Bilkent University
CS 319

Bombplan

Analysis Report

Group #1

Asena Rana Yozgatli – 21000132
Berk Yurttas – 21200581
Mehmet Furkan Sahin – 21201385
Saner Turhaner – 21100475

Ugur Dogrusoz

Table of Contents

1. Introduction

Bombplan is a brand-new version of classical Bomberman game. The main aim in Bomberman is to reach to exit door without being killed by the monsters in the game. The main character has the power to plant the bombs in different locations on the map. The map is in form of a maze and there are some wall blocks that can be destroyed by the bombs and some others that can never be destroyed. The character starts from one end of the maze and by destroying the suitable walls and running away from the monsters, tries to reach to the exit door. The bombs are not only affective on the walls, if there is the main character or one of the monsters in the range of the bomb, they can also be destroyed.

In this report, the main design of the Bombplan will be discussed. We will provide Bombplan with an OOP implementation. The following sections will lead to discussions related to requirement analysis, and the analysis with object models and dynamic models. During the requirement analysis part we will examine functional and non-functional requirements, constraints, scenarios, use case models, and user interface of Bombplan. At the end, we will conclude with a general revision of the whole report.

2. Overview

Bombplan is a desktop application that will be created with Java. Game instructions will be as following;

- The main character (from now on, it will be referred as bomber) will have only the ability to plant a bomb.
- A bomb has a range of one wall block from 4 sides. The waiting time for the explosion of the bomb is 3 seconds, default. The bomb features can be changed by the taken

tokens that are randomly distributed inside of the walls on the map. The features that can be changed are as following;

- Bomb explosion can be made up to the user. By the help of a taken special bonus, bombs do not explode themselves in 3 seconds; instead, they wait for the user to explode it.
- The range of the bomb is 1 wall block from 4 sides -default-, but it can be extended by a special token. After the token, the range increases 1 more wall block in depth from 4 sides. Since, user can take from that special token more than once, at some point, range can be 4 wall blocks in depth from 4 sides.
- Normally, user cannot plant one more bomb if there is already a planted bomb. However, by the help of another token, user can plant multiple bombs. Each token increases 1 the maximum number of the bomb that can be planted at once.
- There is a special token that different from the other three tokens. After this special token is taken, it gives a random token to player.
- There is also a token that does not affect bomb. It resets the timer after player takes it.

- Bomber has 3 lives at starting. This can be increased with hearth shaped tokens by one and of course, it decreases by dying.
- Bomber can die by being in the range of an exploding bomb, fail to kill all of the monsters and find the door in the given time or touching to one of the monsters.
- There are 2 types of monsters -slow and fast- in the game. The number of the monsters from these types changes according to the level. Total number of monsters also changes according to the level.

- There are 2 types of walls in the game. One of them can never be destroyed by bombs, the other can. Their design changes according to the type of the wall and of course all of the tokens are in the walls that can be destroyed.
- Number of blocks will be definite for each level. However, the blocks will be distributed randomly on the map. Of course there will be some constraints for the distribution of the walls to provide a more homogenous map.
- To pass a level, bomber need to kill all of the monsters and find the secret door inside of one of the wall blocks in the given time.
- To finish the game, bomber should pass 3 levels of maps or spend all of his lives.
- During the game, user earns some points with destroying walls, killing monsters, taking tokens or passing levels. At the end of the game, the collected point is put in high-score list if it is one of the highest 10 scores ever played on that computer.

3. Functional Requirements

New Game

The player will start to play a game by choosing “New Game” from the main menu of the game. The game will start from the very first level that is already prepared by the developers.

Play Game

The game starts by locating the main character to the top left corner of the map to each level. The map is in form of a maze and its difficulty changes according to the level. The monsters in the field distributed and move randomly. Speed of the monsters again changes according to the level and different type of monsters will be seen in high levels.

Their only aim is to touch to the main character and kill him. The main character should kill all of the monsters in the map before losing all of his lives. By default, the main character will start to the game with 3 lives that will give him a chance to die 3 times before loosing the game. To kill a monster, bomber should plant a bomb to a location that possibly a monster will be in the range of it. With only one bomb, it is possible to kill multiple monsters if they are all in the range of the bomb. Additionally, bomber can be stuck in an area but there will be some walls that can be destroyed by bombs and some others that cannot. By destroying the walls, bomber can open a way for himself. In the same time, bonuses will be hidden in these destroyable walls randomly. To reach a bonus, bomber should first destroy the wall. A bomb can both destroy a wall and kill some monsters if there is any in the range of it in the same time. To finish a level of the game successfully, bomber should both find the secret door that is hidden in one of the destroyable walls and kill all of the monsters. Even if he finds the door, it does not allow the bomber to pass if any one of the monsters is still alive. After passing from the door, the game will load the next level if there is one. If the user passes the last level, or die before succeeding in the game, he will see a menu if his score is one of the highest 10 and he will be able to save his score with a name.

Save Game

In any time of the game, user will be able to save the current state of the game with a name so that he can continue after a while if he needs to quit the game in that moment.

Load Game

The user will be able to see his saved games by choosing “Load Game” from the main menu and we will list his saved games. After he choses one of the saved games, we will load it to the game.

Pause Game

The user will be able to pause a game in anytime and do his urgent job if he should. The game will be paused if he pushes the button “Esc” from his keyboard or choose pause button from the screen of the game. He will see a push menu after pausing the game and there will be the list of Save Game, Continue Game, Help, Return to Main Menu, and Quit.

Help

For a user who is completely new to the game, help menu is crucial. Thus, it should be visible to him before starting to play a game. Additionally, it should be reachable while the user playing the game since he might want to check what a bonus is or simply how to move the character.

In the help menu, we will provide the information on the main aim generally, abilities of the character, bonuses, how to use them, and how to move the character.

High Scores

The user will be able to see the high scores in the game by choosing High Scores from the main menu. High scores will be given in form of a table with their names. If there is no saved high score, it will give an appropriate message.

Options

Options menu will be reachable from both main menu and pause menu. Through the options menu, user will be able to set character type, music (on/off), and effect sounds (on/off).

Credits

Credits on the game will be presented in “Credits” that is available on the main menu. It will include developer information, library names and version number.

Quit

From main menu or pause menu, user will be able to quit the game by choosing Quit. If he did not save his progress before quitting, it will be lost.

4. Nonfunctional Requirements

4.1 Performance Requirements

- In order to have a smooth game flow, the software will handle the amount of throughput that is necessary for continuity of the game.
- Response time will be low enough so that it will never surpass 1 second.
- Highscores will be saved (written) to the txt file. Reading this data and creating highscore table will be fast and easy.

4.2 Reliability Requirements

- The game will be robust. It will not give any exceptions to the wrong inputs, empty highscore list and corrupted previous option preferences.

- The software will not loose any data; option preferences, listed highscores or saved games. It will store them in files.

4.3 Supportability Requirements

- The game will be executable for every platform that Java works.
- The software will not require any installation process.
- The software will be updated periodically in order to maintain it.

4.4 Usability Requirements

- There will be simple controls for the player on the keyboard so that player will not need to try to remember control buttons.
- The game rules will not be hard to understand, it will not have complex structure.
- The game provide help for the player to understand basics and rules of the game.

4.5 Constraints

- Bombplan is going to be implemented using Java programming language.
- Game graphics will be drawn by Paint.Net.
- Game language will be English.
- Space that is occupied by the Bombplan on the computer will not exceed 20 MBs.
- It will be an open source software licensed.

5. System Models

5.1 Scenarios

Scenario 1: Starting the Game

Player Ilkay wants to play a game and starts the game. Since she is pretty new to the game, she wants to get information and sees the help option in the main menu. After clicking the “Help”, she gets enough information on the game. She comes back to the main menu and chooses “New Game”. Default level 1 is loaded and she starts to play the game. Although, she learns the game from “Help”, she dies 3 times and loses all of her lives. Since she is the first player, her score is the top score. Thus, she sees the high scores menu and saves her score. After saving her score, Ilkay comes to the main menu and chooses quit.

Scenario 2: Change Settings

Player Osman enters a new game by pressing “New Game” from Main Menu. Osman plays the game without his earphones, and then he thinks that he can mute the sound since he already does not hear anything. In order to do that Osman stops the game by pressing “Esc” key. Then game pauses and a pause menu shows up. Osman change sound and the music on to off. Then he wants to continue to play game, so he presses “Return to Main Menu” button from the pause menu. After a while he is just bored and he wants to quit. From the pause menu he presses “Quit” and exit from the game.

Scenario 3: Load and Save Game

Player Cenk starts the game. After main menu appears, Cenk clicks “Load Game” button. He chooses last game from all saved games. Game initializes game board and settings

according to the last game which Cenk saved. After playing a while, Cenk stops the game and clicks to “Save Game” button in the pause menu. Game saves last settings successfully and returns to pause menu again. Then he presses “Quit” button and exits the game.

5.2 Use Case Model

Use Case Diagram

Diagram is shown in Figure 1:

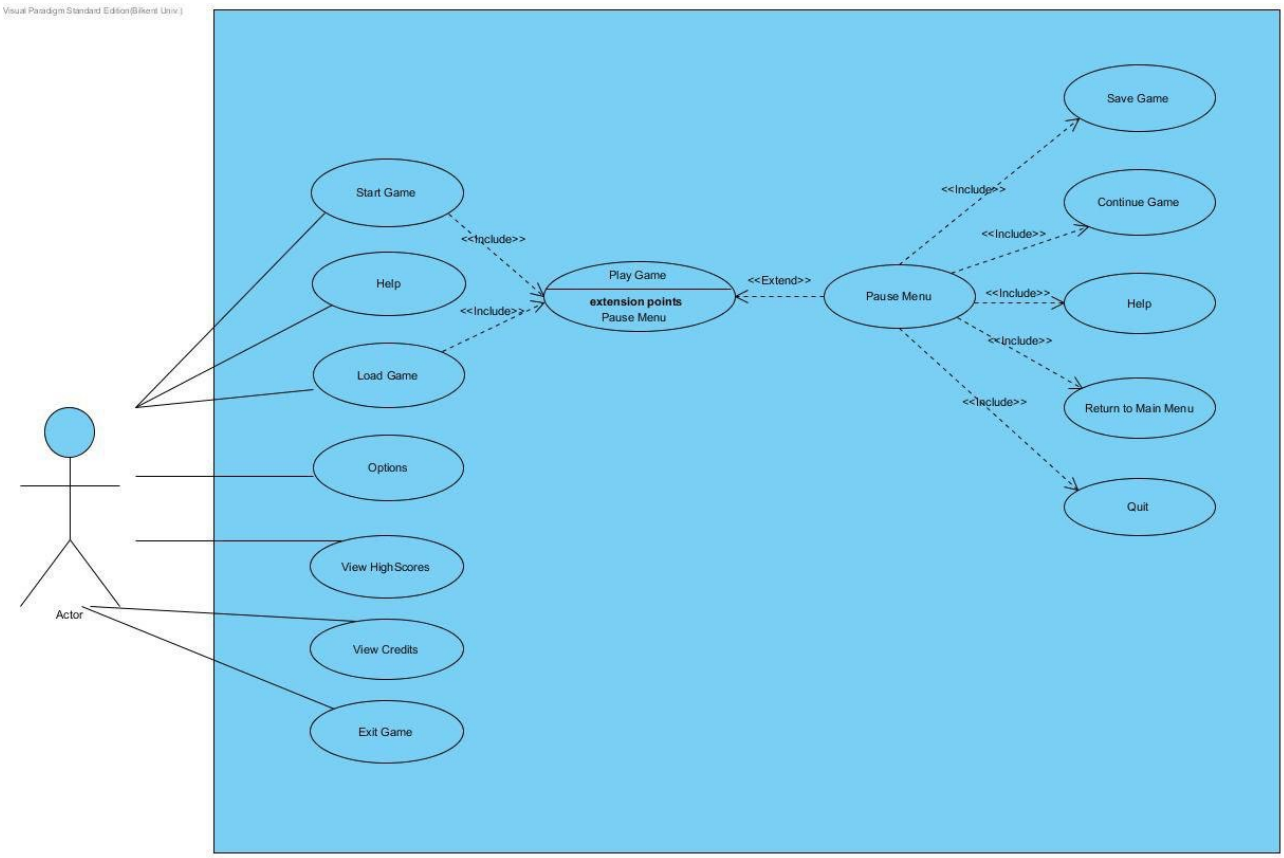

Figure 1: Use Case Diagram

Use Cases

Use Case 1

Name: Start Game

Participating Actors: Player

Flow of events:

- Player starts the application and encounters with the main menu.
- Player selects the “New Game” button.
- The system loads the default Level 1 map.
- The system distributes bonuses, monsters and the door randomly through the destroyable walls in the map.
- The system gives 3 lives to the user and locates the character in top left corner of the map. Also the time counter starts to countdown.

Entry Condition: User should start the application, and then press the new game button.

Exit Condition: The map is loaded with all monsters, bonuses, and the door successfully and the player is ready to play.

Exceptions: A possible loose of power may interrupt the application.

Use Case 2

Name: Load Game

Participating Actors: Player

Flow of events:

- Player chooses Load Game from the main menu.
- Saved games are listed on a table.
- The user chooses a game to load.
- Game is started by the system.

Entry Condition: User should choose the Load Game button from the main menu.

Exit Condition: User should choose the Back button on the screen that we list all of the saved games.

Exceptions: If there is no saved game, “No saved game” will be written in the middle of the screen and user should go back to the main screen.

Use Case 3

Name: View Credits

Participating Actors: Player

Flow of events:

- Credits frame is created by the system.
- Player sees names, e-mails of developers and current version of the game.

Entry condition: Player clicks “View Credits” button in the main menu.

Exit condition: Returning to the main menu by selecting “Return to main menu” button in the frame.

Exception: A possible loose of power may interrupt the application.

Use Case 4

Name: Quit

Participating Actors: Player

Flow of events:

- Player chooses Quit from the main menu.
- System asks the user "Are you sure you want to quit?"
- User chooses the "Yes" button.

- Application terminates.

Entry Condition: User should choose the Quit button from the main menu.

Exit Condition: User chooses "Yes" button or "No" button.

Exceptions: User choose "No" button.

Use Case 5

Name: Help

Participating Actors: Player

Flow of events:

- Player chooses Help from the main menu.
- System shows information about the main aim, abilities of the character, bonuses, how to use them, and how to move the character.

Entry Condition: User should choose the Help button from the main menu.

Exit Condition: User should choose the Back button on the screen.

Exceptions: A possible loos of power may interrupt the application.

Use Case 6

Name: Enter Options from the Main Menu

Participating Actors: Player

Flow of Events:

- The game is started and from the main menu user selects the “Options” button.
- Options panel is displayed on the screen. In this panel there are music and sound on / off options, and Return to Main Menu button.

- User mutes sound and the music by changing radio buttons from on to off.
- Settings will be saved by the system.
- User returns to the main menu by pressing “Return to Main Menu” button.

Entry Condition: The game is started and from the main menu user selects the “Options” button.

Exit Condition: User returns to the main menu by pressing “Return to Main Menu” button.

Exceptions: A possible loose of power may interrupt the application.

Use Case 7

Name: View High Scores

Participating Actors: Player

Main Flow of Events:

- The game is started and from the main menu user selects the “High Scores” button.
- High Scores panel is displayed on the screen. In this panel there are list of top ten scores that are previously played.
- User looks over the high score list.
- User returns to the Main Menu by clicking “Return to Main Menu” button.

Entry Condition: The game is started and from the main menu user selects the “High Scores” button.

Exit Condition: User returns to the Main Menu by clicking “Return to Main Menu” button.

Exceptions: There is no high score in the list, since the game has never been played or a possible loose of power may interrupt the application.