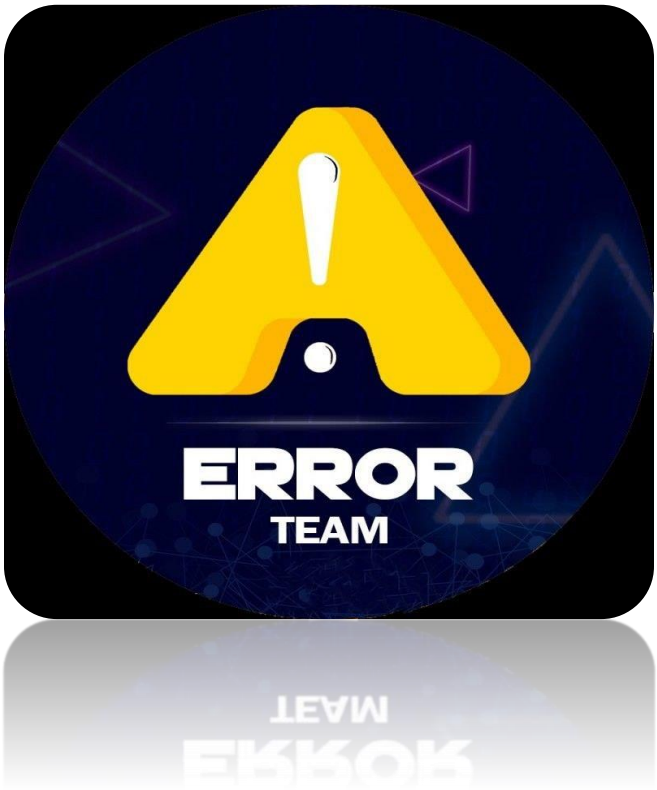
E R R O R



ERROR TEAM

IN

Snake Project

Team with ID:

1. Mariam George Eskander. (18)
2. Mariam Nader Farouk. (18)
3. Menna Gamal Mohamed. (18)
4. Manal Ali Abdallah. (18)
5. Mai Mohamed Naguib. (18)
6. Noura Hamza Mohamed. (19)
7. Naira Ebrahim ELDsoky. (19)
8. Heba Mohamed Hasan. (19)
9. Heba Ragheb Abdtawab. (19)

Main idea of project:

👉 The purpose of the game:

The main idea of the game Snake is to control a snake that moves around the game board, eats food, avoids walls and eats itself.

When the snake eats more food, it grows longer, and increases in each eater the score and the maximum score that the snake reaches, amount of food that was eaten.

Which makes it more difficult to navigate without hitting an obstacle.

The game continues until at which point the game ends. The goal is to achieve the highest possible score by eating as much food as possible.

👉 Game conditions to win:

1. Quick reflexes.

2. strategic thinking to guide the snake across the game board while avoiding obstacles and collecting food.

3.Get a high score.

👉 Game Rule:

In the game there are settings to modify

1. The Music
2. The Background Color.
3. How to Play.

👉 Game's Controls:

Pause/ Start : Space.

Restart : Enter.

Move Up : Arrow Up.

Move Down : Arrow Down.

Move left : Arrow Left.

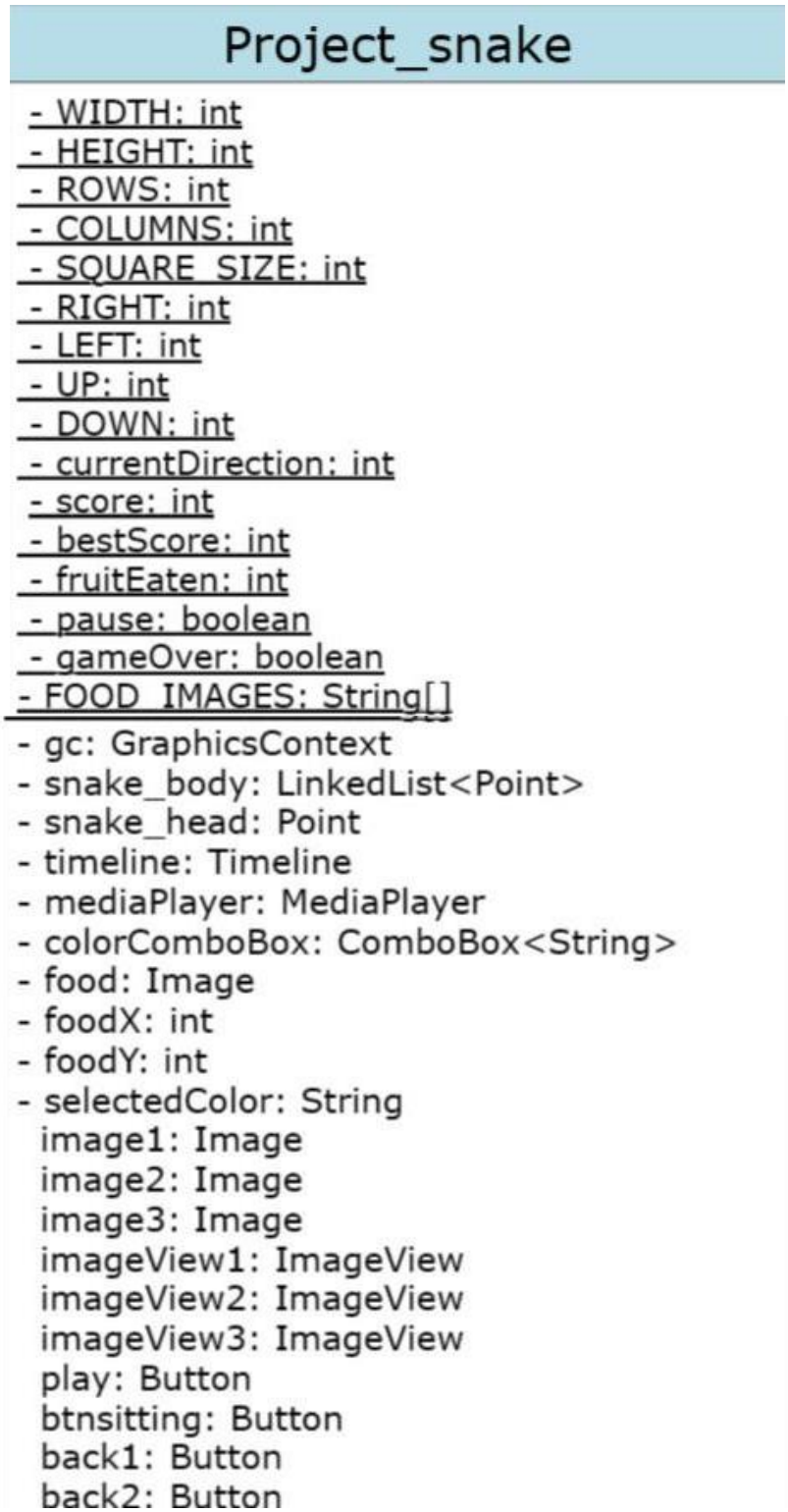
Move Right : Arrow Right.

To Return Back : Esc.

👉 Screen shoot of all import:

```
package projeect_snake;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.Background;
import javafx.scene.layout.*;
import javafx.stage.Stage;
import javafx.scene.canvas.Canvas;
import javafx.scene.canvas.GraphicsContext;
import javafx.scene.input.KeyCode;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import java.util.LinkedList;
import java.awt.Point;
import java.nio.file.Paths;
import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.geometry.Insets;
import javafx.scene.effect.DropShadow;
import javafx.scene.input.MouseEvent;
import javafx.scene.media.Media;
import javafx.scene.media.MediaPlayer;
import javafx.scene.text.FontWeight;
import javafx.util.Duration;
```

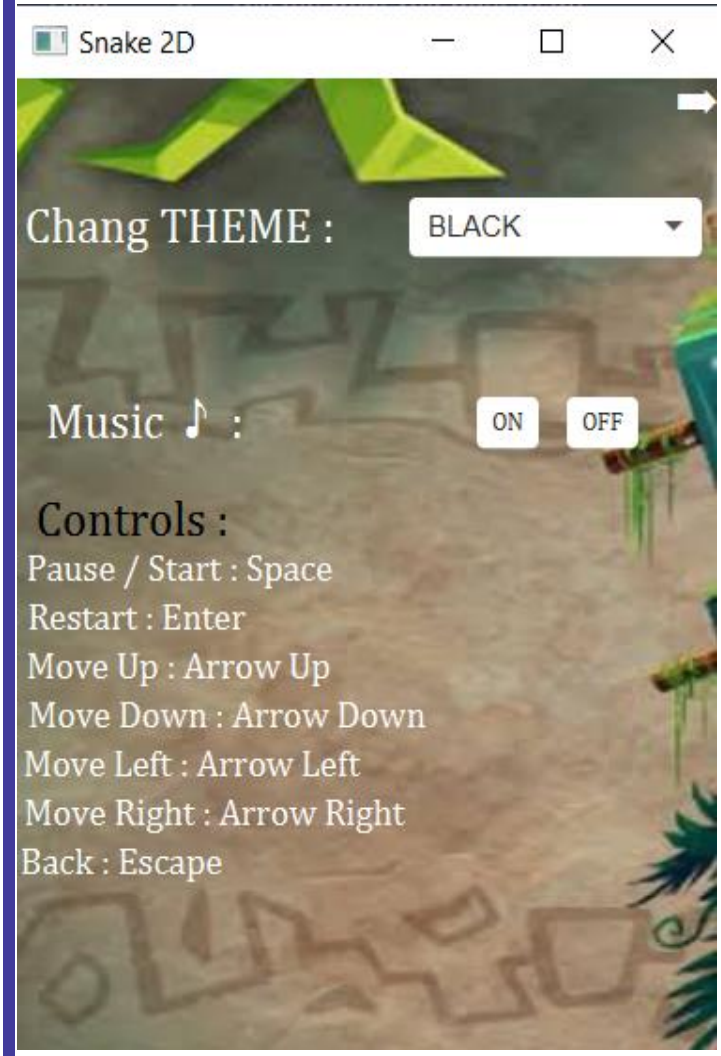
👉 UML 👈


Project_snake
- WIDTH: int
- HEIGHT: int
- ROWS: int
- COLUMNS: int
- SQUARE_SIZE: int
- RIGHT: int
- LEFT: int
- UP: int
- DOWN: int
- currentDirection: int
- score: int
- bestScore: int
- fruitEaten: int
- pause: boolean
- gameOver: boolean
- FOOD_IMAGES: String[]
- gc: GraphicsContext
- snake_body: LinkedList<Point>
- snake_head: Point
- timeline: Timeline
- mediaPlayer: MediaPlayer
- colorComboBox: ComboBox<String>
- food: Image
- foodX: int
- foodY: int
- selectedColor: String
image1: Image
image2: Image
image3: Image
imageView1: ImageView
imageView2: ImageView
imageView3: ImageView
play: Button
btnsitting: Button
back1: Button
back2: Button

on: Button
off: Button
exit: Button
about: Button
t1: Label
t2: Label
t3: Label
t4: Label
t5: Label
t6: Label
t7: Label
t8: Label
t9: Label
t10: Label

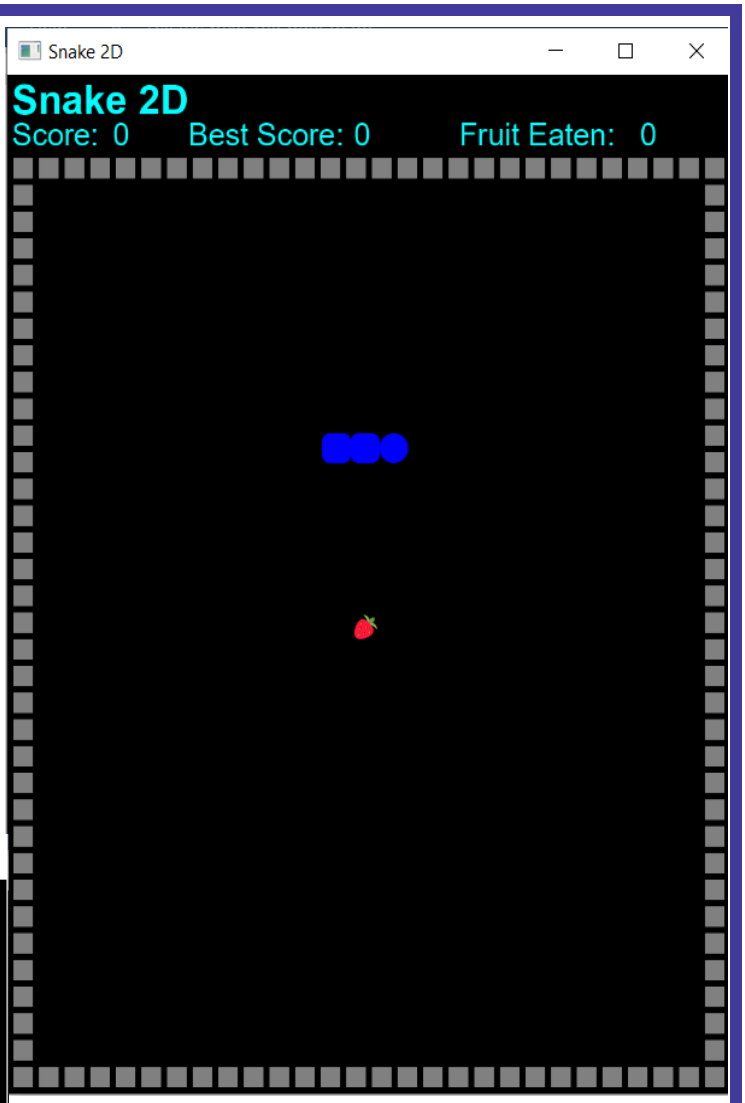
+start(primaryStage: Stage) : void
+ moving_snake(): void
+ buliding_snake(): void
+ drawing_snake(): void
+ making_food(): void
+ drawing_food(): void
+ eating_food(): void
+ background(): void
+ score(): void
+ bestScore(): void
+ fruitEaten(): void
+ gameOver(): void
+ pauseGame(): void
+ resumeGame(): void
+ restartGame(): void
+ music(): void
+ about(): cene

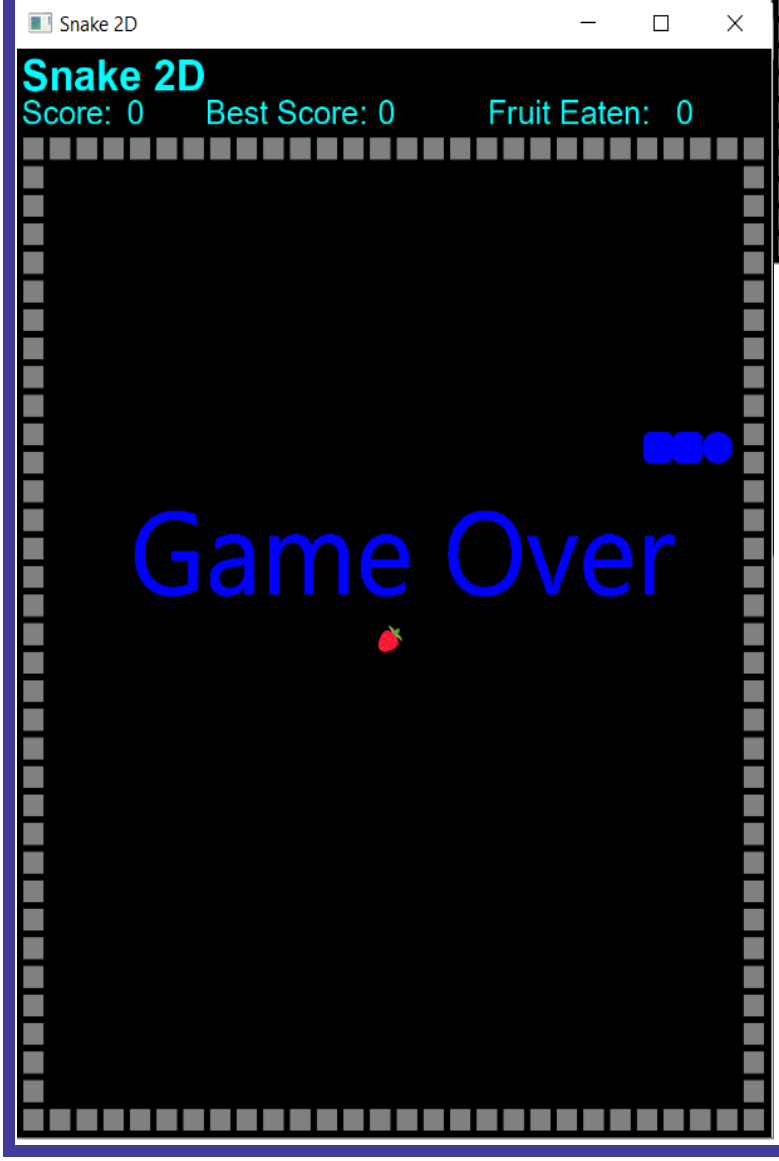
Snake Cover Game:



Siting ⚙

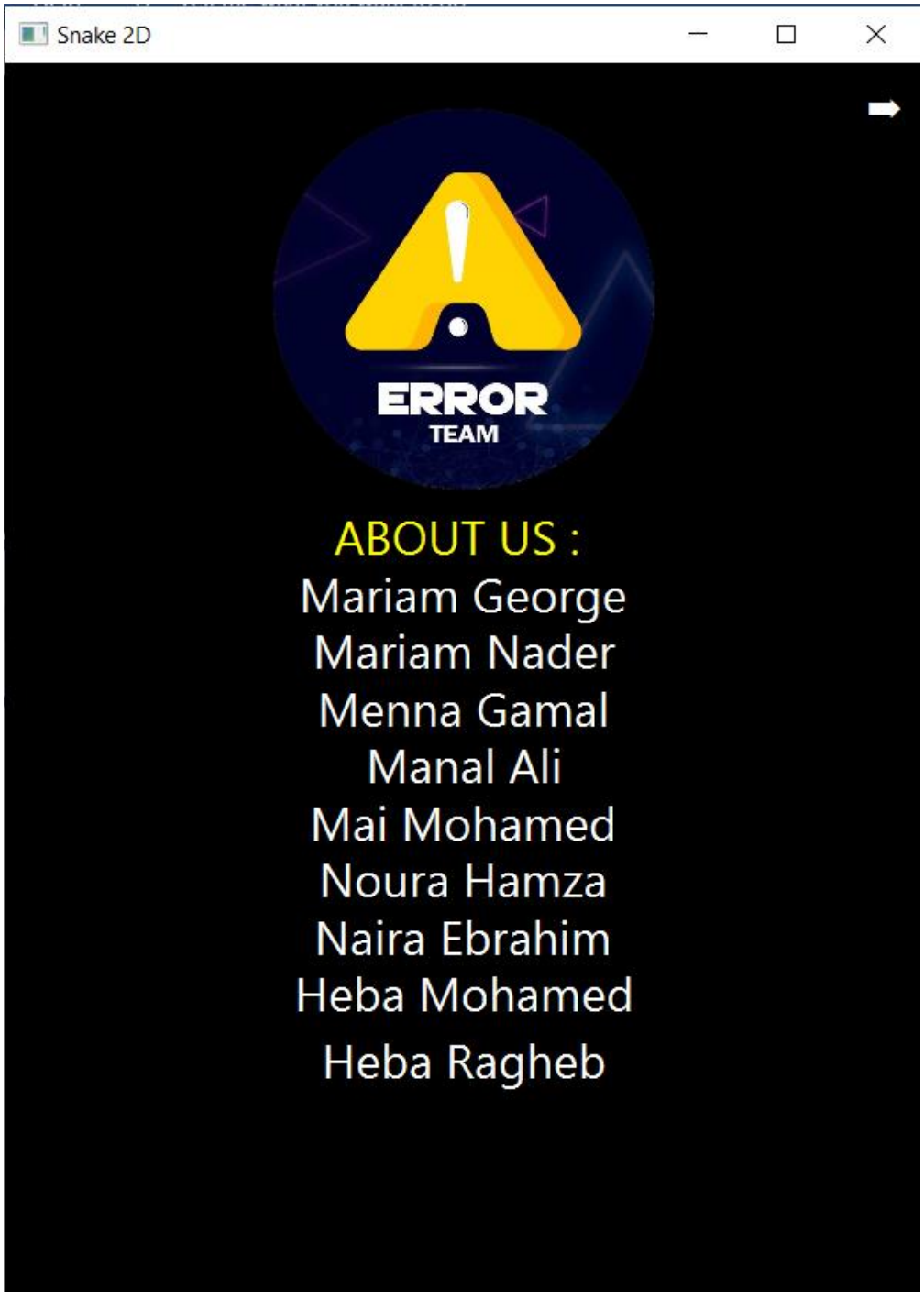
Playing:





Game Over

ABOUT US


Snake 2D
ERROR
TEAM
ABOUT US :
Mariam George
Mariam Nader
Menna Gamal
Manal Ali
Mai Mohamed
Noura Hamza
Naira Ebrahim
Heba Mohamed
Heba Ragheb

👉 METHOD 👈

1. **public void moving_snake( ){ };**

this method to move snake in scene2 when we change diracton by keyboard.

2. **public void buliding_snake( ){ };**

this method to build the first body of the snake that contain head and 2 square in the body = 3.

3. **public void drawing_snake( ){ };**

the method that drawing snake in scene2 and control the size and color of snake.

4. **public void making_food( ){ };**

method that generate food randomly in scene exept on the snake or out the frame.

5. **public void drawing_food( ){ };**

method that generate food in size of square.

6. public void eating_food( ){ };

method that calculate score-bestscore-fruiteaten when snake eat the food when the point of heat exist in food.

7. public void background( ){ };

method that control the colors of background in scene2 and control change the colors by comboBox.

8. public void score( ){ };

method to calculate the score didn't store.

9. public void bestScore( ){ };

method to calculate the best score until close the main stage store.

10. public void fruitEaten( ){ };

method to calculat the numbers of fruit from one point.

11. public void gameOver( ) { };

method that appear "gameover" when the sanke crush to frame or his body.

12. public void pauseGame( ){ };

this method to pause the game when we enter the space if the game on.

13. public void resumeGame( ){ };

this method to resume the game when we enter the space if the game off.

14. public void restartGame( ){ };

thes method to restart the game and clear the number of score and fruitEaten without clear the number of best score.

15. public void music( ){ };

method to play music when we start the game.

16. public Scene about( ){ };

this method that control the appearance of scene4 when we click "about" button it contain 10 labels and the logo's photo.

👉The problems Which we solved:

1. Multiple stages so for game, the settings ,the external image and about us.

Solution of (1).
That these were made on the same stage using the number of 4 scene on one stage.

2. **The problem of drawing on a pane using appropriate dimensions such as the border, the appearance of food, etc...**

Solution of (2).
Use of GraphicsContext which helps drawing easily with proper dimensions.
(import javafx.scene.canvas.Canvas)
(import javafx.scene.canvas.GraphicsContext)

3 . Drawing the snake's body. It was first thought to draw it using [array], but its problem is that it cannot be added or deleted from the matrix, because every time it eats food, its body increases.

Solution of (3).

Solved using linkedlist.

(LinkedList<Point> snakeBody = new LinkedList();)

4. As we use class Point to add another part of the snake after eating fruit. the part added appears in another location in the pane.

Solution of (4).

we use point with coordinate outside the Pane like (-1, -1).

5. The problem of adding the sound in terms of the track to the program.

Solution of (5).

The audio track is placed and then the mode is placed .mp3

Because of using function (Mediaplayer).

6. Add images with different path from jpeg & png.

Solution of (6).

The image track is placed and then the mode is placed is placed png (fruit) & jpg(cover).

7. Different between Arraylist & linkedlist.

Solution of (7).

1. ArrayList:
   - Uses a dynamic array to store elements.
   - Provides fast random access using indices.
   - Efficient for insertion and deletion operations at the end of the list.
   - Consumes less memory.
2. LinkedList:
   - Uses a doubly linked list structure to store elements.
   - Does not provide direct random access using indices.
   - Efficient for insertion and deletion operations.
   - Consumes additional memory for maintaining the linked structure.

☞The problems which we didn’t solve:

1. **When defining (Down, Up, Left, Right,) they were defined as int and were not defined as is string.**

```
private static final int RIGHT = 0;
private static final int LEFT = 1;
private static final int UP = 2;
private static final int DOWN = 3;
private boolean pause =false;
```

2. **constructing a ComboBox , configure the game's difficulty levels easy, medium, and hard and position it on the timeline.**

3. there are problems with making the back button inside the game, which leads to a malfunction with space.

4. The problem of adding the sound in terms of the track to the program.

5. Eating appears in the same places as the snake’s body.

👉 Ideas Include some recommendation about ideas to extend this project in the future:

1- "Adding barriers to the game to turn it into a maze system where the snake enters and loses when it fails to exit or collides with a wall."

2- "Adding a button to the game that allows the player to change the color and shape of the snake."

3- "Increasing the snake's speed as its size grows."

4- "When the snake eats a certain number of food items, the game stops and the player achieves the highest score."

5- "When the player loses, they have a certain number of attempts to continue the game from the point of failure."

6- "Converting the score into virtual coins that the player can use to purchase extra attempts or new snake designs."

LINKS

1. OUR Drive.
   (Project - images – pdf- video)

👉 https://cutt.us/Errorteam

SOURCES

1.harmash.com. (https://2u.pw/TNLfq4).

2.GITHUP. (https://2u.pw/jrCPQO).

3.chatgpt.