



# Bilkent University

**CS 319**

**Object-Oriented Software**

**Engineering Project**

**Analysis Report**

**Group 03**

**Mert Kutluca**

**Ahmet Namlı**

**Halil Selman Atmaca**

**Muhammed Emin Tosun**

**Instructor: Bora Güngören**

## **TABLE OF CONTENTS**

### **1) Introduction**

### **2) Overview**

#### **2.1) Game Play**

#### **2.2) New Weapons and Ammo**

#### **2.3) New Enemies**

#### **2.4) Extended Maps**

#### **2.5) Multiplayer Mode**

#### **2.6) High Score**

#### **2.7) Continue Game**

### **3) Requirements**

#### **3.1) Functional Requirements**

#### **3.2) Non-Functional Requirements**

#### **3.3) Pseudo Functional Requirements**

### **4) System Models**

#### **4.1) Use Case Diagram and Explanations**

### **5) Object/Class Diagram**

### **6) Dynamic Models**

#### **6.1) Sequence Diagrams**

##### **6.1.1)GamePlay Sequence Diagram**

##### **6.1.2)View High Score Sequence Diagram**

##### **6.1.3)Get Weapon Sequence Diagram**

#### **6.2)Activity Diagrams**

### **6.3)State Diagrams**

## **7)Interface Mockups**

### **7.1)Main Menu**

### **7.2)GamePlay**

### **7.3)Credits**

### **7.4)Help**

### **7.5)Pause Menu**

### **7.6)High Score**

## 1) Introduction

“More Dangerous Dave” is a new version of “Dangerous Dave” which is a popular game of MS-DOS games. We’ll almost originate all features of original game and “More Dangerous Dave” ’s gameplay, as it is seen, will be very similar to original “Dangerous Dave” game. In game objects like coins, bricks, chalices and diamonds will be located on “More Dangerous Dave”, and most of objects will be likened to original game. Some new features and objects will be added on game. Some basic weapons and guns, some new enemies and some new maps are examples of these new features. In our project’s most significant innovation is new multiplayer mode. In this mode, Dave and other player Davie fight against each other. Also, user can continue her/his game after he paused it. When adding these new features, we’ll take care about theme of original game and these features will be properly added.

Aim of the game is getting chalice then completing checkpoints. To complete checkpoints and getting chalices, player should jump over fires and weeds, and beat some enemies. When player complete one of the checkpoints, next checkpoint and map will be unlocked. At the top of the screen there will be some bars. They represent health, score, high score and level.

To upgrade the game, present better gameplay and supply course requirements, as we stated, new features and traits will be added.

- New Weapons and Ammo
- New Enemies
- Extended Maps
- Multiplayer Mode
- High Score
- Continue Game

The target platform of the game is JAVA FX. Only keyboard will be used in the game play, but in the pause and main menu also mouse will be used for controlling.

## 2) Overview

### 2.1) Game Play

In “More Dangerous Dave”, you are a man who is an adventurer. User plays in Dave’s dangerous and adventurous world. As we said, “More Dangerous Dave” is a game which inspired from a pioneer in game industry which named “Dangerous Dave” and in this game Dave spawn in a kind of room(map). In the map, there are some obstacles and some rewards. To complete level, player should cope with these obstacles. For example, user should jump on bricks and jump over fires, to get better score player should collect coins and other rewards, also some collectible guns and chalices will be placed on this room. Player can control Dave from keyboard by using arrow keys. To use gun and jetpack another assigned key will be used.

### 2.2) New Weapons and Ammo

“Dangerous Dave” game has only one type of gun it can be called as a magnum pistol. This gun can be collected like a coin and it has bullets. When you fire it, you cannot fire it again for a few seconds. When a bullet hit the enemy, enemy explode with a basic animation. We will upgrade this gun system. We will have two types of weapon, they are a pistol and a kind of sword. Player will use gun as ranged weapon and sword as melee weapon. This sword has a range far less from gun. We can calculate this range from distance between Dave and enemy. On the other hand, gun has a long-range but it is not infinite range. Also, bullets are limited. Ammo feature will be added on pistols and this restrict bullets. Ammo will be gained with pistol. Every gun item on map has a limited ammo. If player has a gun already, when he collects a gun his/her ammo will be added from collected gun. When gun fired, bullet’s height from ground decreasing according to distance which is covered by bullet. Also, bullet’s route, movement and gun’s re-fire will be arranged.

### 2.3) New Enemies

We will use completely different enemies from original “Dangerous Dave”. We will change all enemy logic of “Dangerous Dave”. In “More Dangerous Dave” there are four different enemy kind. When Dave collide enemies, Dave died and one health bar decreases. It is only way to die from “dumb enemy”. “Dumb enemy” is an enemy who is similar to his name. He walks around dumbly and when Dave collide him, Dave dies. One of other enemies

is an enemy who have a blade. This enemy approach to Dave when Dave entered to his range. His blade similar to Dave's sword. We have a similar enemy to the enemy who have a blade, this enemy has a gun like pistol. He fires his gun when Dave entered his zone. His gun is long ranged gun like Dave's pistol. Fourth of our enemies is boss enemy, he will confront to Dave at last map. He keeps on him all of other enemies' skills. Also, he is smarter than other ones and he won't die with a single bullet or sword blow.

#### 2.4) Extended Maps

"Dangerous Dave" game is an old game. It is perfect for its time, but it is for old technological systems. So it should be compatible to low memory systems. Because of that they use a kind of "static map". In their map system, Dave is in a room like map and he changes maps by using a door. We will change this map system by removing the borders of room. Dave can achieve his checkpoints by going right side of map and next map parts and levels will be located on there. This system can be assimilated to other famous game "Mario". Like "Mario" 's map, we will place Dave on the center of screen. When he moves maps will be harmonised to him. With making this feature, we will present better gameplay to user.

#### 2.5) Multiplayer Mode

As it stated before, multiplayer mode is most significant innovation in our project. In "Dangerous Dave" player can play only one single player mode. We will change this situation in "More Dangerous Dave". In main menu, there will be some options to player. Two of them are single player and multiplayer. When player select multiplayer mode, a "Davie" will be spawn as a second player. Control configuration of them, will be on different zones of keyboard. They will be play in a different but still similar map. In this map, there will be items on different and independent areas and this map will have borders with different from single player's map. "Dave" and "Davie" will compete for collect this items like weapons or coins. Also, they can kill each other with weapons. After this competition, player will be won who is still alive.

#### 2.6) High Score

In single player mode, we will add an high score parameter. This parameter keeps high scores on system. Whenever player play the game, he can see his/her high score. This

parameter will be reached on main menu and at top of the gameplay screen. When player enter from main menu to high score option he can see her high score list. This list contains a few number of high scores. But on top of gameplay screen, he can see only top of the list, in other words top score.

## 2.7) Continue Game

One of the other innovation is saving game. When player reach a checkpoint, “More Dangerous Dave” ‘s system saves this progress. If player exit from the game, system saves his last checkpoint, then when he selects continue game on the main menu he can play as from his last checkpoint. Also, system will save and load his score and health.

## 3) Requirements

### 3.1) Functional Requirements

#### Play Game

In this game, player controls a character whose name is Dave. Main mission of Dave is to take chalice in order to pass next level. Furthermore, Dave collects point while taking diamonds and coins. There are some weapons in the game Dave use these weapons to kill enemies. He should take heart to increase health point. He use jetpack to increase his speed.

#### Sound

In the game, player can turn on or off the sound of the game in the pause menu.

#### Help

In this part, player can learn anything related to how to play game. For example, there will be story of the game, there also be key’s information of usage.

#### Pause

In the play screen there will be a pause button on the right up button, which lead to pass pause screen. In this section, game is paused and there will be resume and exit button.

### High Score

This section shows the highest score, which is collected by player. The highest score will be kept in a text document.

### Credits

Coin is 50 points and Diamonds are 100 points. Total points will be shown in the play screen.

### Collision Detection

Every object will have a collision box. Thanks to collision box, collisions can be detected easily.

### Next Level

After Dave takes chalice, the door become enabled and when player object collide with door, the next level is generated.

### End of The Game

In the last level, Dave must kill the boss in order to finish the game.

## 2) Non-Functional Requirements

### Usability

Dangerous Dave is an easy game because there are few keys to play it. Moreover, structure of the game is very basic, even small a child can understand the logic of the game.

### Supportability

This game will be created by using java, which is appropriate for many platforms.

### Graphics

Interfaces are simple and efficient for user and object are distinguishable. In addition to simplicity, graphics are more colorful comparing to other Dave games.



## Performance

The game will have sufficient fps and run-time performance. Response of keyboards will be arranged in order to help players have fluent game experience.

## Extensibility

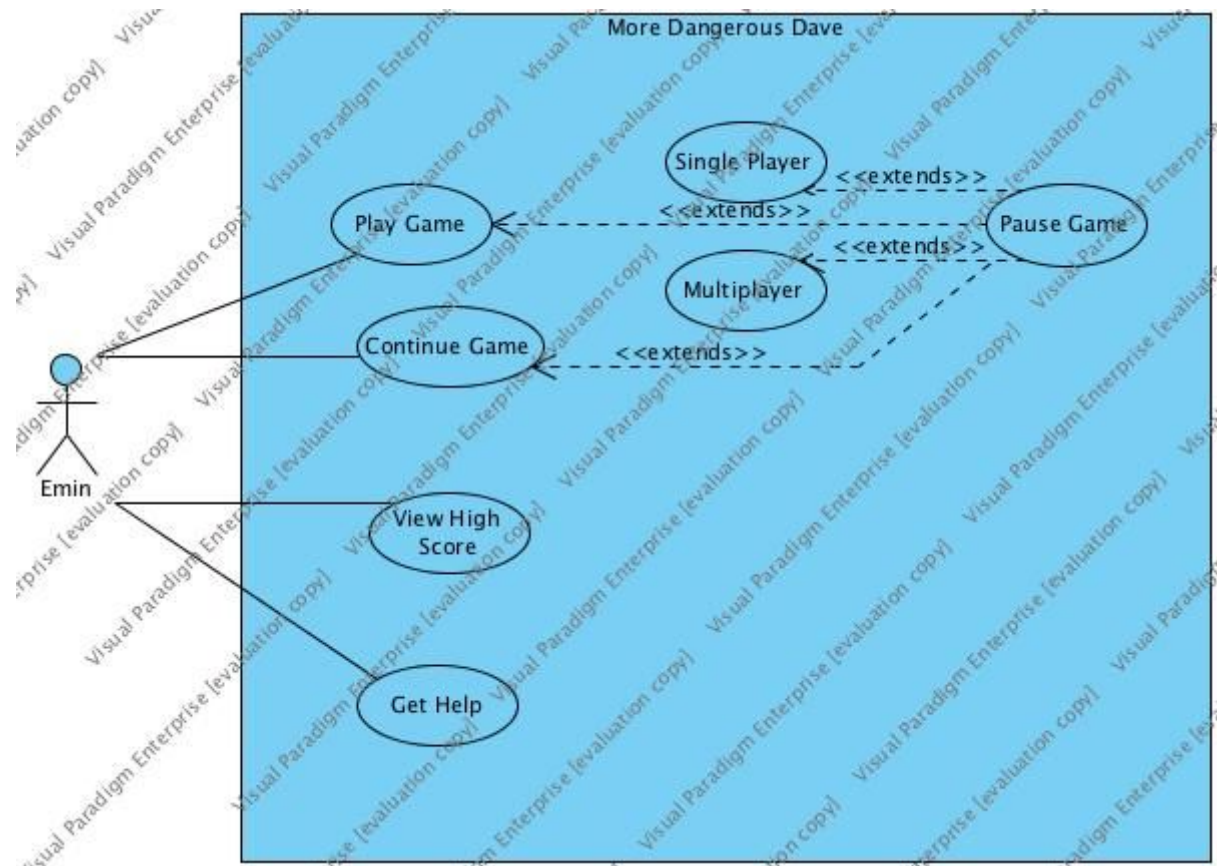
While designing Dave, being capable of adding new features to game will be considered. Especially sections of Dave are regarded as the most critical part of the game. New parts will be added easily.

### 3) Pseudo Functional Requirements

Only java will be used while designing this game. There is no need to used game engine. Furthermore, text documents will be used rather than database.

## 4) System Models

### 4.1) Use Case Diagram and Explanations



### **Use Case #1:**

**Use case name:** Play Single Mode

**Participating actor:** Player

**Entry Condition:**

-Player runs the application

**Exit Condition:**

-Player finishes the game.

-Player loses his all 3 lives.

**Main Flow of Events:**

1. Player opens the game.
2. Player clicks the Play Game button.
3. Player clicks the Single Player button.
4. Game loads the first level.
5. Player tries to finish the level.
6. Player get the final Chalice and finishes the level.
7. Player decides to quit.

**Alternative Flow of Events:**

1. If player can't get Chalice in three attempts, it jumps to 7th step.

### **Use Case #2:**

**Use case name:** Play Multi Mode

**Participating actor:** 2 Players

**Entry Condition:**

-Player runs the application.

**Exit Condition:**

-Player finishes the game.

-One of players loses his 3 lives.

**Main Flow of Events:**

1. Player opens the game.
2. Player clicks the Play Game button.
3. Player clicks the MultiPlayer button.
4. Game loads the battle map.
5. Players tries to kill each other 3 times.
6. One of the players kill the other one 3 times.
7. Player decides to quit.

**Alternative Flow of Events:-**

### **Use Case #3:**

**Use case name:** Continue Game

**Participating actor:** Player

**Entry Condition:**

-Player runs the application.

**Exit Condition:**

-Player finishes the game.

-One of players loses his 3 lives.

**Main Flow of Events:**

1. Player opens the game.
2. Player clicks the Continue Game button..
3. Game loads where the player saved the game last time.
4. Player tries to finish the level.
5. Player get the final Chalice and finishes the level.
6. Player decides to quit.

**Alternative Flow of Events:**

1. If player can't get Chalice in three attempts, it jumps to 7th step.

### **Use Case #4:**

**Use case name:** View High Score

**Participating actor:** Player

**Entry Condition:**

-Player runs the application

**Exit Condition:**

-Player back to the main menu.

**Main Flow of Events:**

1. Player opens the game.
2. Player clicks the View High Score button.
3. Game reads High Scores file.
4. Game shows the high scores.
5. Player backs to the main menu.

**Alternative Flow of Events:-**

### **Use Case #5:**

**Use case name:** Get Help

**Participating actor:** Player

**Entry Condition:**

-Player runs the application

**Exit Condition:**

-Player back to the main menu.

**Main Flow of Events:**

1. Player opens the game.

- ### Alternative Flow of Events:-

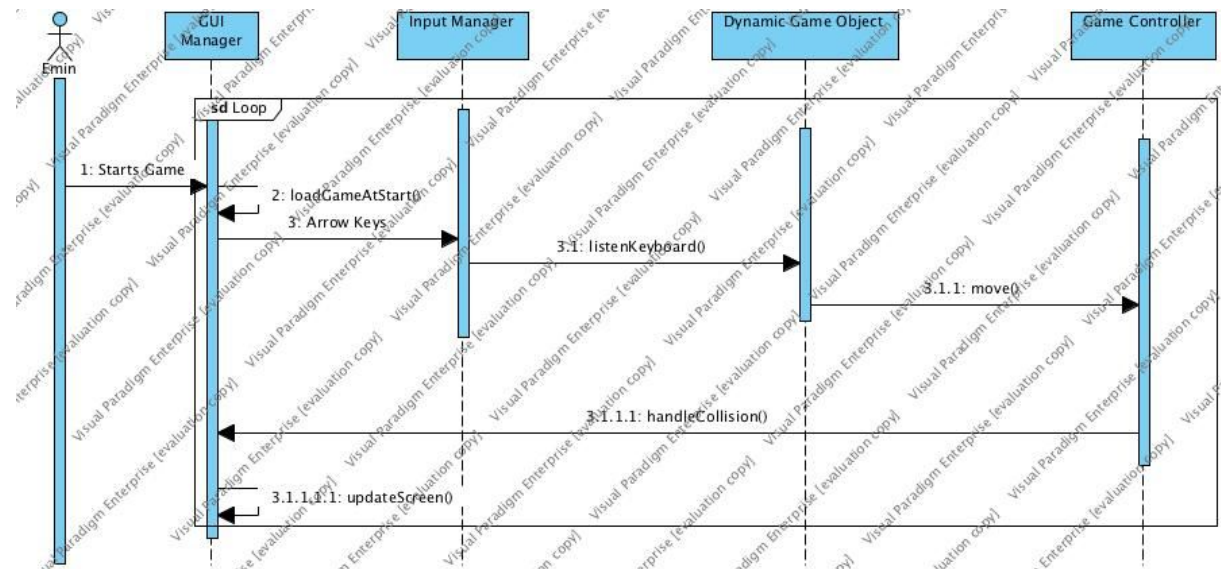
## 5) Object/Class Diagram



## 6) Dynamic Models

### 6.1) Sequence Diagrams

#### 6.1.1) Gameplay Sequence Diagram



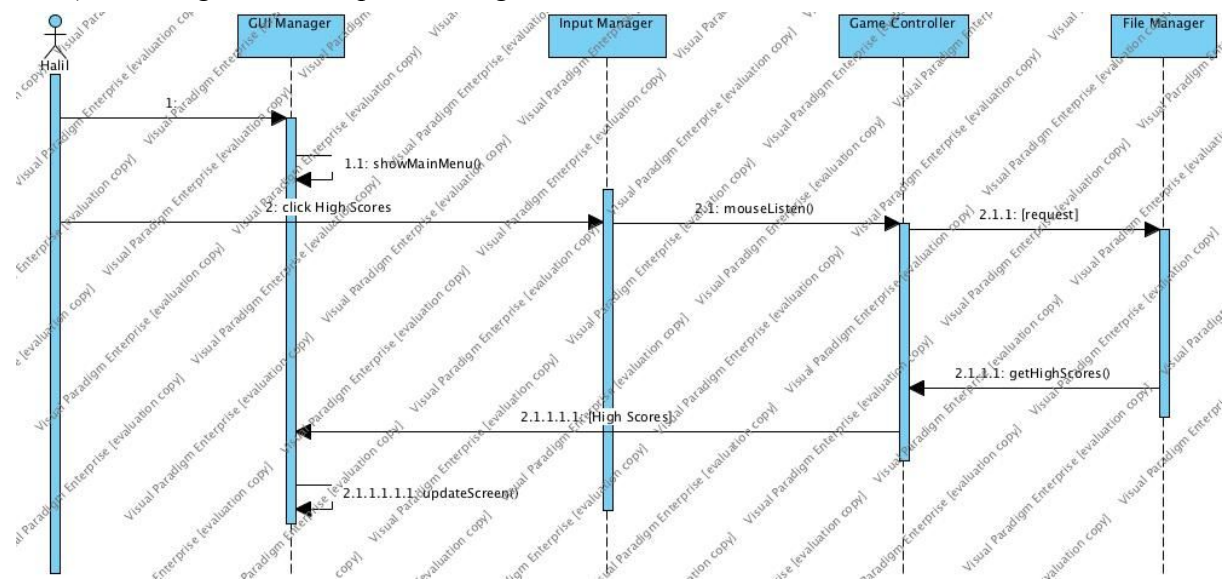
#### Scenario

Emin plays the game. Using arrow keys, he moves Dave. Dave stands on bricks and moves on bricks. He faces a poisoned brick and he loses one life.

#### Description

When Emin starts the game, game loads the game screen. Using arrow keys he sends inputs to Input Manager. Input Manager gets these inputs and sends them to Player object which is a Dynamic Game Object. Dynamic Game Object uses move() method in order to move Dave. Game Controller object checks whether there is collision or not. After that GUI Manager updates screen using updateScreen() method.

## 6.1.2)View High Score Sequence Diagram



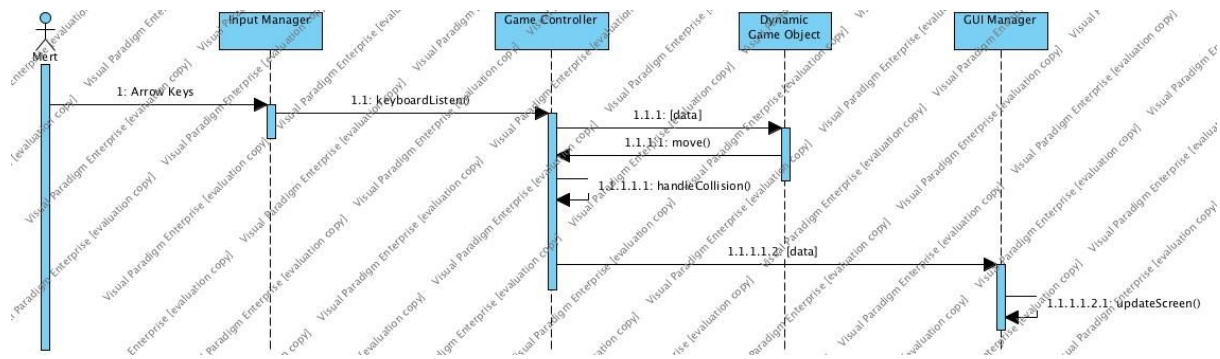
### Scenario

Halil wants to see his high scores in the More Dangerous Game. After opening the game he clicks the High Score button. Clicking the button retrieve him the high scores.

### Description

When Halil opens the game, GUI Manager uses showMainMenu() method in order to display main menu. Halil clicks the High Scores button. When he clicks, Input Manager uses mouseListen() method to perform this task. Game Controller object gets data from Input Manager and deliver it to File Manager. File Manager reads the high scores from file and deliver these data to Game Controller via getHighScore() method. Game Controller transfers this data to GUI Manager. GUI Manager uses updateScreen() method in order to show High Scores screen.

### 6.1.3)Get Weapon Sequence Diagram



#### Scenario

Ahmet is playing More Dangerous Dave. He sees a weapon and he wants to get it. He moves the Dave and gets the weapon.

#### Description

Ahmet uses arrow keys to get weapon. When he press an arrow key Input Manager takes it as an input. Via keyboardListen() method, data is transferred to Dynamic Game Object. Dynamic Game Object uses move() method to move Dave. Game Controller gets it as an input and execute handleCollision() method. This method creates some data. This data is delivered to GUI Manager. GUI manager updates the screen using updateScreen() method.

## 6.2) Activity Diagrams

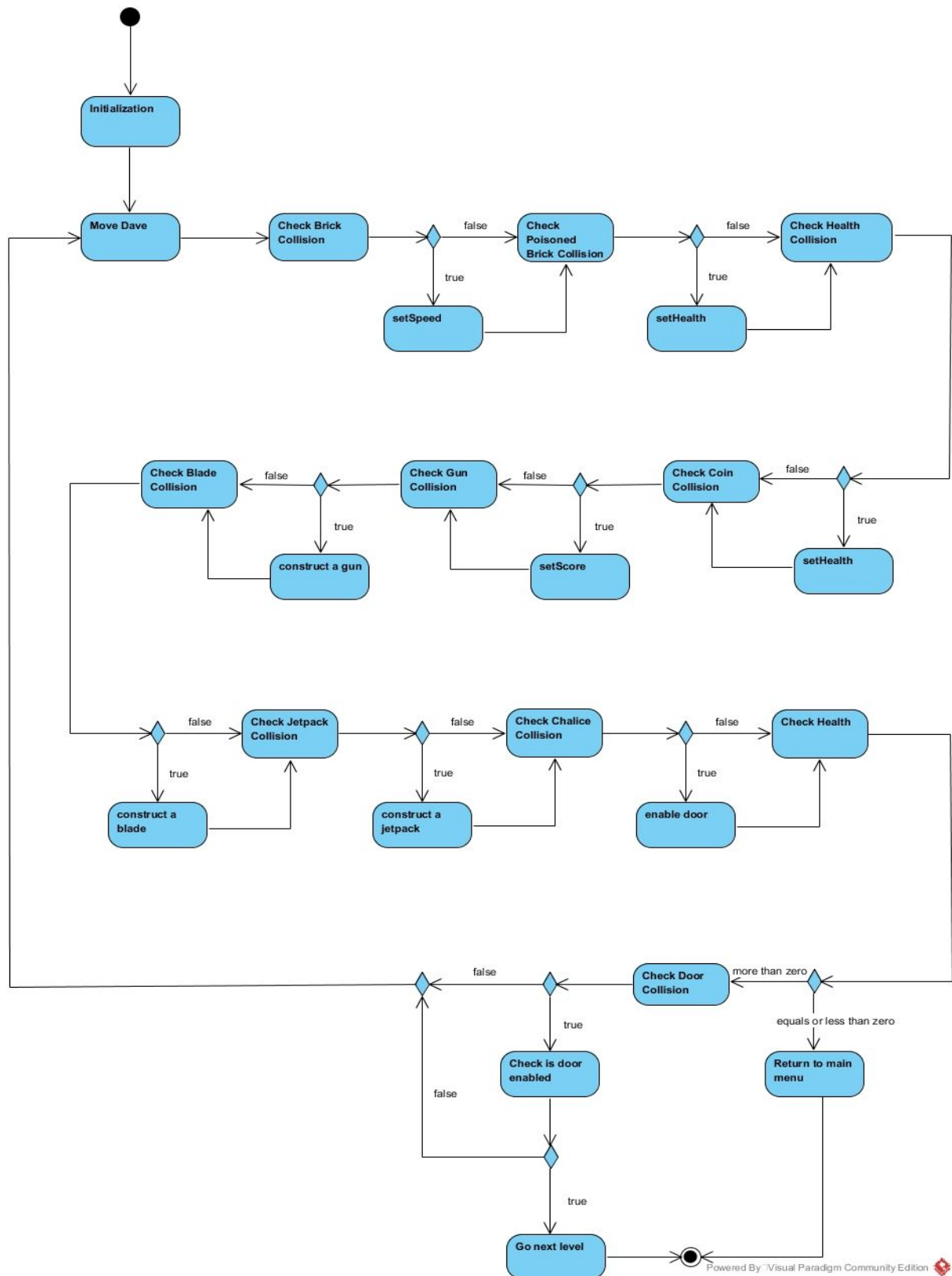
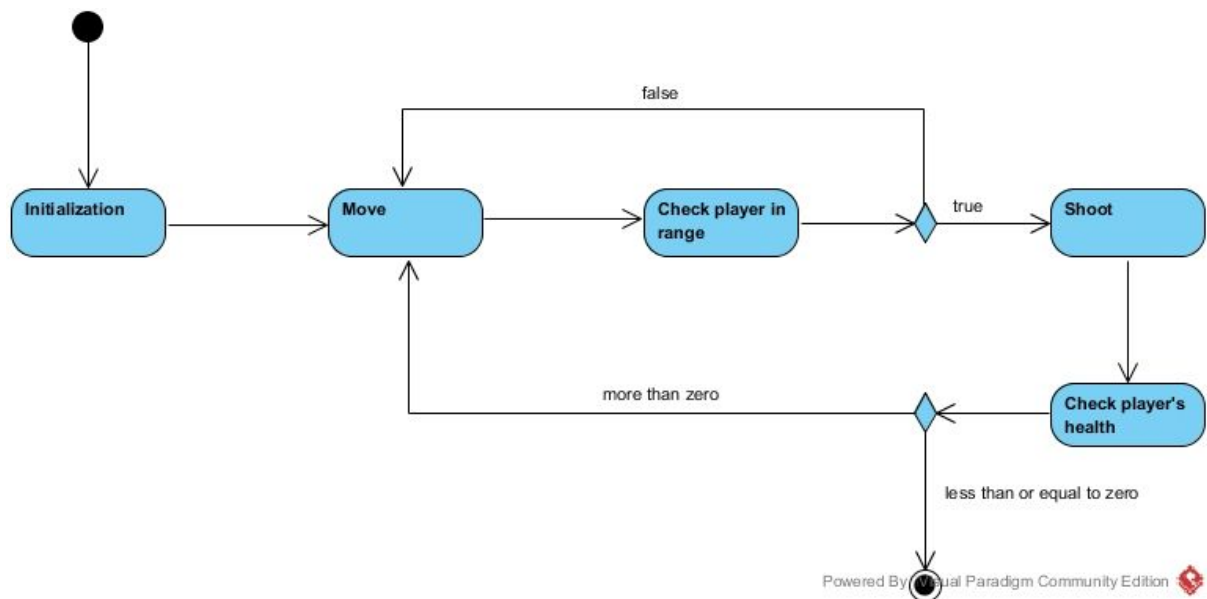


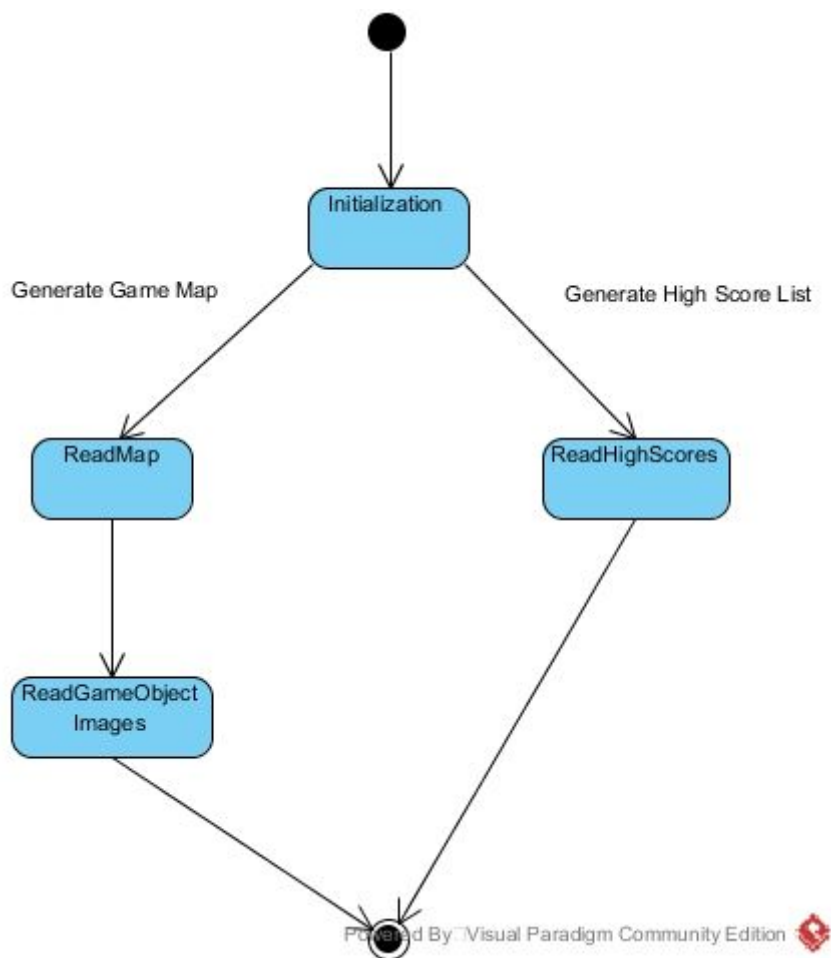
Figure 2: GameController handleCollision Activity





**Figure 3: Activity Diagram for EnemywithGun AI**

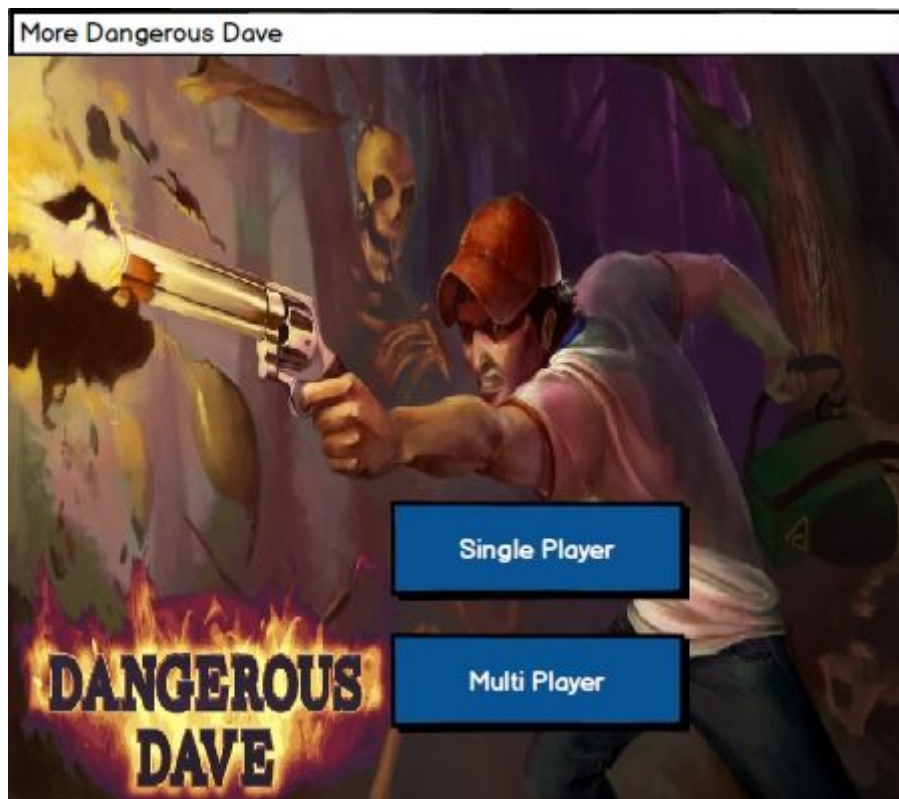
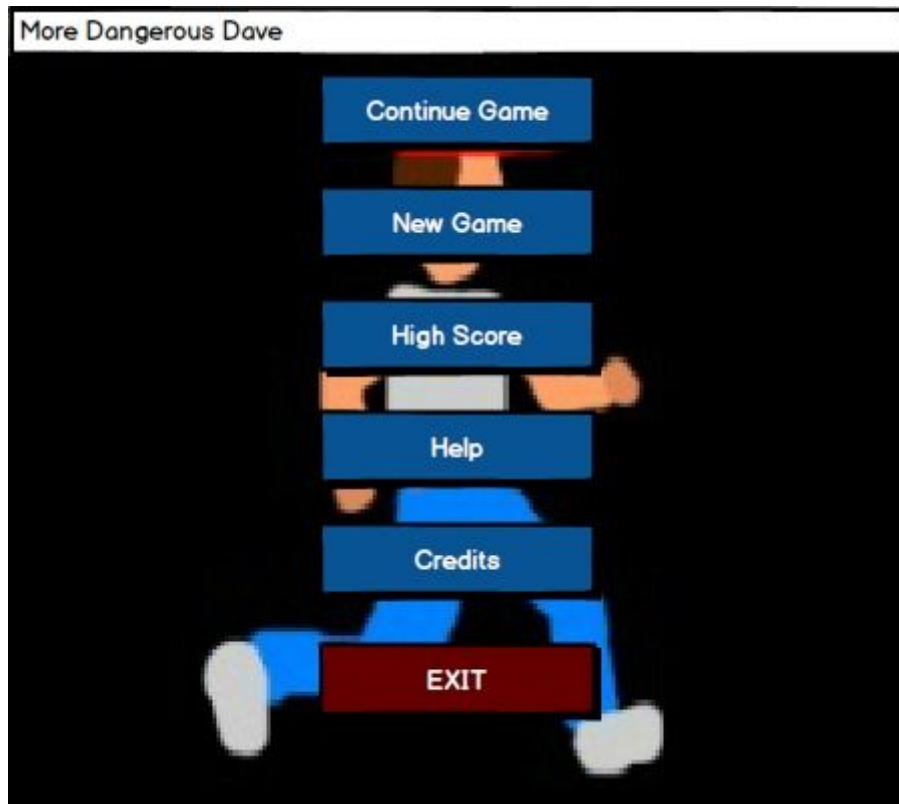
### 6.3) State Diagrams



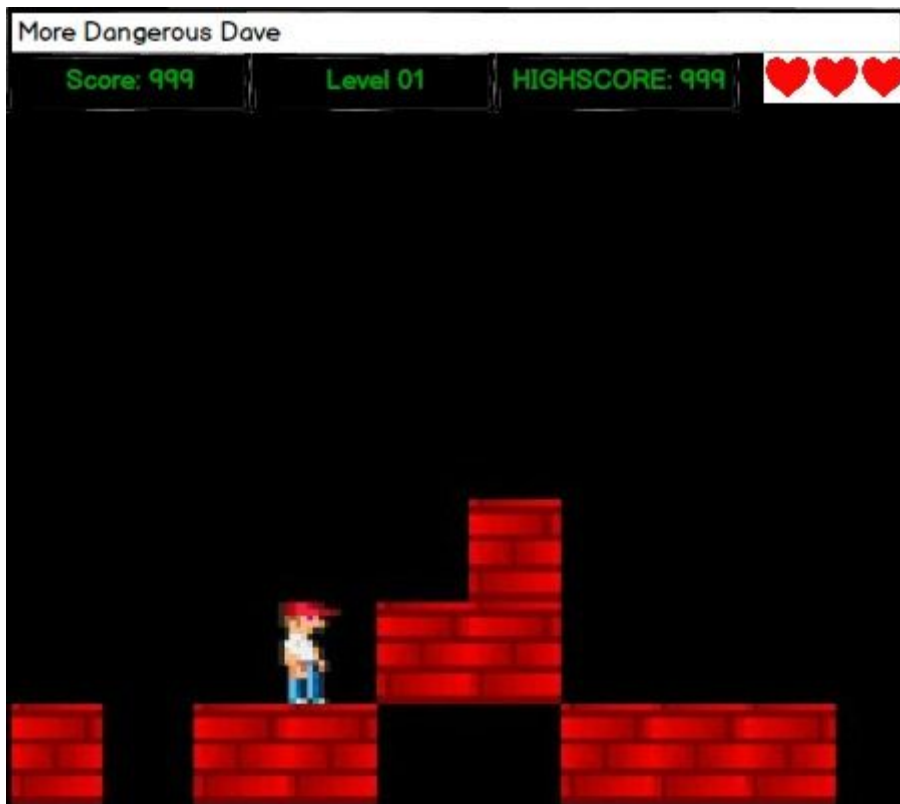
**Figure 4: FileManager State Diagram**

## 7) Interface Mock-ups

### 7.1) Main Menu



## 7.2) Game Play



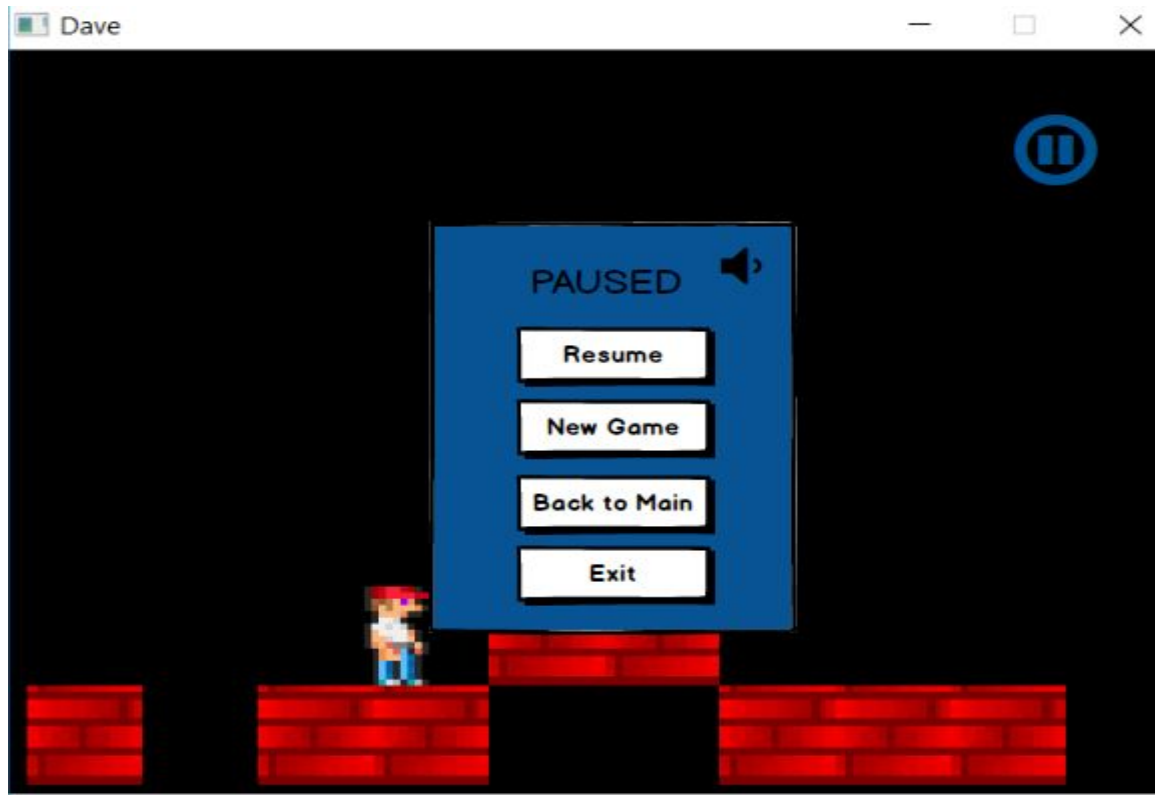
## 7.3) Credits




#### 7.4) Help




#### 7.5) Pause Menu



## 7.6) High-Score





# HIGH SCORES

RANK	SCORE
1 ST	6500
2ND	3500
3RD	2500
4TH	2000
5TH	1300