



**CS319 Object-Oriented Software Engineering
Analysis Report**

Run To Live

Group 2

Mehmet Nuri Yumuşak

Ege Yosunkaya

Merve Çapar

Selin Özdaş

Table of Contents

1.Introduction	2
2. Overview	2
2.1. Gameplay	3
2.2. Game Variables and Inputs	3
2.3. Levelling	4
2.4. Rules	5
2.5. Power-ups	7
3. Requirement Specification	8
3.1. Functional Requirements	8
3.1.1. Play Game	8
3.1.2. Settings	8
3.1.3. View Help	9
3.1.4. View Credits	9
3.2. Non-Functional Requirements	10
3.2.1. Playability and Replayability	10
3.2.2. Extendibility	10
3.2.3. Graphical Smoothness	10
3.2.4. User-Friendly Interface	10
3.2.5. “How to Play” Explanation	11
4. System Model	12
4.1. Use Case Model	13
Use Case Descriptions	14
4.1.1 Play Game	14
4.1.2. Get Help	15
4.1.4. Set Options	15
4.1.5. View Credits	16

1. Introduction

Run To Live is a kind of survival game. While the player aims to run away from virus and protect himself, virus transmits itself to kill the player. In the game there is a viral virus that infects lots of people in the world and kills them in some time. The player is going to try to run away from the virus until this virus stops its transmission through the map. There will be a map including many countries. In every turn, they will move between countries until the virus is destroyed. There will be some inputs in the very beginning of the game which will indicate the difficulty of it. Some of the inputs are the infection risk, the killing risk, the destroying possibility and the immune people. At the same time, there will be some rules like in every turn, players can just move 1 step, if they do not have any power-up to go further. In every country, we will show the number of dead people killed by virus and the number of doctors in that country to indicate the risks. According to these, the virus aims to kill the player, while he tries to run from the virus.

2. Game Overview

Run To Live is a strategy game that a man run from very dangerous virus to survive. Our character will try to run away from virus by travelling to other countries in some rules. In the meantime, virus also try to be separate itself and infects as much as people it can. Virus has AI and it travel across countries in some rules. As a time variable, there will be a turn system. In every turn both virus and character make move. After both of them moved, there is no time limitation to think next move. Game can be saved and continued after. There will be also doctors in the game. They will act like a normal person but they have privilege to develop a vaccine and save the people. When doctors vaccine all the infected people and prevent virus to separate game and also if our character still alive game finishes. Moreover, if our character gets infected and died game finishes. Otherwise, it continues. Furthermore, both virus and character will have some special powers.

2.1 Gameplay

Run To Live will be a 2D game for desktop. It is a single player game which you play against AI. To play this game you will need a mouse because every move in the game will be done by mouse click. Also our game will have some sounds. Therefore, you will need a speaker to increase game fun. Also, to enter some input before the game begins you need a keyboard. It will be a game easy to play. You just make moves with mouse and try to run away from virus by seeing the statistics about the virus on the countries.

Player will be able to see how many infected, dead people at the country also see how many doctors are there. All of these information can be seen on the countries. Moreover, on the right side of the game world statistics like how many dead people so far, how many infected or super people etc. can be seen there. On the top of the screen, there will be special powers.

2.2 Game variables and inputs

In this game some inputs should be entered. Because it is a survival game there is some variables about virus and the environment. According to this variables, difficulty can change and the time game takes can change.

- *Total people in the world (number)*
- *Infected percent at the beginning of the game (%)*
- *How many countries will be there (Because travelling can create problem world should be a square therefore given number should be square of some number)*
- *Super people percentage (% , super people means they never infected)*
- *Doctor number (number, doctors can vaccinate only healthy people in the same country with him and after vaccinate a people that people has immune for life)*
- *How many people a doctor can vaccinate in a turn (number)*
- *Air travelling percentage (% , this percentage means that when a person decides to travel with this percentage it choose air travel and can go to any country he wants. Otherwise he can just travel neighbor countries)*

2.3 Levelling

In this game there will be 3 level and 1 free mode. Because our game has too many variables we will serve 3 level of difficulty to player by default entering those variables. However, if the user wants to make his difficulty by himself, there will be a free mode for that.

Level-1-Easy:

- Map size: 8x8
- Total People in the world: 1600
- Infected percentage: %10
- Super people: %10
- Doctor number: 10
- Vaccinate number per day: 2 people
- Air travelling percentage: %25

Level-2-Medium:

- Map size: 6x6
- Total People in the world: 576
- Infected percentage: %15
- Super people: %7
- Doctor number: 7
- Vaccinate number per day: 2 people
- Air travelling percentage: %15

Level-3-Hard:

- Map size: 5x5
- Total People in the world: 875
- Infected percentage: %20
- Super people: %3
- Doctor number: 5
- Vaccinate number per day: 2 people
- Air travelling percentage: %7

In free mode every variable can be changed by the user and there can be created very customized world. However, of course there will be some restrictions like maximum country number etc.

Restrictions:

- Max. Map size: 10x10
- Min. People per country:5
- All the percentages should be min. %0 max. %100
- Doctor number cannot be more than total people
- Vaccinate people per day cannot be more than total people

2.4 Game Rules

1. Each country, except those at the edge of the world, has 4 neighbors at the north, south, west, east side. Countries are connected to their neighboring countries. A person can travel from a country to a neighboring country. A direct travel from a country to another is otherwise not possible.

2. At the beginning, there are P people in the world. As an initial condition, $X\%$ of them are infected with the virus (but they are not sick yet). Initially people are assigned to countries uniformly at random.

3. After 6 days of becoming infected, a person becomes sick and is therefore visibly infectious (i.e. other people can understand that she is unhealthy).

4. After 14 days of becoming infected, a person dies with a probability of 25%. Dead people do not move, but stay visibly infectious.

5. After 16 days of becoming infected, a person becomes immune and is no longer visibly infectious, but remains infectious.
6. After 18 days of becoming infected, a person turns healthy. He is now in the same state as he was before his infection, which means that he can get infected again.
7. Each step of the game corresponds to one game day.
8. At the beginning of the game, each person decides when to move within the next 5 days. More precisely, she picks the moving day uniformly at random among the next 5 days. After moving, she repeats the same process. For example, if on day 0 the person decides to move after 3 days, she will move from her current country on day 3. After that move, she will make another decision: for example, she may decide to move after 5 days this time, so her next move will be on day 8, etc.
9. On the day of each move, a person will pick one of the countries available to her with equal probability and will move to it. The set of available countries is defined as the neighboring countries that contain no visibly infectious people. A person avoids countries with sick or dead (visibly infectious) people. This means that if a person is surrounded by visibly infectious people, she does not change position; however, she might change position the next time she tries to move (for example, if a visibly infectious person moved out of one of the neighboring countries or became immune).
10. When a person moves into a country with an infectious person she might get infected according to the transmissibility rate of 40%, unless the person is already infected or immune. A person cannot get infected between moves (this is slightly unrealistic, but will simplify your implementation).

2.5 Power-ups

In this game, to make this game more challenging and prevent it from monotony we will add some extra power to both our character and virus. Because virus and character are against each other, for every extra power one of them have, other one have against power. It will make our game more balanced and force player to be more strategic. Because this powers are limited, player should use them wisely. Moreover, some of the powers will be gained after some achievements.

Move 2 country in one turn:

This power is for both virus and player. When this power is used, player or virus can move two countries at the same turn. This is an advantage for player to run away further and also advantage for virus to separate itself faster. They will be able to use this power just 3 times.

Be healthy:

This power is for player. When he is infected and not to be cured he can use this power just 1 times. However, if our player is infected and then he gets rid of from it then he will gain 1 more this power.

Check himself:

This power is for player and it is used to see if player is infected or not. Because infected people cannot be understood before 6 days, it is a disadvantage for player to make moves. By using this power, he will be able to see whether he is infected or not. This power will be used just 2 times.

Kill a doctor:

This power is for virus. It is used for killing a doctor. By doing this move, virus will gain advantage because doctors can cure the sickness. I will be used just 2 times. This is a risky power because virus cannot see if there is a doctor or not in a country, it should use this power wisely.

Increase probability:

This power is for virus. If our player is infected one time this power will be enable to use and increase the probability to infect a people.

Air travelling:

This power is for player. Player can move any country in the world. This power will be used just for 2 times.

3.Requirement Specification

3.1 Functional Requirements

3.1.1. Play Game

Run To Live is a type of strategic game that programmed by C#. The aim of the game is staying alive when you are escaping from the virus. When the game starts, some number of countries that can be changed according to level of the game, will be appeared first. In every country, the information about how healthy country is will be seen on the map. The mean of how healthy the country is, player can see percentage of infected people, super people, number of doctors, vaccine number per day, and percentage of air travelling on the screen. To win the game, player needs to stay healthy and virus should be demolished from the world.

To make the game more attractive, there will be some incentives like giving a chance to player in favor. It can be various powers to enhance player's ability on the game.

3.1.2. Settings

To be able to appeal several types of players Run To Live provides users to several setting options. Before starting the game, player can change some attributes of the game according to his or her preferences. Therefore, player enjoy the game more and more. The list of setting is like:

- Character selection Man / Woman
- Level of game Easy / Middle / Hard / Free
- Sound On / Off

The character of the game is important for the perspective of player in lots of times because player go into play by reflecting himself or herself into character that play with. Therefore, game provides players to option of selecting own character. Player can also

select level of game in terms of easy middle and hard. In addition to these settings, player can select on or off for the sound of the game.

3.1.3 View Help

To help understandability of the game, there will be option that explains game rule and problems about to play, help. In the help section, information takes place listed below:

- Aim of the game
- Rules of move of player
- Rules of spreading and demolishing virus
- Player controls
- Explanation of how to use power ups.

To get more fun from the Run To Live, user should know how to play and understand the essence of game. For fulfilling these requirements, game provides user to help window.

3.1.4. View Credits

Player can see the followings under view credits tab:

- Situation of player in terms of infected or healthy
- Number of infected countries and people in the world
- Types of power ups that player owns currently.

Stated information above will be shown under this tab and it provides users to play more carefully and consciously. Since Run To Live is a strategy game and every player set their own strategy according to current situation and equipment like remaining power ups.

3.2 Non Functional Requirements

3.2.1 Playability and Replayability

Since we are using a simulation for the virus, which the player is trying to avoid get infected by through the game, the simulation should be neither hard as the player cannot play nor too simple that the player can win effortlessly. In that sense, we planned to design a logical and smooth rule set for the virus that will provide necessity for strategic thinking and enjoyable gameplay for the player.

On the replayability hand, our intention as a group, is find the right values for the simulation, so that every new game would be different from the previous one. Moreover, when the game restarts simulation would maintain its difficulty and forces player to recreate his strategy.

3.2.2 Extendibility

For a computer game, future improvements and changes in gameplay are essential. As a group we intended to design the simulation algorithm for the virus easy to adjust according to feedback from the player audience. Moreover, some improvements can be applied from the user side to provide more enjoyable gameplay such as adjusting the player's rule for the game.

3.2.3 Graphical Smoothness

Graphical beauty is not our main concern since we are developing a game which has a strategic feature, however graphics is an important part for a game. Thus we decided to keep graphics simple and well designed to contribute the gameplay and at the same time provide a smooth gameplay experience, which includes fast response time and minimal waiting time in between the turns.

3.2.4 User-Friendly Interface

The UI of the game is an important concern for the player. Because of that, developing a simple, easy to understand and nice-looking interface is important for the developing process. Since our game involves many data to inform player about the current state of the game, UI should provide aforementioned data in an organized way, such that user can understand and keep track of data with ease.

3.2.5 “How To Play” Explanation

The game includes a set of rules both for the player and simulation. It is essential for the user to understand those rules and determine his strategy accordingly throughout the game. To achieve this, there will be an easy to understand guide to teach the user those rules.

4. System Models

4.1 Use Case Model

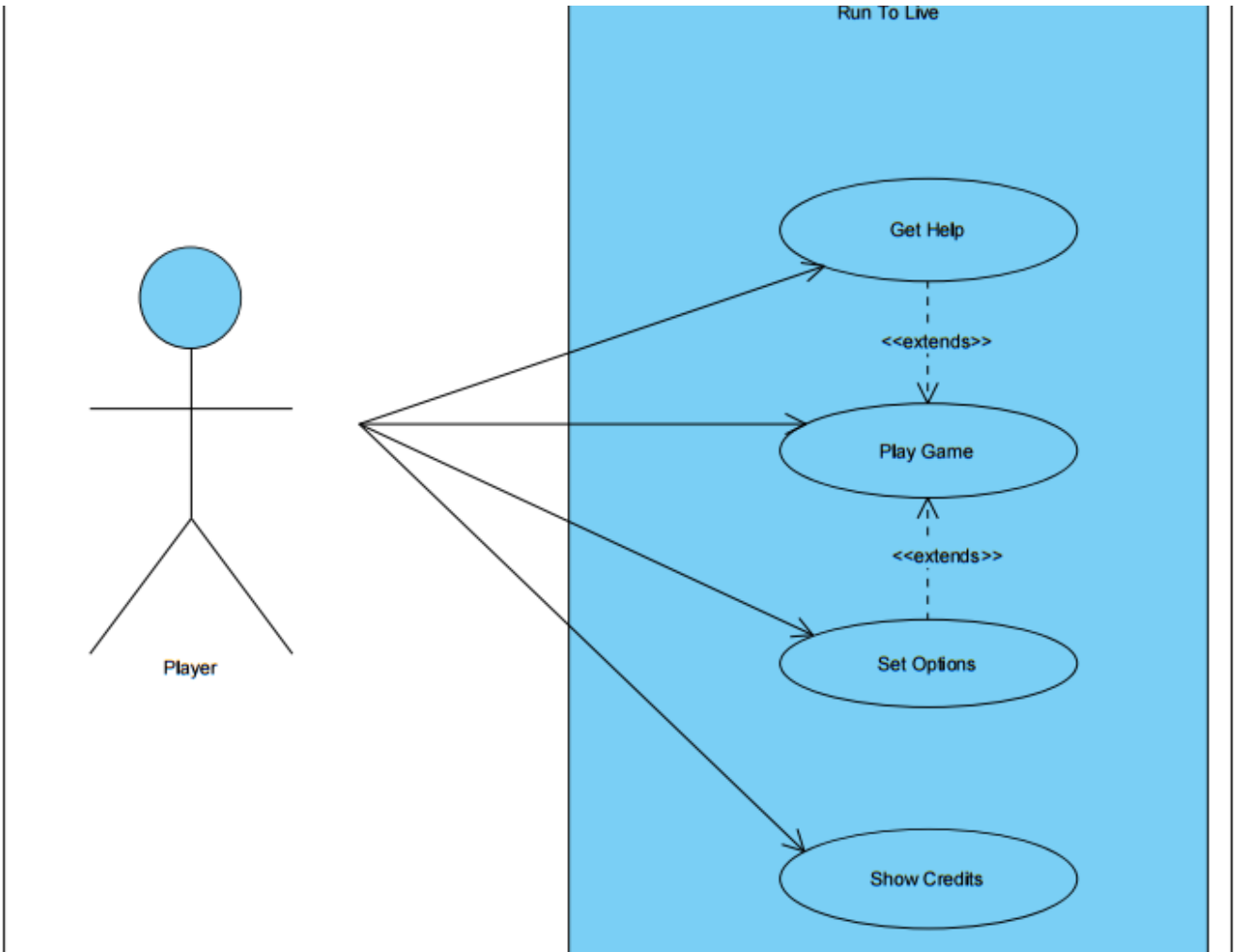


Figure 1 - Use Case Diagram

4.1 Use Case Descriptions

4.1.1 Use case #1

Use case name: Play Game

Participating actors: Player

Stakeholders and Interests:

- Player aims to survive by running away from the virus and healing himself.
- System keeps the data of the Player and the other statistics

Pre-condition: Game starts with default settings unless it is changed.

Post-condition: None

Entry Condition: Player opens the game and starts to play it.

Exit Condition:

- Player dies, OR
- There is no infected person which may transmit the virus to the player, OR
- Player exits the game via menu.

Main Flow of Events:

1. Player starts the game.
2. The system constructs the game according to the default settings.
3. Player succeeds to survive and the virus cannot transmit itself anymore.
4. System shows a dialogue of the player statistics.
5. System returns to the main menu.

* This case may be represented due to the choice of the Player.

Alternative Flow of Events

3A. Player tries to run away and to heal himself:

3A.1 Player moves according to his own pleasure from one country to another.

3A.2 Virus spreads through the map.

3A.3 Player gets infected and killed by virus.

3B. Player is awarded with power ups:

3B.1 Player moves according to his own pleasure from one country to another.

3B.2 Virus spreads through the map.

3B.3 Players wins power ups by qualifying the stated conditions.

3B.4 System allows user to play with the features of the power ups.

3B.5 Due to the play game, this steps might be followed by 3B1-3B5,

OR 3A

OR the main flow step.

A. Player requests to exit the game during gameplay:

A.1 Game resets itself with default settings.

4.1.2 Use Case #2:

Use case name: Get Help

Participating Actors: Player

Stakeholders and Interests:

- Player wants to learn instructions, purpose and rules of the game.
- System shows those by supporting the texts with images.

Pre-condition: Player should be in Main Menu.

Post-condition: None

Entry Condition: Player clicks on the button called “Get Help” from the Main Menu.

Exit Condition: Player clicks on the button called “Return to Menu” from Get Help screen.

Main Flow of Events:

1. Player clicks on the “Get Help” button from Main Menu.
2. System shows the information file including instructions and purpose of the game.

Alternative Flow of Events:

A. Player wants to return to the Main Menu:

A.1 Player presses on the “Return to Menu” button.

A.2 System returns the player to the Main Menu.

4.1.3 Use Case #3:

Use case name: Set Options

Participating Actors: Player

Stakeholders and Interests:

- Player wants to change the difficulty level of the game, game character and set the sound on/off.
- System applies the changes due to the preference of the player.

Pre-conditions: Player should be in Main Menu.

Post-conditions: None

Entry conditions: Player clicks on the “Settings” button from the Main Menu.

Exit conditions: Player clicks on the button “Apply” or “Cancel” button.

Main Flow of Events:

1. Player clicks on the “Settings” button from the Main Menu.
2. System displays the available set types to the user.
3. Player changes all of the settings due to his preference.
4. Player clicks on “Apply” button.
5. System returns player to the Main Menu.

Alternative Flow of Events:

3A. Player chooses to change some of the settings:

3A.1 System keeps the default settings for the unchanged set types.

3B. Player chooses not to change the settings.

3B.1 System keeps the default settings if there is no previously applied setting change. If there is any previously applied change, system keeps that settings.

4A. Player changes the settings but clicks on the “Cancel” button:

4A.1 System brings back the previously applied settings.

4.1.4 Use Case #4:

Use case name: Show Credits

Participating Actors: Player

Stakeholders and Interests:

- Player wants to get information about the developers, contributors etc. of the game.
- System shows the list of those subjects.

Pre-condition: Player should be in Main Menu.

Post-condition: None

Entry conditions: Player clicks on the “Credits” button from the Main Menu.

Exit conditions: Player clicks on the “Cancel” button from the Credits screen.

Main Flow of Events:

1. Player clicks on the “Credits” button from Main Menu.
2. System shows the text file showing the related information.
3. Player clicks on “Cancel” button to return to Main Menu.
4. System returns player to the Main Menu

Alternative Flow of the Events: None