“Snake Game RPG” Backlog

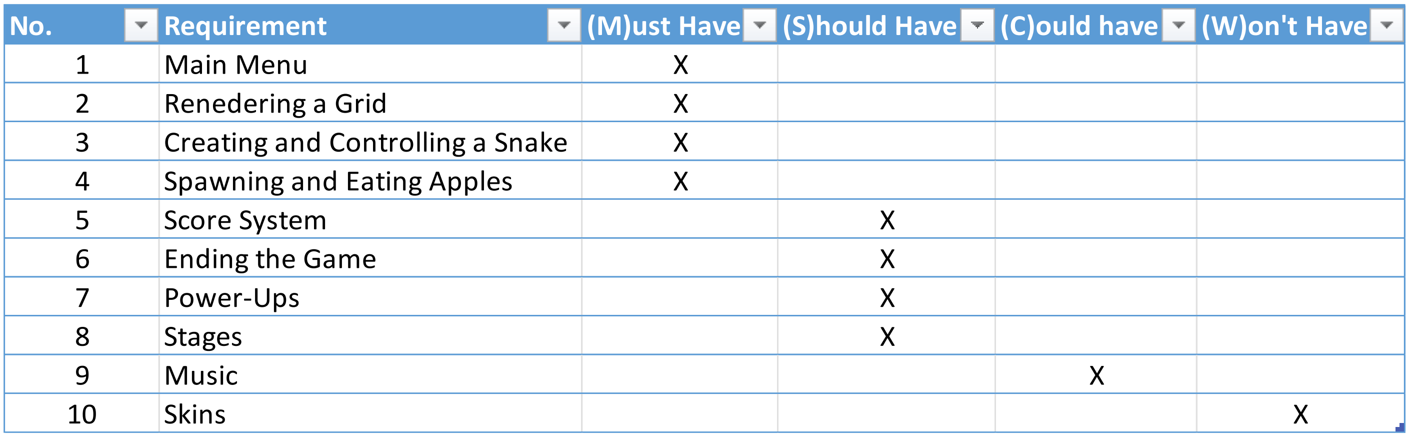
Student 1: Horia-George Dună

Student 1 ID: 1949284

Student 2: Radu-Cristian Sarău

Student 2 ID: 1939149

Features Included in the Game:



No.	Requirement	(M)ust Have	(S)hould Have	(C)ould have	(W)on't Have
1	Main Menu	X			
2	Renedering a Grid	X			
3	Creating and Controlling a Snake	X			
4	Spawning and Eating Apples	X			
5	Score System		X		
6	Ending the Game		X		
7	Power-Ups		X		
8	Stages		X		
9	Music			X	
10	Skins				X

1) Main Menu

When the user starts the program, they are greeted with the name of the game, positioned on the northern side of the window, the “PLAY” button, which begins the game; the “OPTIONS” button, which shows a panel from where the user can turn the music off, go back to the Title Screen, or exit the game; and the “EXIT” button, which aborts the program.

Category: User Experience.

Learning goals: Rendering custom-made buttons using ImageIcon components, creating buttons which lead to different JPanels.

2) Rendering a Grid

Start the application, and a 700 * 700px grid of 29 * 28 (LENGTH * HEIGTH) squares appears in the center of the screen.

Category: Game design

Learning goals: Rendering a frame using a JPanel which is focusable, not resizable, has a background picture, a preferred size, and a KeyListener.

3) Creating and Controlling a Snake

Upon starting the application, a snake is rendered in the top-left corner of the window. It immediately starts moving, or as its movement is called in the game files, slithering. The snake can travel in 4 directions: to the right, to the left, up and down. If it collides with itself, or with one of the borders, the player loses.

Category: Game design

Learning goals: Rendering a movable snake using a Graphics component inside two arrays, checking collision conditions inside these two arrays, using the switch instruction for the snake's direction, checking input from the user, working with KeyEvents, working with the Timer data type.

4) Spawning and Eating Apples

When the application starts, an apple is spawned at a random location in the grid. If the snake eats it, a new one spawns. The snake eats the apple when its head's coordinates are the same as the apple's.

Category: Game design

Learning goals: Rendering an apple at a random location using the Random data type and a Graphics component, increasing the snake's length by incrementing a variable which stores it, and increasing the score using the same tactic.

5) Score System

Once the game starts running, the player begins with a score of 0. Their score increases based on the number of fruits they eat (1 fruit = 1 point). The score is rendered in the middle of the northern side of the window.

Category: Game design

Learning goals: working with a paintComponent by drawing Strings inside a JPanel, dealing with FontMetrics, incrementing a variable which stores the player's score after every eaten apple.

6) Ending the Game

If the player collides with itself, or with the border of the grid, the game stops. The grid disappears, and the window displays the player's score in the position it was in before, and the message "GAME OVER" in its center.

Category: Game design

Learning goals: working with a paintComponent by drawing Strings inside a JPanel, dealing with FontMetrics.

7) Power-ups

The game spawns different-colored fruit at a random time and position. Each fruit has a different effect on the game. For example, one of these fruits makes the snake faster.
The game spawns different-colored fruit at a random time and position. Each fruit has a different effect on the game. For example, one of these fruits makes the snake faster. A list of these fruits can be found below:

1. Blueberry (dark blue): increases the snake's speed.
2. Orange (orange): increases the score and the snake's size by +2.
3. Rotten apple (brown): decreases the score by -1, the snake's length by -1, and its speed.
4. Lemon (yellow): the snake gains the ability to slither over itself.

Category: Game design
Learning goals: Rendering fruit sprites at a random location using the Random data type and a Graphics component, altering the snake's abilities by changing variables.

8) Stages

The player starts at stage 1. Each stage will have its own background and theme. Each time the player collects a nonzero multiple of 25 apples, they advance to the next stage, as follows:

1. Stage 1 —> Stage 2: 25 apples
2. Stage 2 —> Stage 3: 50 apples
3. Stage 3 —> Stage 4: 75 apples
4. Stage 4 —> Stage 5: 100 apples

When the player's score reaches 125, they win the game, and the game is over.

Category: Game Design

Learning goals: Provide the user with a clear way of measuring their performance in the game, rendering different backgrounds using ImageIcon components, and implementing different sound files, which is beyond this course's material.

9) Sounds

Music will play all the time while the application is running. The Main Menu and all the different stages will have different theme-songs.
Category: Game design, User Experience
Learning goals: Implementing sounds in a Java application, which is beyond this course's material.

10) Updates

10/10/2023 - Base game completed. The game draws a grid of squares, a snake that moves from square to square and apples that the snake has to eat to gain length and points. The game's UI features the point counter at the top of the window and the stage counter at the bottom.

11/10/2023 - ”Main Menu” screen and “Pause” screen implemented. The game starts with the “Main Menu” screen, where you can start the game or close the application. The player can pause the game with the “ESC” key.

12/10/2023 - The music for the first stage is completed.

13/10/2023 - The music for the second stage is completed.

14/10/2023 - The music for the fifth stage is completed.

15/10/2023 - The music is implemented with a minor bug: the music file starts playing at every game tick if the player reaches the point condition to change the stage.

17/10/2023 - Added an options menu, which includes the option to mute the game, the option to go back to the title screen, and the option to quit the game.

19/10/2023 - Implemented rotten apples, lemons and oranges. The lemon is drawn in the corner for a short period before disappearing.

20/10/2023 - Implemented the blueberry, with the same bugs as the lemon.

24/10/2023 - Fixed the blueberry spawning in the corner for a short while. Now, it behaves as expected. Music for stage 3 was completed and implemented. All backgrounds are constructed and implemented.

26/10/2023 - Completed and implemented music for the fourth stage. Completed and implemented sprites for all fruits and the game icon.

28/10/2023 - Fixed the bug where lemons and other fruits spawn only in the corner. Now all fruits behave as expected. The game is considered complete.

11) Game Design Elements

1. Moveable Snake: the player can move the snake using the arrow keys on the keyboard.
2. Randomly Generated Fruits: the game randomly generates fruits which can be eaten by the player to increase/decrease their score.
3. Score System: each fruit increases/decreases the player’s score by a certain number.
4. Stage System: when the score is a multiple of 25, the player advances to the next stage.
5. Win/Lose System: when the player reaches the score of 125, they win, but if they collide with the border of the grid, or with themself before eating a lemon, they lose.
6. Power-Up System: each fruit, besides the apple, has a special effect on the snake, as described above.

12) User Experience Elements

1. Simple-to-understand concept: Anyone can start the game and enjoy a tried and tested formula which progressively gets harder.
2. Fun additions to a classic formula: The power-ups provide a unique spin on the “Snake” formula, keeping things fresh and improving the game experience.
3. Title screen with Main Menu: upon opening the game, the player sees a title screen, which includes a main menu. This title screen gives the player three options, namely “PLAY”, “OPTIONS”, and “EXIT”. Easy-to-comprehend options lead to a straightforward experience that leaves no room for error.
4. Options Menu: The options menu gives the player the ability to mute the game, go back to the title screen, or exit the game.
5. Pause Button: while playing the game, the player can press the “ESC” button on the keyboard to pause/resume the game.