

# HEX GAME PROJECT REPORT

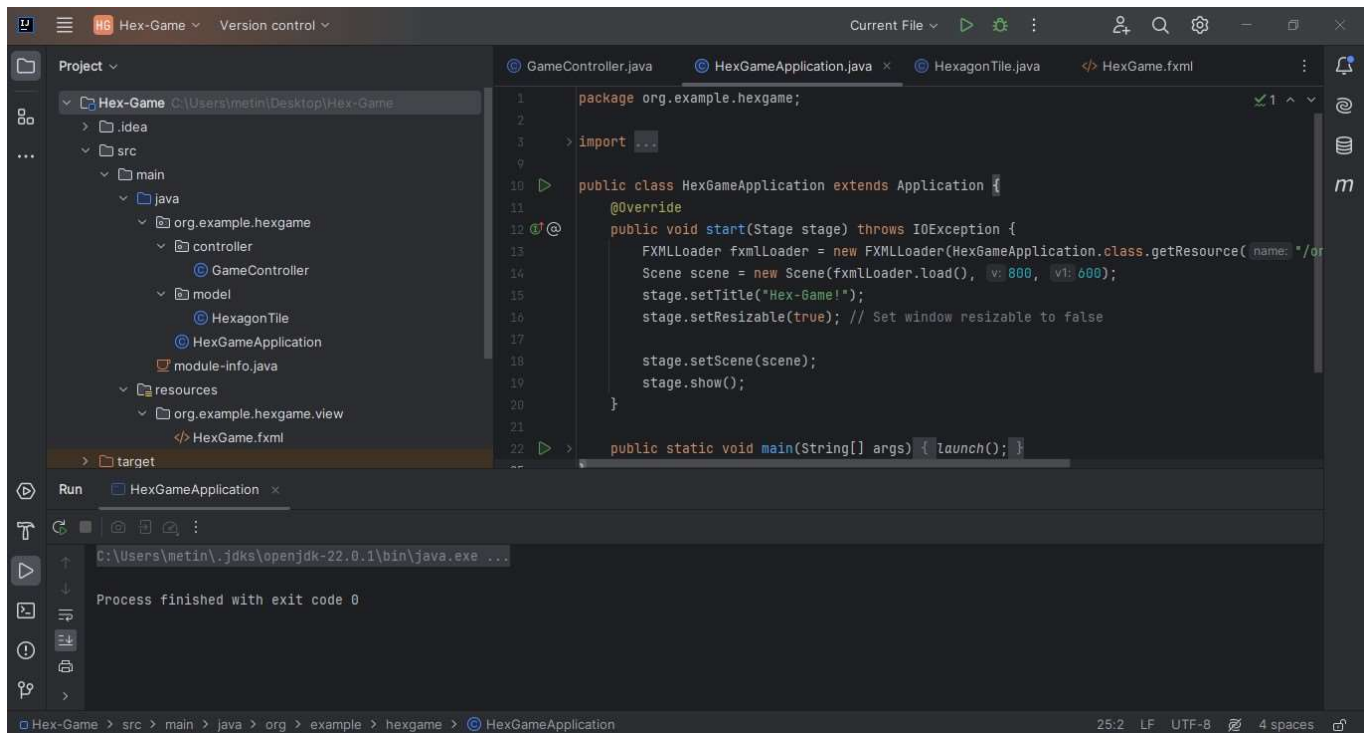
25/05/2024

Project Name: HEX GAME	Project Owner Name Surname: Metin YURDUSEVEN
Creating a graphical interface using JavaFX, writing backend code, preparing project documentation.	

**NOTE:** You can access the project from my github account.

**GITHUB LINK:** <https://github.com/metinyurdev>

**NOTE:** This project was developed using IntelliJ IDEA with Oracle OpenJDK version 22.0.1. We recommend using this environment for the best experience.



This project aims to develop a Hex Game composed of hexagonal tiles. The game offers three different grid size options: 5x5, 11x11, and 17x17. Players take turns marking empty tiles, with the game featuring two players represented by red and blue colors. A player wins by creating an unbroken path of their tiles. Developed using JavaFX and Scene Builder, the project delivers a strategic gaming experience with a user-friendly interface. This report details the project's overall structure, development process, and results.

The Hex Game project consists of three main classes and an FXML file developed on the JavaFX platform. The software design is structured to manage user interactions, implement game logic, and create the visual interface.

## 1. HexGameApplication Class

Description: This class serves as the main application class, creating a hexagonal tile-based game using JavaFX.

Methods: The `start` method loads the FXML file, constructs the user interface, and adds it to a scene. The `main` method launches the JavaFX application.

## 2. HexagonTile Class (within the model package)

Variables:

- `playerColor (Color)`: Indicates the current player color of the tiles.
- `centerX (double)`, `centerY (double)`: Stores the center coordinates of the tiles.
- `radius (double)`: Specifies the radius of the hexagon.
- `isFirstOrLast (boolean)`: Indicates if the tiles are in the first or last column.

Methods:

- `HexagonTile`: Constructs a hexagonal tile object.
- `calculateHexagonPoints`: Calculates the vertices of the hexagon.
- `getPlayerColor`: Returns the color of the tiles.
- `setPlayerColor`: Sets and updates the color of the tiles.

## 3. GameController Class (within the controller package)

Variables:

- `Label playerTurnLabel`: Displays the current player's turn.
- `Button startButton`: Button to start the game.
- `Pane hexGridPane`: Area where hexagonal tiles are placed.
- `RadioButton radio5x5`, `radio11x11`, `radio17x17`: Radio buttons to select different grid sizes.
- `Color currentPlayerColor`: Holds the current player's color.

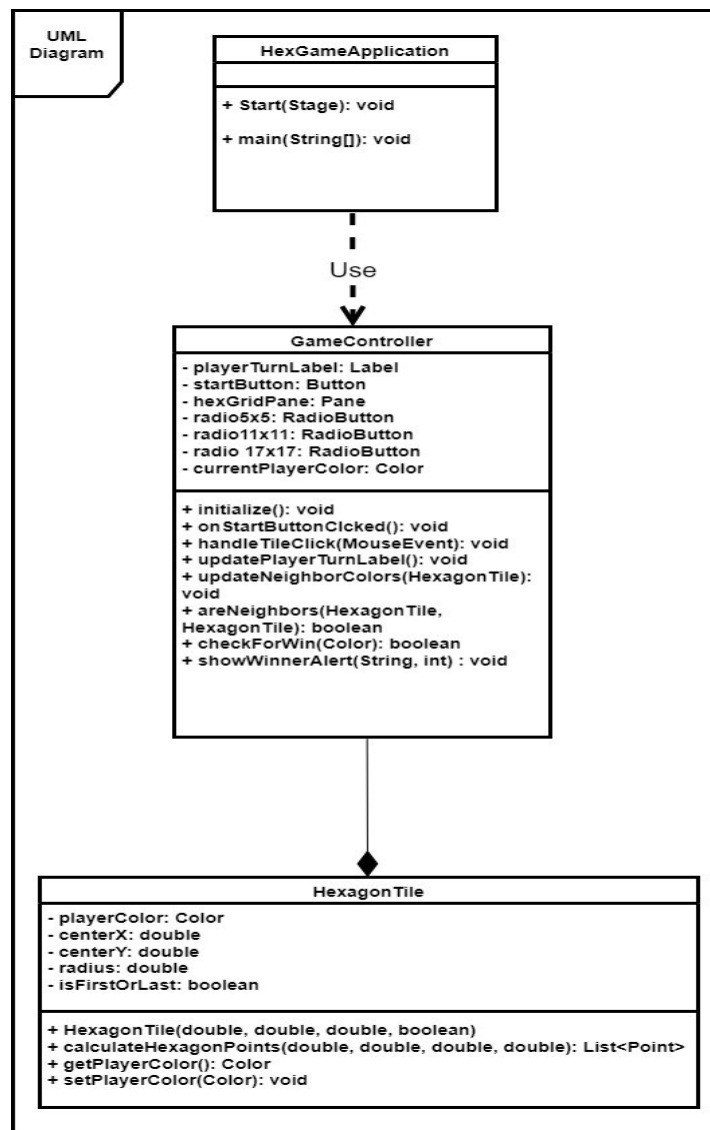
Methods:

- `initialize`: Sets up the radio buttons and selects the default 5x5 size.
- `onStartButtonClicked`: Places tiles according to the selected grid size.
- `handleTileClick`: Changes the color of a tile when clicked and updates neighboring tiles.
- `updatePlayerTurnLabel`: Updates the player's turn.
- `updateNeighborColors`: Updates the edge colors of neighboring tiles.
- `areNeighbors`: Checks if two tiles are neighbors.
- `checkForWin`: Checks if a player's tiles form a connected path.
- `showWinnerAlert`: Displays an alert when a player wins.

#### 4. HexGame.fxml

- Structure: Created using the BorderPane layout manager and consists of three main sections:
- Center: 'Pane' component where hexagonal tiles are placed.
- Bottom: 'HBox' containing radio buttons and the start button.
- Top: 'Label' displaying the player's turn and game status.

Below is a UML Diagram of our project.



The Hex Game project is a strategic game developed using JavaFX and Scene Builder. It offers three different grid size options, a user-friendly interface, and a game logic responsive to player interactions. The goal is for players to create an unbroken path to win. During development, efforts were made to write clean and understandable code, although the full implementation of SOLID principles was not achieved, marking a significant shortcoming. Nonetheless, successfully completing the project within the targeted timeframe is a notable achievement. This project has provided opportunities to enhance both software development skills and user interface design capabilities. Future projects will benefit from the experiences gained in this project, leading to even greater success.