# bilkent.png (709×709)

**Bilkent University  
CS 319**

**Object-Oriented Software Engineering**

**Fall 2014**

**Analysis Report  
Project Name: Gold Miner  
Group #2**

**Murat Selman Gündoğmuş 21000213**

**Şükrü Efe Keleşoğlu 21001407**

**Ezgi Nostar 21101633**

**Ahmet Said Sayarlıoğlu 21101612**

Contents

[1](#_Toc433811820)

[1. Introduction 6](#_Toc433811821)

[2.Requirement Analysis 7](#_Toc433811822)

[2.1 Overview 7](#_Toc433811823)

[2.1.1 Gameplay 8](#_Toc433811824)

[2.1.2 Levelling 8](#_Toc433811825)

[2.1.3 Bonus Items 9](#_Toc433811826)

[2.2. Functional Requirements 10](#_Toc433811827)

[2.2.1. Loading Stage 10](#_Toc433811828)

[2.2.2. User Inteface 10](#_Toc433811829)

[2.2.3. Start Button 10](#_Toc433811830)

[2.2.4. Exit Level Button 10](#_Toc433811831)

[2.2.5. Time 10](#_Toc433811832)

[2.2.6. Point 11](#_Toc433811833)

[2.2.7. Target Point 11](#_Toc433811834)

[2.2.8. Level 11](#_Toc433811835)

[2.2.9. Hook 11](#_Toc433811836)

[2.2.10. Items 11](#_Toc433811837)

[2.2.11. End Game 11](#_Toc433811838)

[2.3. Non-functional Requirements 12](#_Toc433811839)

[2.3.1. Extendibility 12](#_Toc433811840)

[2.3.2. Easy to Play 12](#_Toc433811841)

[2.3.3. Availability 12](#_Toc433811842)

[2.3.4. Performance 12](#_Toc433811843)

[2.4. Constraints 12](#_Toc433811844)

[2.5. Scenarios 12](#_Toc433811845)

[2.6 Use Case Models 13](#_Toc433811846)

[2.7. User Interface 18](#_Toc433811847)

[2.7.1 Main Menu 18](#_Toc433811848)

[2.7.2. Game Screen 19](#_Toc433811849)

[2.7.3. Help Screen 20](#_Toc433811850)

[2.7.4 Sound Setting 21](#_Toc433811851)

[3. Analysis 23](#_Toc433811852)

[3.1 Object Model 23](#_Toc433811853)

[3.1.1 Domain Lexicon 23](#_Toc433811854)

[3.1.2 Class Diagram 25](#_Toc433811855)

[3.2. Dynamic Models 26](#_Toc433811856)

[3.2.1 State Chart Diagram 26](#_Toc433811857)

[3.2.2Sequence Diagram 28](#_Toc433811858)

[4. Conclusion 32](#_Toc433811859)

[Figure 1 Use Case Diagram 2](#_Toc433809349)

[Figure 2 Main Menu 7](#_Toc433809350)

[Figure 3 Game Screen 8](#_Toc433809351)

[Figure 4 Help Screen 9](#_Toc433809352)

[Figure 5 Instructions 9](#_Toc433809353)

[Figure 6 Sound Settings 10](#_Toc433809354)

[Figure 7 Gold 10](#_Toc433809355)

[Figure 8 Rock 10](#_Toc433809356)

[Figure 9 PowerUp Bag 10](#_Toc433809357)

[Figure 10 Diamond 11](#_Toc433809358)

[Figure 11 Animal 11](#_Toc433809359)

[Figure 12 Class Diagram 14](#_Toc433809360)

[Figure 13 State Chart Diagram 15](#_Toc433809361)

[Figure 14 Sequence Diagram for Start Game 17](#_Toc433809362)

[Figure 15 Sequence Diagram for the Game Loop 18](#_Toc433809363)

[Figure 16 Sequence Diagram for the Settings 20](#_Toc433809364)

# Introduction

We decided to design and implement a game called Gold Miner. It is a basic flash game that users try to collect gold and hidden treasures in limited time. The main purpose of the game is to score maximum point by collecting maximum amount of gold, jewelry and power ups. While doing this , users have to escape the randomly distributed bombs and dynamites to gain more points. Bombs and dynamites are a danger for gold and other collectibles items by destroying the collectible items in the area where they are belong.

In addition to that, the game consists of a number of different levels which are following another. User can play next level by reaching the target of a certain level.

The game that we inspire by is following.

http://www.kraloyun.com/Oyun/Hazine-avcisi

We plan to develop our game with different features. In Gold Miner, we are going to use new features including new power ups, new maps and challenges.

The game will be a desktop application and will be controlled by a keyboard’s up and down buttons.

This report contains two parts indeed. Requirement analysis part contains an overview of the game, describes functional and non-functional requirements, constraints and scenarios, presents use case models and user interface. Analysis part contains object model including class diagrams and dynamic model including state chart and sequence diagram.

# 2. Requirement Analysis

## Overview

Gold Miner is very easy to play and funny. Users just need to click down arrow on keyboard to activate the hook which provides to collect gold and hidden treasures. The main object in this game is the hook which has two states: rotate and rewind. Rotating provides the hook is rotating from left to right at the top of the screen. Rewinding provides the hook returns to its initial position in a straight path and then changes the state to rotate by allowing the player to shoot again. Every time hook is thrown, hook can collect the gold or other items depending on location of the gold and path that hook is thrown. Every time player collects a new item, user can increase the cash according to point of a collectible item to reach the target. If the player reaches the goals by hitting shot without missing any time, player can see the new map by passing the new levels.

## Gameplay

The player will need keyboard to play the game. Players have to use the hook and reel to mine gold and other treasures out of the land. When the hook swing back and forth, user press the down button to activate and collect items. Once the hook has grabbed something, it will begin to reel it up. Heavy objects like rocks and large pieces of gold will be more time to pull. PowerUp Bag contains random amounts of cash a strength power-up, or a stick of dynamite. Users have to collect the target amount of money by the end of the level if users do not meet their goal by the end of the level, it's game over. Users’ cashes carries over from one level to the next.

## Levelling

In Gold Miner, there are 10 levels which are in difficulty order. When the player passes a next level, the new map contains more bombs which destroy the collectible items. In addition to that in next levels, the target point is higher than the previous levels. In this way, user has to reach higher target point in a more dangerous map. Also, even the target point is increased, the given time is not changed and it will be fixed to 60 seconds. The advantageous point about the leveling is that the game can add new power ups and diamond which are more valuable or cause to decrease the cash point.

Jewels are designed to their values. The different kinds of jewels will come up in each new level and it will see as in ascending value order. Jewels are stone, gold, diamond, ruby, sapphire. In addition to that, to make our game is harder, bombs will appear in higher levels. Also, PowerUp Bag and animals which blocks to collect to gold and jewelry will be appeared.

**1st Level:** Just stones and golds will be appearing.

**2nd Level:** Stones, golds and diamonds will be appearing.

**3rd:** All jewels and bombs will be appearing.

**4rd and Higher Levels:** PowerUp Bags and animals will be appear.

## Bonus Items

There are several bonus items in the game. Some of them are power ups and some of them are penalties.

Users can either obtain a bonus or a time penalty from the PowerUp Bag in the game. Users will also get a time penalty when they pull back the hook empty. Also the player do not know exactly what type of power-up or power-down coming when they pull the PowerUp Bag. In addition to that, some of power up in the PowerUp Bag such as lots amount of cash is rare compare to other power ups such as dynamites or power downs such us time penalty.

**PowerUp Bag:** It will be appeared in 4rd and higher levels and provides to reach different bonus items.

**Bomb:** It causes to destroy collectible items nearby. Also it blocks to the bigger golds.

**Dynamite:** It provides to reset user’s hook when they collect the stones and unwanted items. In this way, user can collect more items without losing any time.

**Time Penalty:** It comes from the PowerUp Bag. Users will take 3 seconds time penalty.

**Extra Time:** It comes from the PowerUp Bag. Users will take extra 10 seconds.

## Hook

The hook is the only item that user can control. User can collect the item with the hook by using keyboard down button. The hook can keep and pull the items according to their size and weight. If the hook keeps the bigger item or stones, the time for pulling increase and user can lose more time. On the contrary, the item which is keep is small such as diamond, the hook can pull this item easily without wasting a time. Also, hook can be strengthen by power ups and in this way hook speed is increased.

## Collectible Items

In Gold Miner, there are different types of collectible items such as bomb, stones, jewellery, gold, animals and power ups and downs which contains dynamite, time, strength. This different kind of items will come up in each new level. There are different kind of gold according to the size and points, one type of diamond, one type of animal, one type of bombs and different kind of stones according to the their size and weight.

## 2.2. Functional Requirements

### 2.2.1. Loading Stage

The game loads all elements which are required for different stages such as images, audio files and panels.

### 2.2.2. User Inteface

User interface of the game consist of three page which are main page, side-page and game page. In main page, there is instruction panel on the bottom of the page to tell how to play the game. Another thing in main page is start button. Start button is interactive with Mouse actions. Also main page has audio file and image to make more attractive the game. Another page type is side-page. In side-page, there is a panel to tell users’ target point and remaining time also it shows player the current point and level and the last thing on the side page is the exit level button. Last page type is game page. This page has hook, and collectable items such as various size of gold and rock, diamond, random powerUP bag and bomb.

### 2.2.3. Start Button

Start button is interactive with Mouse actions. It directs to game page.

### 2.2.4. Exit Level Button

Exit level button is interactive with mouse actions. It ensures to user exit the current level and it directs to main page.

### 2.2.5. Time

It is visible part of the game page. It counts 60 seconds reversely at each level

### 2.2.6. Point

It is visible part of the game page. User points will increase when user catches the item such as gold, rock, diamond or random pouch. Also if user point is less than the target point, user fails the level and game is over.

### 2.2.7. Target Point

Target point is differs in each level. It increases when user passes the level.

### 2.2.8. Level

Each level has difficulty. When user passes the level, level difficulty will increase. Using heavy items and increasing the level target point ensure to make more difficult level regarding to previous level.

### 2.2.9. Hook

Hook is moving from the left to the right at the top of the game page. It is interactive with keyboard actions. When specific keyboard button is pressed, hook moves in a fix direction until it hits an item and pulls up the item

### 2.2.10. Items

There are five types pf item such as gold, rock, diamond,bomb and random pouch. If user catches the bomb, there will be exploison around the bomb and it destroys the item which around the bomb. Diamond is another item in the game. There is only one type of diamond, it is light and it has higher point regarding to other items. Because of its’ lightness, hook pulls up easier and it takes less time.In Random pouch there are third type of item. It has various items in itself such as dynamite,strength or point. System randomly picks one of them when user catches the random pouch. Also there are rocks and golds in the game. Both of them have different weight and point.Rocks have significantly lower point than golds. Heavier rocks and golds have higher points. However, user spends to more time to pull up the item. Regarding to golds and diamonds, rocks give less points.

### 2.2.11. End Game

The game should end if point is less than target point or user clicks the “exit level” button during the game. In this situation, it directs to main page.

## 

## 2.3. Non-functional Requirements

### 2.3.1. Extendibility

Game should available to update the functional requirements in the future. For instance, new levels or new items can be add. It can be change the value of items or add new obstacles.

### 2.3.2. Easy to Play

Game is easy to play. Children, teens and adults can play this game easily. There is no complex user interface. Also there is a description of gameplay at the main page.

### 2.3.3. Availability

Game will be played in most type of computers even old fashioned systems in the market.

### 2.3.4. Performance

Because of the limitation of time during the game, system response time has utmost importance. So the game should be high performence.

## 2.4. Constraints

- User should have flash player on their computer to play this game

- If developer wants to upgrade or change the game, java programming language should be used.

## 2.5. Scenarios

- User enters the game and sees the game instruction

- User clicks “Start Game” button and sees the target score.

- During the game, user shoots the hook with keyboard actions and collects the items

- If user catches gold, rock, diamond or random pouch, user gains points. If user catches TNT, user looses time.

- If user gain target score, user will play next level.

- If user click the “Exit Level” button or does not collect target points, the game will be end.

## 2.6 Use Case Models

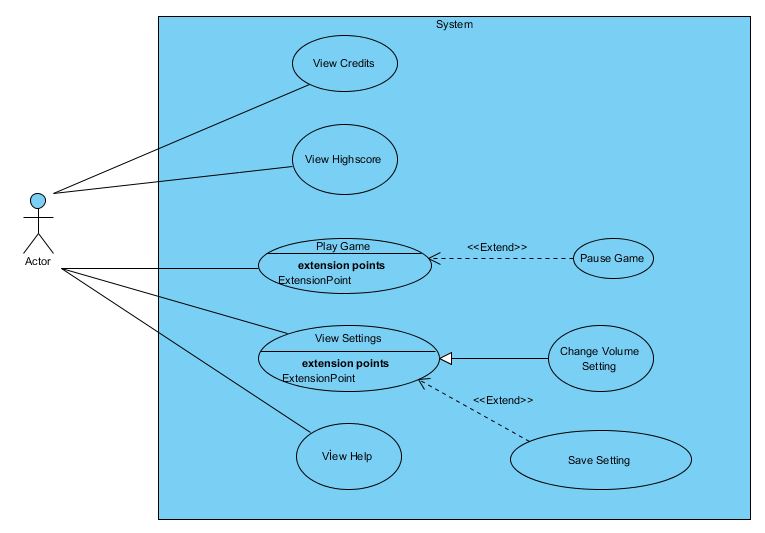
****

Figure 1 Use Case Diagram

**Use Case Model #1**

**Use case name:** Play Game

**Participating actors:** Player

**Entry Condition:** Player presses the play game button and starts the process.

**Exit Condition:**

* Player has completed all the levels successfully OR
* Player has chosen to exit the game in pause menu OR
* Player has not reached the target at the given time.

**Main Flow of Events:**

1. Player starts the game.
2. The system constructs the first map.
3. Player reaches the target point
4. The system generate new map.
5. Player finishes all the level.
6. The system shows the score of the player.
7. Player enters the name.
8. The system saves the score and adds the score to the high score table.
9. Players return the main menu.

**Alternative Flows of Events:**

* Player has not reached the target point when the time is over.(display game over message and go steps 6)
* Player has chosen to exit (go to step 9).

**Use Case Model #2**

**Use case name:** Change Settings

**Participating actor:** Player

**Entry Condition:** Player presses the settings button and starts the process

**Exit Condition:** Players make the changes and return the main menu

**Main Flow of Event:**

1. Player chosen to view settings menu.
2. Player can adjust the settings of the game.
3. The system saves the changes
4. The system returns the main menu.

**Alternative flow of the event:**

* Player does not change any settings(go to step 4)

**Use Case Model #3**

**Use Case Name:** View High Score

**Participating Actor:** Player

**Entry Condition:** Player presses the high score button and starts the process

**Exit Condition:** Players return the main menu

**Main Flow of the Event:**

1. Player chosen to view high score.
2. The system display high score.
3. Players return main menu

**Use Case Model #4**

**Use Case Name:** View Help

**Participating Actor:** Player

**Entry Condition:** Player presses the help button and starts the process

**Exit Condition:** Players return the main menu

**Main Flow of the Event:**

1. Player chosen to view helps and hints.
2. The system display help and hints.
3. Players return main menu.

**Use Case Model #5**

**Use Case Name:** View Credits

**Participating Actor:** Player

**Entry Condition:** Player presses the credits button and starts the process

**Exit Condition:** Players return the main menu

**Main Flow of the Event:**

1. Player chosen to view credits.
2. The system display credits.
3. Players return main menu.

**Use Case Model #6**

**Use Case Name:** Pause Game

**Participating Actor:** Player

**Entry Condition:** Player presses the pause button and starts the process

**Exit Condition:**

* Player chose to continue to play game OR
* Player can chose to return Pmain menu OR
* Player can exit the game.

**Main Flow of the Event:**

1. Player presses the pause button during the game.
2. The system displays the pause menu.
3. Players continue the game

**Alternative Flow of Events:**

* Player chose to return main menu.
* Player directly exit the game

## 2.7. User Interface

### 2.7.1 Main Menu



Figure 2 Main Menu

When player open the game, s/he sees screen like this with 4 choice Play Game, high Score, Credits and Exit Game

### 2.7.2. Game Screen



Figure 3 Game Screen

This screen is a example of our game screen. After pushing the button of the “Play Game ” at the main menu player have a screen like that. There will be several differences in our game for example there is only points instead of money and coins but it will be generally like this.

### 2.7.3. Help Screen



Figure 4 Help Screen

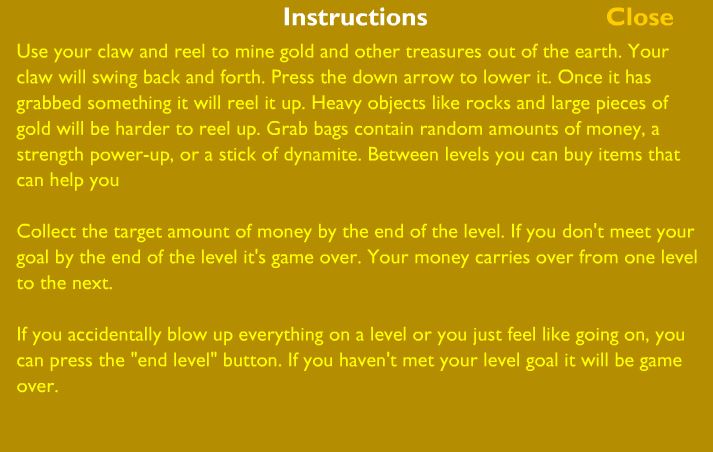


Figure 5 Instructions

The player can enter this screen by pushing the button of “Help” in the main menu. This is our help screen of the game. Player can learn instruction of the game and general information of the game.

### 

### 2.7.4 Sound Setting



Figure 6 Sound Settings

This button is in the settings page and we can reach setting page from the main menu.

**2.7.5. Collectibles**



Figure 7 Gold

This is system main source of the point. In the map there are several f gold with different sizes. Size indicates that amount of point that the gold has.



Figure 8 Rock

In the game like gold there are several of rocks. But these rocks do not have any points. And like gold there is different size of rocks. Bigger ones are harder to pull and consume much more time than smaller ones



Figure 9 PowerUp Bag

In the game there are 3 powers up and they are hidden in the powerUp bags.



Figure 10 Diamond

In the game after first level there are diamonds in the game map which are smallest element in the game but the amount of point to of the diamond is highest in the all game

C:\Users\Ahmet\AppData\Local\Microsoft\Windows\INetCache\Content.Word\aniaml.jpg

Figure 11 Animal

This is the last element of the collectibles. It has animal shape and has the lowest point of the game.

# 3. Analysis

### 3.1 Object Model

### 3.1.1 Domain Lexicon

The object model of the gold miner consists of 20 classes.

In our design Game Manager is center of the all class interaction It’s manage the all user interaction by the system also it is the manager class of our system that carries out the task of the organization of all game. Our other classes explained below.

* **CollectionManager:**  This class is responsible for managing the collection issues of game objects that we collect.
* **Menu:** This class provides menus of the game to the game manager
* **Input Manager:** This class handles player’s inputs.
* **SoundSetting:** This class handles the sounds volume of the game
* **MapManager:** This class manages the map itself and background image of the map.
* **GameMap:** This class contains GameObject class. It also contains dynamic objects like hook
* **Sprite:** This class consist of objects images and positons so all game objects at least have one sprite
* **GameObject:** This class basically contains all entity objects of the system which include hook and collectible class. Also collectibles class has 6 child classes.
* **Hook:** This class is the main object of the game. It collects the collectibles.
* **Collectibles:** This class contains objects which can be collectible by the hook in the game. There are 6 child classes of the collectibles.
* **Bomb:** This class is the child class of the Collectibles. This objects main purpose is the make the game harder with the explosion of the bomb.
* **Rocks:** This class is the child class of the Collectibles. This objects main purpose is the make the game harder with occupation of the place with lower point.
* **Jewelry:** This class is the child class of the Collectibles. This is a collectible object which has most point in the game.
* **Gold:** This class is the child class of the Collectibles. It is the main source of the point in the game.
* **PowerUps:** This class is the child class of the Collectibles. Its purpose is the make the game easier to play with the several power ups for the hook. This class has 3 child class.
* **Dynamite:** This class is the child class of the PowerUps. Its purpose is the reset the hook if the player took the heavy and pointless object.
* **Time:** This class is the child class of the PowerUps. Its purpose is the increase the remaining time of the game.
* **Strength:** This class is the child class of the PowerUps. Its purpose is the decrease the pullHook time.
* **Animals:** This class is the child class of the Collectibles. . This objects main purpose is the make the game harder with occupation of the place with lower point.

### 

### 3.1.2 Class Diagram

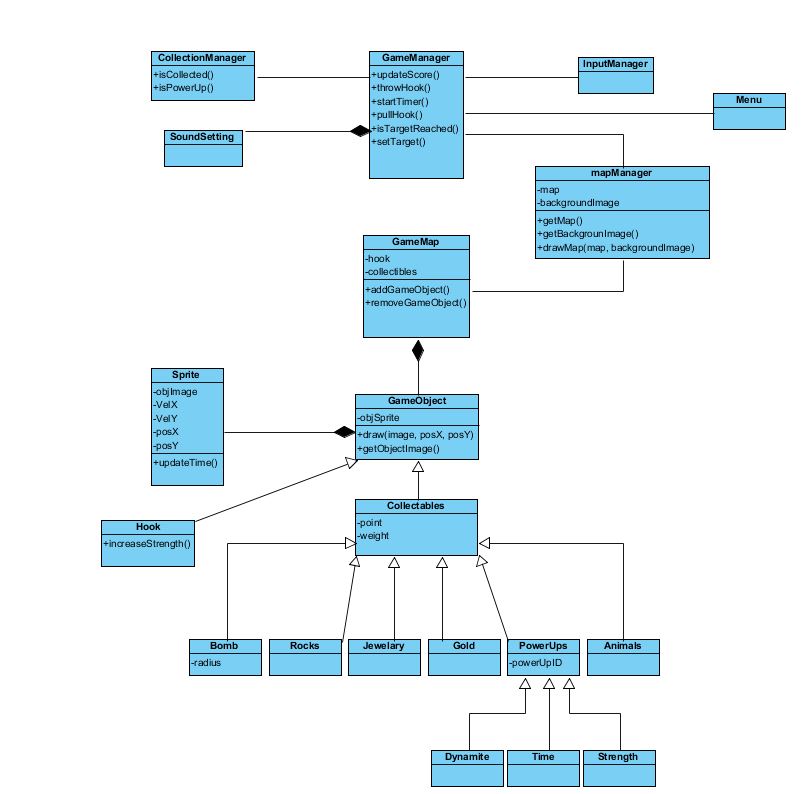


Figure 12 Class Diagram

### 3.2. Dynamic Models

### 3.2.1 State Chart Diagram

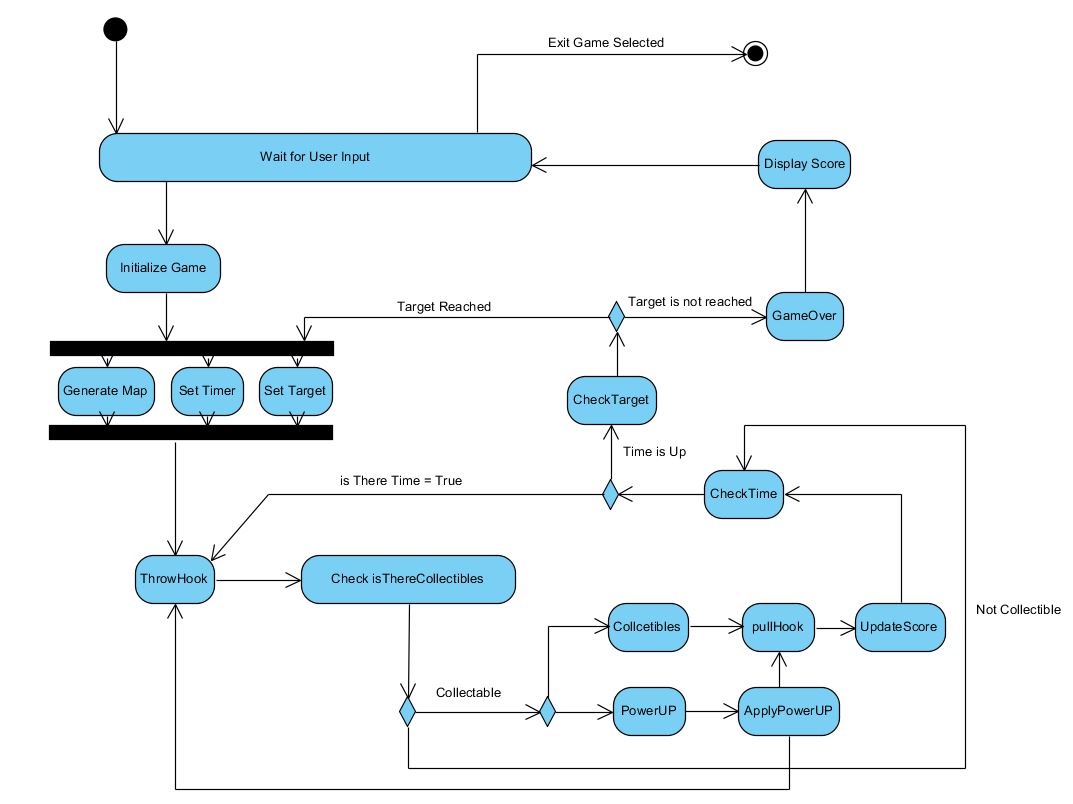


Figure 13 State Chart Diagram

Activity diagram show the main flow of the game event. In every cycle, Game Manager controls the hook and collected items simultaneously. When user select the Play Game the system start to initialize game which contains generate map, set timer and set target functions.

After the creation of the map finished, system wait for the player input from keyboard to start the process of the throwing hook. After the throwing hook process finished the system control if there is a collectible item where the hook is thrown. If there is not any collected item hook return back the initial state without any collected item. After this state, system control the remaining time for the rest of the game. If there is enough time player can throw hook again. If there is collectable item where the hook is thrown, collection manager check if the collected item is a power up or cash point. If it is a power up, system apply the power up to the hook and hook can pull the item rapidly. If the collected item is cash point, system adds the value of the cash point item to the players point and updates the score. Then system checks the remaining time and the system see that the time is over, the system check if the total point exceeds the target point of the level. After checking the target point of the level the system decides whether the new map generated or not. If the target is reached, system generate new map of the next level and also set the new target point and the new time. If the target is not reached, current game is over and the system displays the high score screen and wants from player to enter his name, after entering the name of the player the system displays the high scores. Then the system returns the main menu.

### 3.2.2Sequence Diagram

**Sequence Diagram for Start game**

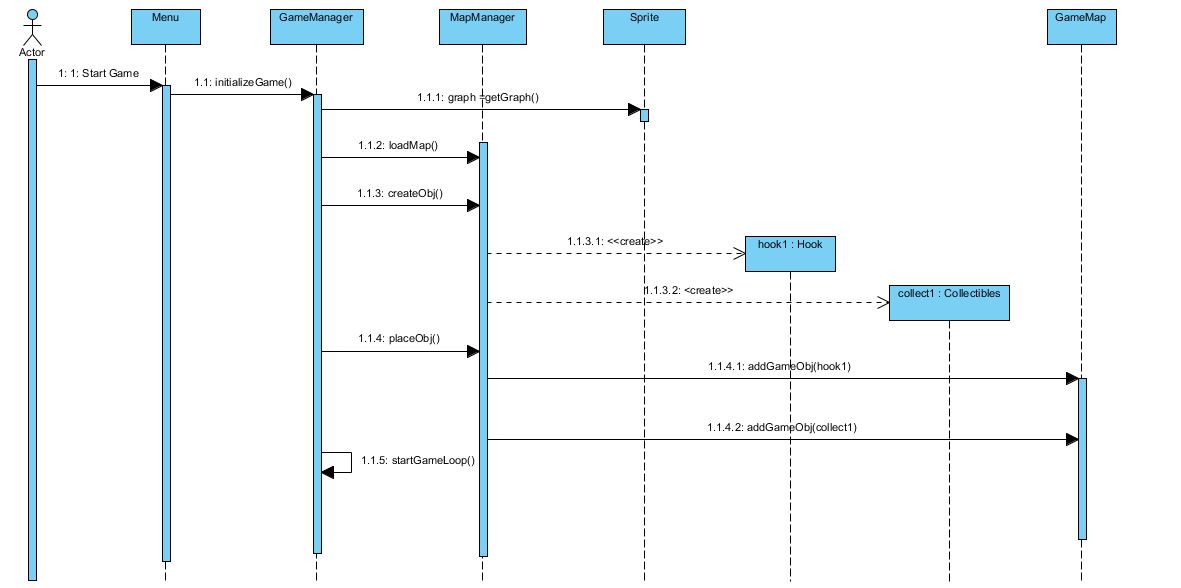


Figure 14 Sequence Diagram for Start Game

**Scenario for Start Game:** Player starts the game by pressing the start game button from the Main Menu. After the system get graphic images and objects to current window. System loads the current map by generating objects for the game and placing them randomly. Then system starts the game loop.

**Sequence of the Game Loop**

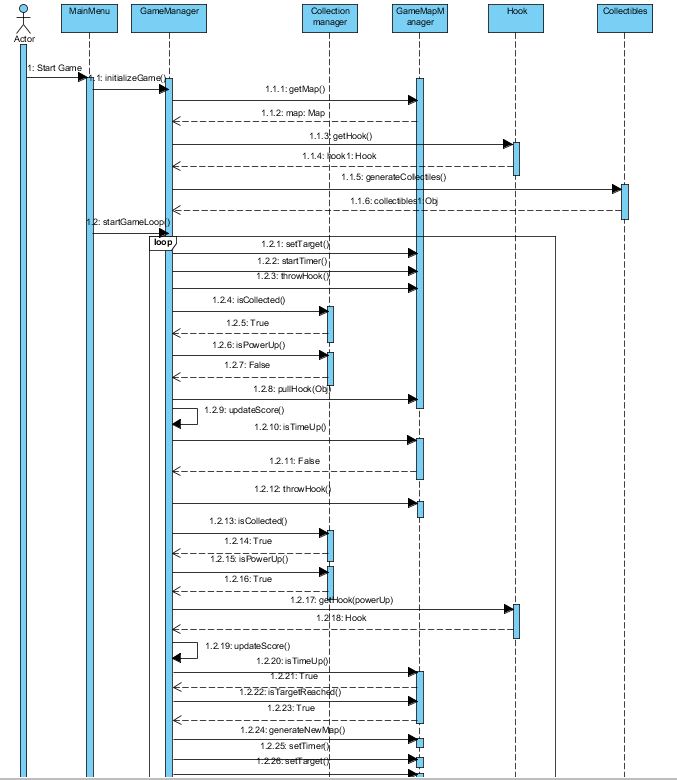


Figure 15 Sequence Diagram for the Game Loop

As seen above collection manager handles the collecting objects and communicates gameManager which manages the actual game dynamics by interacting with other control objects of the system

**Scenario for Game Loop:** Player starts the game by pressing the start game button from the Main Menu. First the system gets the map, hook and object to the map, and then we entered the game loop. Start of the game loop, the system set the remaining time and target point with the constant unchangeable values. Then player throw the hook where he want to throw, after throwing the hook, the system check if the location of the hook where we throw and location of the collectibles match. If it matches the system activates the collection manager which handles the collection of the collectibles. After that the system updates the score according to point of the collected object. If collected object is a powerUp system activate the special function for the powerUp and gets the new upgraded hook which decreases the time of the pullHook() function.

**Sequence Diagram for the Settings**

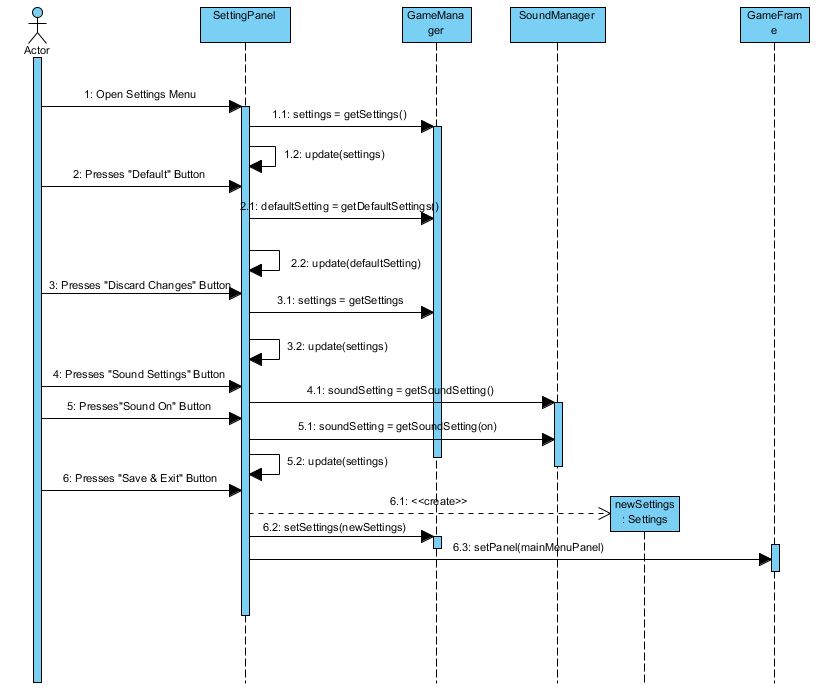


Figure 16 Sequence Diagram for the Settings

**Scenario for Settings:** This diagram shows how player can make changes setting of the game

Player enters the setting menu by using Settings button in the main menu. Player can press default button for default settings. Player can discard the changes by using the Discard Changes button and lastly player can change the volume level by volume setting. Then use save & exit to return main menu.

# 4. Conclusion

In this report, we designed analysis report of our project, called “Gold Miner”. We tried to explain what the game does, what functions it has. This analysis report has two parts. First part is requirement specifications. Second part is analysis part. In analysis part, we designed object model including class diagrams and dynamic model including state chart and sequence diagram.

In requirement specification part, we explained all requirements of game instructions. We tried to explain functional and non-functional requirements item by item. Constraints and scenarios were examined one by one. Lastly, we tried to give shape user interface and use case models according to constraints, scenarios and functional and non-functional requirements. To designing our use case models, we tried to specify which requirements and actions are really use cases.

In analysis part, we divided this part into 2 new parts. First one is “Object Model”. In object model, we tried to design and implement class diagrams and domain lexicon. Second one is “Dynamic Model”. In dynamic model, we tried to infer state chart and sequence diagram.

In class diagrams, we tried to perform perfect class model by examine possible classes and connections between them. We tried to show possible combinations of game process between users and system by sequence diagrams. All possible states were identified by state diagrams.

At the end we tried to create an analysis report of our “Gold Miner” game. This will be guide for our project’s design and implementation process. We tried to our best to avoid facing possible problems in future.