

BZ 214 Visual Programming Project Report

Group No: 020

25/05/24

Number: 1030510701	Name Surname: Gevher Büşra Yetişen
- Coding the classes "TheHexGame", "HexTypedLists_TypeChanger", "Hexagon", "TheGameBoard"	
Number: 1030510779	Name Surname: Saliha İşgören
- Coding the classes "StartButton", "EndGame" - Coding the methods "moveCheck" & "showWrongMove" in "MoveChecking" class	
Number: 1030510551	Name Surname: Ece Cingöz
- Coding the classes "SizeOfBoard", "Supplies", "Move" - Coding the method "overlap" in "MoveChecking" class	

Our Working Process:

Every person on the team had tried their best to make the project good. Each member had written code, and we checked each other's works to make it better. We started with a planning process, firstly we defined the classes and some common variables; then we shared classes and started to coding process. Our UML and Use-Case diagrams is designed by Saliha İşgören & Ece Cingöz.

How our game works;

Our game starts with an empty board and the player turn set to 1. The player selects a hexagon on the board. The MoveChecking class checks if the move is valid. If the move is valid, the Move class changes the color of the hexagon to the current player's color and updates the playersHexMatrix. The EndGame class checks if the current player has won the game. If the game is not over, the player turn is switched to the next player. The game continues until one player wins or the board is full.

The Classes Tasks in shortly:

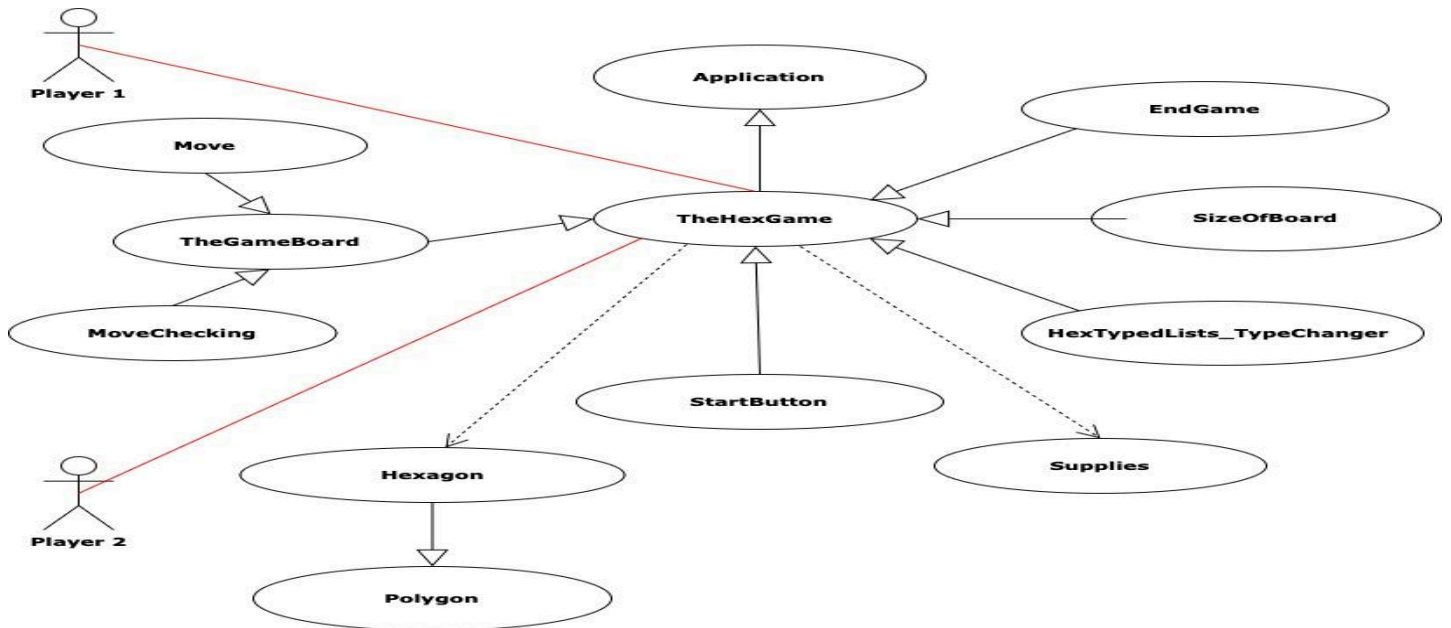
- 1. TheHexGame Class:** This is the main class that initializes the game and sets up the scene. It creates instances of other classes like StartButton, SizeOfBoard, Supplies, TheGameBoard, and MoveChecking. It handles user interactions with the start button and radio buttons for selecting the game board size. It adds the game board and other elements to the scene.
- 2. Supplies Class:** This class provides methods for creating labels and buttons used in the game interface. It defines labels for displaying information about the game, player turn, and winner. It also defines a button for setting the game board size.
- 3. SizeOfBoard Class:** This class provides methods for creating radio buttons to select the game board size. It defines radio buttons for 5x5, 7x7, and 11x11 game boards. It handles user selection of the game board size and stores the selected size in a variable.
- 4. TheGameBoard Class:** This class is responsible for creating the hexagonal game board. It calculates the positions of the hexagons based on the selected size and arranges them on the screen. It creates instances of the Hexagon class for each hexagon on the board. It adds event handlers to the hexagons to handle player moves.
- 5. Hexagon Class:** This class represents a single hexagon on the game board. It stores the coordinates, size, and color of the hexagon. It provides methods for drawing the hexagon and handling mouse clicks.
- 6. MoveChecking Class:** This class checks if a player's move is valid. It checks if the selected hexagon is empty and not overlapping with another player's hexagon. It displays an error message if the move is invalid.
- 7. Move:** This class handles player moves. It changes the color of the selected hexagon to the current player's color. It updates the playersHexMatrix to track the positions of each player's hexagons. It checks for win conditions after each move.
- 8. EndGame:** This class checks if a player has won the game. It uses a depth-first search algorithm to determine if a player has connected their hexagons from one side of the board to the other. It displays a message announcing the winner of the game.
- 9. StartButton:** The Start Button class displays the game starting message and starts the game when the Button is clicked
- 10. HexTypedList_TypeChanger:** This class changes the shape of the structure that holds the Game Board.

Process Of Software Design

We tried to design our project process at the start but according to the needs there have been some changes. In lots of topics sometimes we couldn't find a way to solve our problems. In that kind of situation we get help from a relative; Hakan Aydın, a classmate; Berra Nur Betül Bolat and Google's artificial intelligence; Gemini. We mostly tried to fix our problems on our own but sometimes we used Gemini to show us a road path.

At the end there is some working expectations in our code but we couldn't achieve to handle and find the problems . We tried our best and learn a lot more about Java in this project.

Use-Case Diagram



UML Diagram

