



CS 319 - Object-Oriented Software Engineering

Analysis Report

Virus Attack

Section 2 - Group F

Cholpon Mambetova

Doğan Can Eren

Melisa Onaran

Nergiz Ünal

Table of Contents

1. Introduction	2
2. Overview	3
3. Functional Requirements	4
3.1. Play Game	4
3.2. Change Settings	4
3.3. High Score	4
3.4. Pause Game	5
3.5. View Help	5
3.6. View Credit	5
4. Nonfunctional Requirements	5
4.1. High Performance	5
4.2. User-Friendly Interface	6
4.3. Extendability	6
5. System Models	7
5.1 Use Case Model	7

1. Introduction

“Virus Attack” is a game we have decided to develop as a project for CS 319 course. There is a virus that tries to get into human body, and player controls the virus’ motions. The main purpose of this game is to avoid colliding with the host body’s immune system by moving and dodging appropriately, and to be alive as much as you can in the host body. The game we took as an inspiration is the indie game called “Wave 2” made by the YouTube channel RealTutsGML.

We plan to add our own features and different difficulty levels, including different types of immune system cells and bonus features for the virus (player). The virus will try to escape from immune system’s attack which are controlled by the computer itself, if virus can survive for the given time it will accomplish the current level and move on to the next one which will be harder. Every level will have different type of immune system cells and virus attribute such as strength of virus against the immune system. If the virus gets hit by the immune system cells, the virus will lose its health points. The health of the virus is shown on top of the screen. If this bar goes empty, the virus will die and the game will be over.

The game will be a desktop application and controlled by a keyboard.

2. Overview

Virus Attack is a java game in which the player controls a virus in a host body which tries to survive by avoiding the immune system cells and gain points as the time goes on. Player should save their life by running away from the defender cells, if the immune system cells hit the virus, the virus loses health points until the collision ends. Also player is going to try to catch some bonus items. If the player can make the virus survive for the given amount of time, they will be granted the access to the next level of the game. The greatest goal is to accomplish all the levels and to survive as the virus travels through the host body, which will be very hard to be

successful.

2.1. Gameplay

The player can control the game via keyboard. The virus's motion will be controlled by up, down, right and left buttons. The player will move the virus by avoiding the immune system cells. Also the player should try to make the virus collide with and get the power-ups which will give some benefits to the virus.

2.2. Leveling

There will be 3 different levels that represent the human body parts. The virus adventure begins with the external part of human body and it continues throughout the internal part of human body to achieve its target. To accomplish levels, the player should keep the virus alive for the given time duration of every level, if the health point(HP) bar of the virus goes empty, the player fails. As the player progress through the levels, the damage taken and the number of the immune system cells will increase.

2.3. Immune System Cells

These cells will attack to the virus to kill it. There are several type of cells which have different attributes. When these cells hit the virus, it will lose from its health points. Every cell will have different attributes like speed and size. As the player progresses through the game, the introduced type of cells may change.

2.4. Power-ups

There will be some power-ups for the virus to make the game more enjoyable. The power-ups will occur on the game screen randomly, their time and place cannot be controlled by player. The player should move the virus through them to get them. These power-ups will give some

advantages to the virus.

The power-ups are:

Freeze the cells: It makes slow down the immune system cells' motion.

Kill the cells: The virus can kill a cell next to it

Invincible Virus: The virus can be invincible for a moment, and the cells cannot hit to it.

Life Lengthener: The HP bar of the virus will be extended, the virus will have more health.

3. Functional Requirements

3.1. Play Game

Virus Attack is a single player game in which the player acts as the virus. Virus comes up against the immune system cell/cells, a host body protected by its own defense mechanisms, according to the upcoming levels. There are some certain options of modes to start such as easy, medium and hard. And each mode has their own levels. As mentioned the user will be able to choose which mode to start (easy-medium-hard). The main goal of the game is to stay alive as long as possible. Therefore, the player is given a HP bar and he has up to several chances even if he is crashed with a immune system cell he will be losing life from the HP bar. If he survives for the duration time without having the HP bar empty then he will advance to the upcoming level. On the other hand, the player will have some chances to become invincible for a duration of time himself so that he cannot be killed, get a life bonus, freeze and slow the immune system cell/cells so that he can dodge easier(or to be able to kill the immune system cells directly as if he crash to the bonus items). In every level, the immune system cells' number of count will be increased and in some levels there will be different cells introduced. Therefore every upcoming level will be more difficult. It is up to the player to be in competition with the others by using the high score system or to play casually.

3.2. Change Settings

The system has its own default settings. Such as the color of the background, changeable color of the virus, HP bar and the sound at the back. The system allows the user to have an option to turn on/off the sound or to choose his own key combinations when playing. For instance, one can choose letters ‘W’, ‘A’, ‘S’ and ‘D’ instead of arrows. The creators on the other hand, always reserve the right to make changes in the game.

3.3. High Score

Player can choose one of the main modes as easy, medium and hard. The game provides different high score tables for each difficulty mode for the player to be able to see his highest score. So that he can see his progress and be in a competition with himself or the other players.

3.4. Pause Game

Player is always free to pause the game when it is wanted or necessary. The only important rule for pausing game is not to close the game totally in the computer. As there is no save option during a session; if it is closed, the current game will be lost and the player has to start a new game.

3.5. View Help

Player is always welcome to get help from the game itself. So he can click to the “View Help” button and read/learn about the following stuff:

- What is the main purpose of the game?
- What are the rules of the game?
- How can player control things?
- What are the HP bar and Bonus items?

The “View Help” button aims to help the player to fully understand the main purpose and

necessities of the game so that the game can be much more challenging and meaningful.

3.6. View Credit

Creators will be available through their contact information such as email addresses. Players can always be in communication with the creators. So that they can express their ideas because they are not allowed to make changes about the gameplay. Players can always suggest new ideas that they would enjoy more. So, creators can receive those suggestions and make some changes on the game accordingly. And at the end, if possible, creators will be interacted with players.

4. Nonfunctional Requirements

4.1. High Performance

The most important of nonfunctional requirements is high performance. Any software reaches the high performance, so does ours. It is essential that the game is played smoothly with minimum of bugs, delays or crashes. Therefore, it should request the minimum requirements of a standard personal computer, and be implemented by thoughtful design. Moreover, the video graphics should not be too “heavy”, yet sophisticated and engaging enough so that it is pleasant to an eye of a user. The time delays should be short in responses. The game should run fast and be played fluently.

4.2. User-Friendly Interface

As in most of the successful software programs, and specifically games, the GUI must be user-friendly. The interface should be easy to understand and interact with. Therefore, no unnecessary options, buttons and other text are allowed. The GUI must be highly organized and clear. Moreover, the design should be pleasant and interesting to watch and play. The sound shouldn't be too aggressive, yet intriguing and fulfilling the mood of adventure. In general, the

interface and the atmosphere of the game should be not only easy to perceive but also very enjoying.

4.3. Extendability

We have plans for the game to develop in future; therefore, the game should be extendable. We are planning to make it more complex by allowing a player to join a more realistic journey of the virus. Since the virus can infect the body through many ways, the user will be able to choose and follow those ways. For example, virus might get into a body through the cut on a finger, or through the nose when an infected person sneezes. The related background animation might take place by the change of a level and the body part the virus is going through. The other extension would be including multiplayer. A few users will be able to play the same game online helping each other to beat the body's defence mechanism, or even play against each other. The details are under development, yet the extendability is certainly decided to be the case.

5. System Models

5.1 Use Case Model

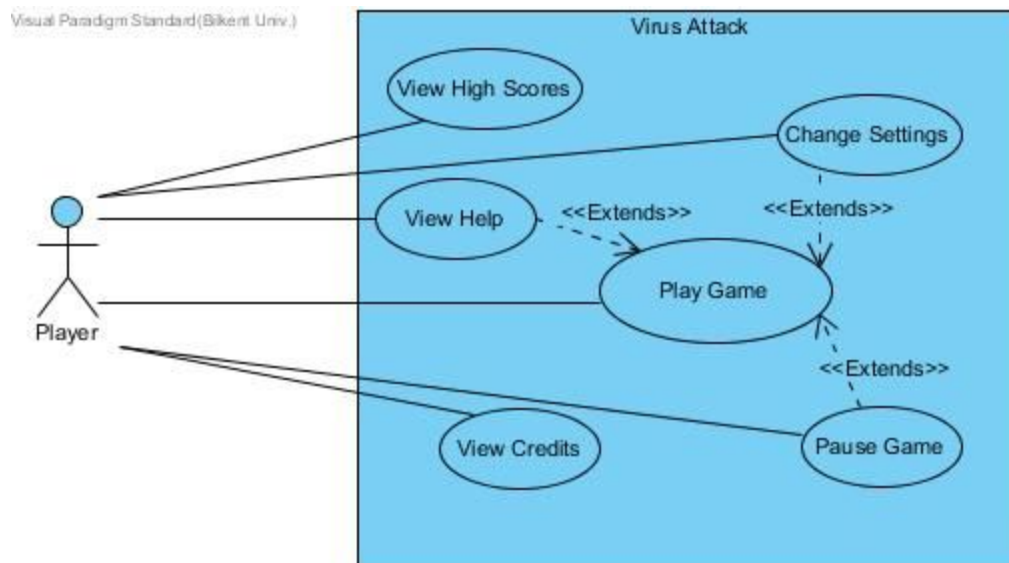


Figure 1 - Use case diagram of Virus Attack

Use Case Model Explanations

Use Case 1: View High Scores

Primary Actor: Player

Interests:

- Player wants to see the top ten high scores and who set them.
- The game shows the high score list with their scores and names.

Pre-conditions:

- Game must keep the top ten high scores recorded and ready to retrieve.
- Player must be in the Main Menu.

Post-conditions: -

Entry Condition: Player chooses the "High Scores" button in the Main Menu.

Exit Condition: Player chooses the "Return to the Main Menu" button in the High Scores menu.

Main Event Flow:

1. Player chooses the "High Scores" button in the Main Menu.
2. Game retrieves and shows the top ten scores in the High Scores menu.
3. Player chooses the "Return to the Main Menu" button in the High Scores menu.
4. Player returns to the Main Menu.

Alternative Event Flows: -

Use Case 2: Change Settings

Primary Actor: Player

Interests:

- Player wants to see change the settings(disable & enable the sound, change the controls).
- The game records the choices made by the player and adjust the settings accordingly.

Pre-conditions:

- For the first run, settings will be set to default.
- Player must be in the Main Menu.

Post-conditions: Game settings will be updated.

Entry Condition: Player chooses the "Change Settings" button in the Main Menu.

Exit Condition: Player chooses the "Return to the Main Menu" button in the Change Settings menu.

Main Event Flow:

1. Player chooses the "Change Settings" button in the Main Menu to set their choices of settings.
2. Game retrieves the Change Settings menu.
3. Player adjusts the settings and chooses the "Save Settings" button in the Change Settings menu.
4. Game updates the settings accordingly.
5. Player chooses the "Return to the Main Menu" button in the Change Settings menu.
6. Player returns to the Main Menu.

Alternative Event Flows:

- a. If the Player wants to return to the default settings:
 1. Player chooses the "Use Default Settings" button in the Change Settings menu.
 2. Game changes the settings to their default values.

Use Case 3: View Help

Primary Actor: Player

Interests:

- Player wants to see the description and the purpose of the game and learn how to play it.
- The game shows a text document which includes the description and the purpose of the game and tells the Player how to play it.

Pre-conditions: Player must be in the Main Menu.

Post-conditions: -

Entry Condition: Player chooses the "View Help" button in the Main Menu.

Exit Condition: Player chooses the "Return to the Main Menu" button in the View Help

Main Event Flow:

1. Player chooses the "View Help" button in the Main Menu.
2. The game displays the help menu.
3. Player chooses the "Return to the Main Menu" button in the help menu.
4. Player returns to the Main Menu.

Alternative Event Flows: -

Use Case 4: Play Game

Primary Actor: Player

Interests:

- Player wants to play the game.
- The game records the score of the current session.

Pre-conditions:

- Game uses the desired settings(default if not changed).
- Player must be in the Main Menu.
- Player must choose a difficulty.

Post-conditions: If the Player achieves a score that is enough to be in the high scores, the game will ask for the nickname and record the score accordingly.

Entry Condition: Player chooses the "Play Game" button in the Main Menu.

Exit Condition:

- Player chooses the "Return to the Main Menu" button in the pause menu.
- Player dies.
- Player completes the levels.

Main Event Flow:

1. Player chooses the "Play Game" button in the Main Menu.
2. Player chooses a difficulty.
3. The game starts according to the chosen difficulty.
4. Player progresses through the levels. - *This repeats until the player completes all of the levels or dies trying.*

5. Game checks if the score is eligible for recording and asks for the nickname to record it if it is eligible.
6. Game displays the High Scores list.
7. Player returns to the Main Menu. *Players can restart the game and repeat the steps if they want to.*

Alternative Event Flows:

- Player dies while progressing through the game (Jump to step 5).
- Player chooses to quit the game (Jump to step 5).

Use Case 5: Pause Game

Primary Actor: Player

Interests:

- Player wants to pause the game.

Pre-conditions: Player must be in-game.

Post-conditions: The game will be paused.

Entry Condition: Player chooses the "Pause Game" button in the game.

Exit Condition: Player either chooses to continue or leave the game.

Main Event Flow:

1. Player chooses the "Pause Game" button in the game.
2. The game pauses and retrieves the pause menu.
3. Player chooses to continue.
4. The game continues.

Alternative Event Flows:

- If the player wants to leave:
 1. Player chooses the "Leave Game" button in the pause menu.
 2. If the score is eligible for the high score table, game asks for the nickname and records the high score accordingly.
 3. Game displays the high score table.
 4. Player returns to the Main Menu.

Use Case 6: View Credits

Primary Actor: Player

Interests:

- Player wants to see who created Virus Attack.

Pre-conditions: Player must be in the Main Menu.

Post-conditions: -

Entry Condition: Player chooses the "View Credits" button in the Main Menu.

Exit Condition: Player chooses the "Return to the Main Menu" button in the View Credits menu.

Main Event Flow:

1. Player chooses the "View Credits" button in the Main Menu.
2. Game displays the credits.
3. Player chooses the "Return to the Main Menu" button in the View Credits menu.
4. Player returns to the Main Menu.

Alternative Event Flows: -