Object Oriented Software Engineering Project

CS 319 Project: Block Breaker Extreme

Analysis Report (Revised)

Kaan BİNGÖL, Mevlüt Türker GARİP, Esat Rıdvan KAVLAOĞLU, Bora YALÇINER, Selçuk YILMAZ

Course Instructor: Uğur DOĞRUSÖZ

Progress
November 30, 2011

This report is submitted to the Moodle in partial fulfillment of the requirements of the Object Oriented Software Engineering Project, course CS319.
Table of Contents

1. Introduction ........................................................................................................................................ 3

2. Game Overview ................................................................................................................................ 4
   2.1. List of Bonuses .............................................................................................................................. 4
   2.2. List of Bad Surprises to Player ...................................................................................................... 5
   2.3. List of Brick Types ....................................................................................................................... 5
   2.4. Stick ................................................................................................................................................ 7
   2.5. Ball ................................................................................................................................................ 7

3. Requirements ...................................................................................................................................... 8
   3.1. Functional Requirements ............................................................................................................... 8
   3.2. Non-Functional Requirements ..................................................................................................... 8
   3.3. Pseudo Functional Requirements ................................................................................................ 8

4. System Models .................................................................................................................................... 9
   4.1. Use Case Model ............................................................................................................................ 9
       4.1.1. Use Case Descriptions ............................................................................................................ 10
             Use case #1 .................................................................................................................................. 10
             Use case #2 .................................................................................................................................. 11
             Use case #3 .................................................................................................................................. 11
             Use case #4 .................................................................................................................................. 12
             Use case #5 .................................................................................................................................. 12
             Use case #6 .................................................................................................................................. 13
             Use case #7 .................................................................................................................................. 13
   4.2. Dynamic Models .......................................................................................................................... 14
   4.3. Object and Class Model ................................................................................................................. 20
   4.4. User Interface – Navigational Paths and Screen Mock-ups .......................................................... 22

5. References .......................................................................................................................................... 27
Table of Figures

Image 1 - Clay Brick’s Appearance ...................................................................................................................... 6
Image 2 - Normal Stone Brick ............................................................................................................................... 6
Image 3 - Cracked Stone Brick .............................................................................................................................. 6
Image 4 - Iron Brick .............................................................................................................................................. 6
Image 5 - Explosive Brick .................................................................................................................................... 7
Image 6 - Appearance of the Stick ........................................................................................................................ 7
Image 7 - Appearance of the Ball ........................................................................................................................ 7
Figure 1- Use Case Diagram ................................................................................................................................ 9
Figure 2 - Sequence Diagram of Executing Game ............................................................................................... 15
Figure 3 - Sequence Diagram of Play Game ....................................................................................................... 17
Figure 4 - Sequence Diagram of Acquiring Bonus ............................................................................................. 18
Figure 5 - Activity Diagram of the Game ........................................................................................................... 19
Figure 6 - Class Diagram of the Game ................................................................................................................. 20
Figure 7 - Navigational Path of the Game ......................................................................................................... 22
Figure 8 - Main Menu ......................................................................................................................................... 23
Figure 9 - Change Settings ................................................................................................................................ 24
Figure 10 - Game Screenshot ............................................................................................................................ 25
Figure 11 - In-Game Menu .................................................................................................................................. 26
1. Introduction

We decided to design and implement a game called Brick Breaker Extreme. It is a basic block breaking game that player tries to clear the all bricks with controlling the stick and player tries to aim the ball towards to bricks by using the stick.

This report contains an overview of the game, describes the basic gameplay elements and rules of the game. Then it describes functional requirements, non-functional requirements, use-case’s models including scenarios and use-case diagrams.
2. Game Overview

The game consists of three levels where the pattern's bricks are designed differently accordingly. There will be a time limit of 4 minutes and remaining time will be shown with the counter. Purpose of this game is completing all the levels without losing all lives. Basically, the game revolves around destroying bricks using stick and ball. While trying to finish the level, player can benefit various bonuses. On the other hand, player can also encounter with bad surprises additional to the bonuses. Besides, there are four types of bricks which will differ from each other in the effects when collision occurs.

During the game, the player can pause at any time and continue afterwards. Additional to this feature, the game can also be saved and loaded in order to continue from where it is saved after even closed.

There will be high-score system which will keep the highest five scores with players’ name. Player gains scores according to the type of bricks therefore the bricks will give points according to its hardness to break. Time bonus which is the multiplication of remaining time in seconds by 1000 will be added to the score gained from breaking the bricks. Therefore, this prevents players that completed the game from getting the same score at the end.

The players will be able to control the game in two different ways. They can control the stick either using mouse or keyboard.

Players are allowed to change game difficulty using the settings menu. There will be two difficulty levels, hard and easy. In the hard difficulty there will be fewer bonuses, more bad surprises and more bricks than easy difficulty.

2.1. List of Bonuses

- **Long Stick**: When player gets the long stick bonus, stick length increases by original stick length. Therefore, player will be able to catch the ball more easily. This bonus can be gained twice in each level.

- **Laser Gun**: With this bonus, player can fire to the bricks with laser gun while the player is still trying to catch the ball. Laser gun is used by either left mouse button or spacebar key in the keyboard. It lasts 10 seconds.

- **Fire Ball**: This bonus fires up the ball so that it will not reflect from any bricks and keep destroying the bricks on its path. Fire ball lasts 5 seconds.

- **Ball Magnet**: After this bonus, the stick will be magnetized therefore it will be able to hold the ball and player can release the ball again by using left mouse button or space
key. Choosing the same keys to release the ball with the keys to fire the gun is intentional. Hence, this prevents players from holding the ball at all times while firing the gun. Ball magnet lasts 10 seconds.

- **Slow Ball**: The ball starts to move slower once the player gets the bonus. It lasts 15 seconds. This allows players to control the ball more easily.

- **Triple Ball**: This bonus blows up the original ball and triplicates the ball. This provides faster clearing of the bricks and continues until one ball is left. Losing any other two balls will not cause the players to lose their lives.

- **Shield**: It creates a shield below the stick so that the ball will not fall down and this continues as if the stick reflected the ball. Purpose of this bonus different than the life bonus is that player can collect any bonuses while not concerning about the ball.

- **Extra Life**: Just like other games, its purpose is to provide an extra life.

**Important Note**: When the ball falls below the stick level, all active bonuses will be cancelled.

### 2.2. List of Bad Surprises to Player

- **Short Stick**: It operates opposite to Long Stick bonus. The stick’s length will be reset to its original size if there is already a Long Stick bonus active otherwise gets smaller to half of its length.

- **Fast Ball**: This counters Slow Ball bonus. The ball’s speed will return to its original speed if there is already Slow Ball bonus active otherwise starts to move faster than its original speed, which makes handling of the ball harder.

- **Neutralizer**: It cancels all active bonuses on both the stick and the ball.

- **Instant Death**: Players directly lose one of their lives as if the ball falls below the stick level.

### 2.3. List of Brick Types

- **Clay Brick**: It is the brick that can be broken by one collision with the ball. Player gets the least score from this brick, which is 1000.
- **Stone Brick**: This brick has double durability than clay brick therefore it requires 2 collisions with the ball, which gives the score of 2000.

- **Iron Brick**: It can only be broken by Fire Ball but finishing the level does not depend on the clearance of this type of brick.

- **Explosive Brick**: When it explodes, it destroys all neighboring bricks as well as itself. Player only gets the scores from adjacent bricks.
2.4. Stick

Stick is main source of interaction between game environment and the player. Player can move the stick horizontally and try to bounce the ball in order to break the bricks.

2.5. Ball

Ball is one of the main components of the game. Player can bounce the ball with the stick and try to break bricks. If the ball falls below the stick, player loses his/her life.
3. Requirements

3.1. Functional Requirements

- User will be able to control the stick either with mouse or keyboard.
- User can change the settings of the game. Available settings are:
  1. Changing the background color
  2. Changing the input keys.
  3. Changing the difficulty.
- User will be able to access the help menu. Help menu contains information about game dynamics and game elements such as purpose of the game bonus descriptions and default controls.
- The game has a high score system which allows the user to see the top 5 high scores.
- The game will offer various power-ups and bad surprises to player during the game sessions. They can be either ease or complicate the game.
- Player will be able to pause the game and resume the game afterwards.
- The game can be saved by the player. Saving operation stores the changes in the level. In order to resume the game, players can load the game where they left off.

3.2. Non-Functional Requirements

- Game graphics will be designed in order to be appealing. There will be no flickering and movement of the ball and the stick will be smooth. Images of graphical objects will be chosen to give the players better gameplay experience.
- Control mechanisms of the game will have low response time, which enables users to play with minimal delay.
- Main plot of the game is easy to apprehend.

3.3. Pseudo Functional Requirements

- The Game will be implemented in java.
- Some of the graphic objects will be designed using Adobe® Photoshop CS4.
4. System Models

4.1. Use Case Model

System: Block Breaker Extreme

Figure 1- Use Case Diagram
4.1.1. Use Case Descriptions

Use case #1

Use case name: PlayGame

Participating actors: Player

Entry condition: Player has already opened the game and is on main menu

Exit condition:
- Player has completed all the levels successfully, OR
- Player has lost all of his/her lives in any level, OR
- Player has chosen to exit the game via in-game menu.

Main Flow of Events:
1. Player starts the game
2. The system constructs the level.
3. Player finishes all the levels
4. The system displays score of the player. If player score is one of the top five scores, it asks player to enter his/her name.
5. Player enters the name.
6. The system records the score and modifies the high score table.
7. Player returns to the main menu.

Alternative Flow of Event:
- Player loses all his/her lives at any level. (display game-over message, go to step 4)
- At any time player can exit the game. (go to step 7)
Use case #3

Use case name: ChangeSettings

Participating actors: Player

Entry condition: Player has already opened the game and is on main menu

Exit condition: Player returns to the main menu.

Main Flow of Events:

1. Player chooses to view settings menu.
2. The system displays different adjustments on the gameplay.
3. Player configures the settings according to his/her comfort.
4. The system saves corresponding changes to HDD in order them to be future gameplays.
5. Player returns to main menu.

Alternative Flow of Event:

- Player does not change any settings. (go to step 5)

Use case #2

Use case name: ViewHelp

Participating actors: Player

Entry condition: Player has already opened the game and is on main menu

Exit condition: Player returns to the main menu.

Main Flow of Events:

1. Player choses to view help menu.
2. The system displays the help documentation about the game.
3. After reading the documentation, player returns to the main menu.
Use case #4

Use case name: ViewHighScore

Participating actors: Player

Entry condition: Player has already opened the game and is on main menu

Exit condition: Player returns to main menu.

Main Flow of Events:

1. Player choses to view High Score menu.
2. The system retrieves high scores from HDD and displays them on the screen.
3. After seeing the high scores, player returns to the main menu.

Use case #5

Use case name: LoadGame

Participating actors: Player

Entry condition:

- Player has already opened the game and is on main menu or Player is in-game menu, AND
- There is already an available saved game.

Exit condition: Player starts to play the game from the saved point.

Main Flow of Events:

1. Player clicks on the load-button.
2. The system retrieves saved game data from HDD and initializes game state accordingly.
3. Player continues the game from where he/she left.

Alternative Flow of Event:

- If there is no saved game load button will be disabled. (warning massage will be shown)
Use case #6

Use case name: PauseGame

Participating actors: Player

Entry condition: Player is already playing the game.

Exit condition:
- Player continues to play the game, OR
- Player can choose to return to the main menu, OR
- Player can exit from the whole game, OR
- Player can load another game, if there is already a game saved.

Main Flow of Events:

1. Player presses the pause button during the game.
2. The system displays the in-game menu.
3. Player continues the game.

Alternative Flow of Event:
- Player choses to save the game and returns to the in-game menu.
- Player choses to return to the main menu.
- Player directly exits the game instead of continuing the game.
- Player loads another game. (go to step 3) If there is no saved game load button will be disabled and warning massage will be shown.

Use case #7

Use case name: SaveGame

Participating actors: Player

Entry condition: Player is in-game menu.

Exit condition: Player returns to in-game menu.

Main Flow of Events:

1. Player clicks on the save-button.
2. The system writes the level data to HDD and displays the massage that shows the status of the process.
3. Player returns to in-game menu.
4.2. Dynamic Models

**Scenario Name:** Execute Game

**Scenario:** Ali double clicks the game icon in the desktop. Then, he sees the main menu and chooses the “Settings” button. In the settings menu, he configures the background image by selecting one of the images shown in the list. Afterwards, he returns to main menu and clicks the “New Game” button to start playing a new game.
Figure 2 - Sequence Diagram of Executing Game
**Description:** After player executes the game; game settings, high scores and help document is loaded by “FileManager” from HDD. After this process, game waits for player’s next instruction. This instruction can be “New Game”, “High Score”, “View Help”, “Settings” or “Exit Game”. However, in this scenario, player only chooses “New Game” and “Settings” options. When “New Game” is clicked, “GameEngine” gets the corresponding level data via “FileManager” from HDD and draws the first frame of the level accordingly. All pre-fetched game data before the level starts will be stored back to the file system with any modifications when player clicks “Exit Game” in the main menu.

**Scenario Name:** Play Game

**Scenario:** Player Ahmet has already finished all other levels and started the last level of the game. He throws the ball by pressing the key configured as “ActionButton”. He destroys the last brick in the last level and completes the game successfully. Ahmet’s score happens to be in the top 5 so he enters his name to be saved with his score.
Figure 3 - Sequence Diagram of Play Game
Description: After the player pressed the actionButton to throw the ball from stick, the main game loop starts to execute. In every cycle, game loop updates the ball and the stick position then checks if any collision happened between the ball and bricks on the map. Finally, the game loop draws the game objects accordingly to the screen.

After the game is over, “Game Engine” calculates the final score by adding the current score with a fraction of the remaining time. If the score is in the top 5, “GameEngine” opens a pop-up window to ask the player to enter his/her name and updates its high score table.

Scenario Name: Acquiring Bonus

Scenario: Melissa has already started playing the level. Ball hits a brick which drops lengthen bonus. She moves the stick towards the bonus and collects it. Stick gets longer.

Figure 4 - Sequence Diagram of Acquiring Bonus

Description: When a brick destroyed, random bonus generating function decides to whether bonus will be created or not. When bonus taken by player, corresponding updates will occur.
Figure 5 - Activity Diagram of the Game
Activity diagram shows the main flow of the game. In every cycle, “GameEngine” will move the ball and stick concurrently. Then, it checks any collision between the stick-ball, brick-ball and brick-bullet. Meanwhile, during the flow, “GameEngine” terminates the game whenever the time is up or the player is out of lives.

4.3. Object and Class Model

Figure 6 - Class Diagram of the Game

In our design, “MainMenu” is the first class that will be constructed by the main method. However, “GameEngine” is at the center of all class interactions. For example, even for loading the saved game data, “MainMenu” calls the functions of “GameEngine” to interact with the “FileManager”.

During flow of the program, “InputManager” is the class responsible for all user interactions.
While “GameEngine” draws the game objects to the screen, “BrickMapManager” is the storage class that contains the brick map and array of “Brick” objects that have not been destroyed yet.
4.4. User Interface – Navigational Paths and Screen Mock-ups

Main menu is the first thing that is shown when the game opens. In main menu, player can choose to play the game, load the previously saved game, change settings, view the help document, view high scores or terminate the program.

Play Game: When this button is pressed, program starts a new game with the specified settings and map.

Load Game: When player hits the load game button, the game starts with the previously saved game data.

Change Settings: This interface provides several customizations of game experience. Player can change difficulty level, background image and all keyboard controls.

View Help: When this button is pressed, help document will be shown to give the player information about the game play.

View High Scores: Player hits this button to see top 5 scores achieved in the game.
Quit Game: This option terminates the program and writes back all required information that will be saved from memory to hard disk.

![Main Menu](image)

**Figure 8 - Main Menu [6]**

Change settings menu allows the player to change background image of the game, difficulty level and keyboard controls.

Difficulty Level: There are two levels such as easy and hard. Number of bonuses, bad surprises and bricks depends on the difficulty level. If not selected, default level will be easy.

Background: There will be 5 different background images consisting of space and landscape images.

Keyboard Control: The player can change the keys for moving the stick, fire and throwing the ball. Default values are R-arrow for moving stick right, L-arrow for moving stick left and space key for firing or throwing the ball.
Figure 9 - Change Settings [6]
In the game screen, the player can see the number of lives left, current level, score and the time left to finish the level before the game is over.

![Game Screenshot](image)

Figure 10 - Game Screenshot [7]

This menu is very similar to the main menu except the player is not allowed to change settings during the game. Besides, instead of load game button there is save game button that will save the current game data to hard disk to continue at a later time. Loading a saved game during the game is not allowed. Load button only exists in the main menu.
Figure 11 - In-Game Menu [7]
5. References


