tank. This is because the enemy tanks moves and shoot the bullet on their own and a basic artificial intelligence is moving enemy tank and shooting to the player. That's why enemy tank move and shoot are challenging part of the project.

3. User's Guide

3.1. Introduction

Siege is an arcade and maze-based game and it is inspired by an old game called Tank 1990. In the game the user tries to protect the sacred object with player tank and kill the enemy tanks. After enemy tanks are all destroyed the user finishes the level and starts to the next level. Siege contains 5 levels and after these 5 levels the game is over. There are randomly appering power-ups in the game which the player tank can pick up. Also if the user's score is one of the highest 5 scores at the end of the game, it gets saved to be shown later.

3.2. System Requirements and Installation Information

3.2.1. System Requirements

The system require Java SE Environment to play. It can be downloaded from this website:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>. The graphics resolution should be 1024x768 and

the operation system should be Windows 7 or up.

3.2.2. Installation Information

The users can found the game files from this website:

<https://github.com/gokhansim/CS319-section3-group26>

After reaching the website, users can run the .jar file of the game to play. If the users are familiar with Java IDE compilers, the second option is that they can use .java files to compile and run the game. In order to use these .jar and .java files, the computer has to contain Java SE Environment.

3.3. Inside Siege

3.3.1. Game Overview

In Siege, player controls his/her own tank and tries to protect the sacred object from the enemies. Player can destroy the walls and the enemy tanks by shooting them. Enemy tanks can also shoot the player and the walls. If the player gets hit, his/her live gets decreased.

There are 5 different levels, where difficulty increases respectively. Player can start playing from any level, by choosing it from the Settings. Each level have different maps and different amount of enemies.

The most essential objects are the tanks and the walls. Player controls the player tank and enemy tanks are controlled by the AI. Walls are pre-defined by level maps and they are the obstacles on the tanks' road. Also, the walls can be destroyed by both player and enemy tanks.

The main goal of the player is to protect the Sacred Object. The game ends when the Sacred Object is captured by the enemies or when the player has no more lives to play.

In Siege, there is a high score list which shows the highest 5 scores that players' got in the game so far.

3.3.2. Game Objects



Player tank is controlled by the player using the keyboard.

Player tank can shoot and destroy other objects. The player tries to protect the Sacred Object from enemies.



This is the most basic enemy tank which can be destroyed with 3 shots and if the player can destroy them, s/he gets 10 points.



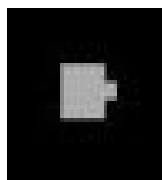
Crazed enemy tanks can be destroyed with 3 shots and if the player can destroy them, s/he gets 20 points.



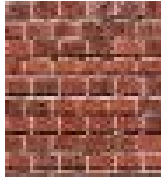
Panzer enemy tanks can be destroyed with 5 shots and if the player can destroy them, s/he gets 15 points.



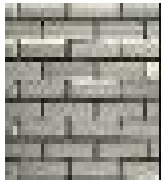
The Sacred Object is the most important object of the game since if it is captured by the enemies, the game is over.



Bullets have only one direction and if it hits an object on its way, it destroys or decreases the object's lives unless it is a Steel Wall.



Brick wall is the most easily destroyed wall in the game. It can be destroyed with 1 shot.



Iron walls are harder to destroy than the brick walls. They get destroyed with 2 shots.



Steel walls are indestructable in the game.



Double shots power up when taken by the player, makes the player shoot two times faster.



Ultimate protection power up makes the Sacred Object non seizable for 10 seconds.



Shield power up protects the player tank from enemies' attacks for 10 seconds and then it disappears.

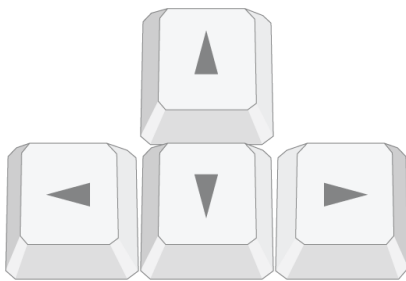


Extra life power up incrases the current live of the player by 1.

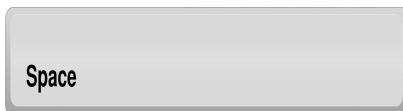
3.3.3. Score and Remaining Lives

Player will get extra points after shooting an enemy tanks or breaking a wall. In order to keep the player excited and motivated, the current score and the remaining lives are shown at the bottom of the game.

3.3.4. Controls



Up, down, left and right moves the player tank up, down, left and right respectively, unless there is a steel wall on the direction or if it points outside the map.



When pressed, space button makes the player tank shoot.

4. Current Status of System

The use cases that we give in the analysis report and functional and non- functional requirements are all added to the system. The player can play the game without any error or bugs. The interface of the system is very little changed. For example, the fonts of the text are different then the analysis report interface model.