CSCI 5003 Team Project - New Feature

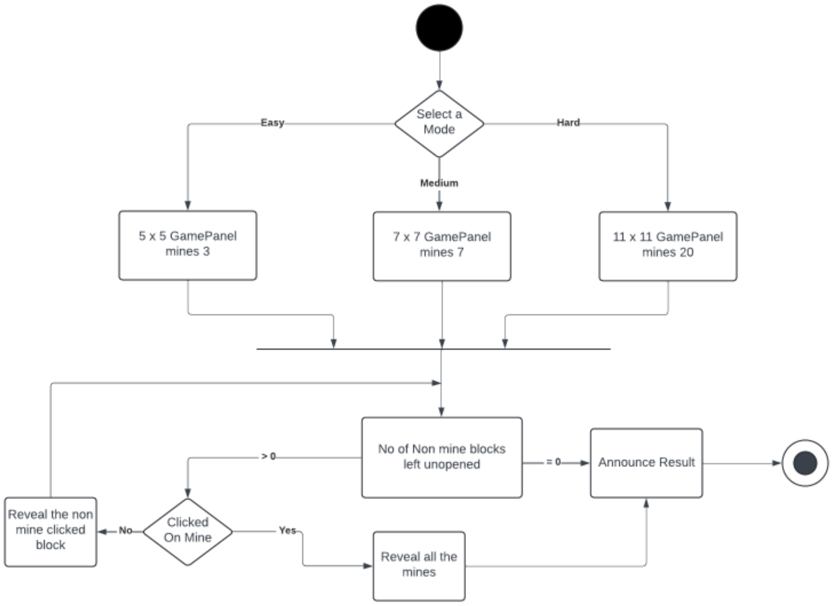
Project Title: Minesweeper

New Feature Description

The new feature entails implementing a Minesweeper game with three distinct difficulty modes that dictate the size of the game board and the density of mines. This addition transforms the user experience by providing variable gameplay complexity and requires updates to the model and view components to support these modes.

This feature introduces a Difficulty Mode Selection, allowing players to choose from Easy, Medium, or Hard modes, each with corresponding board sizes and mine counts—5x5 with 3 mines, 7x7 with 7 mines, and 11x11 with 20 mines, respectively. Upon selection, the Game Board Initialization process kicks off within the model layer, specifically in the **Game** class, which generates a new game grid populated with **Blocks** instances. These **Blocks** represent the individual cells on the Minesweeper board, some of which are randomly assigned as mines based on the chosen difficulty. This setup phase is critical as it tailors the game complexity to the player's preference and lays the foundation for the game's strategic play.

Activity Diagram



Design Changes

To support the new feature, which involves allowing users to select a difficulty mode that determines the board size and mine density for the Minesweeper game, we will need to modify the following classes and methods:

Game.java (Model)

This class will be primarily responsible for initializing the game board based on the selected difficulty. The constructor and several methods need to be adapted or added:

public Game(int rSize, int cSize, int totalMinesCount): Currently, the constructor initializes the game with a given number of rows, columns, and mines.
public void placeMines(int totalMinesCount): The existing PlaceMines method should be updated if any changes are required to the mine placement logic to accommodate the varying board sizes.

public void checkMines(int r, int c): check for the mines in the game.

public int countMines(int r, int c): Counts the number of mines**.**

public void revealMines(): Reveals Mines in the game.

Blocks.java (Model)

The Blocks class extends Jbutton and represents each cell or block on the Minesweeper game board. Given the current properties and the constructor, it captures the row and column position of a block.
public Blocks(int row, int columns)

GameGUI.java (View)

The GameGUI class is responsible for rendering the user interface of the Minesweeper game. To support the new difficulty mode feature, we need to modify this class to enable the user to select a difficulty mode which will then configure the game board accordingly. Here are the proposed changes:

public void run(): It constructs a title panel to display the game's title and sets up a menu panel with buttons for selecting the difficulty level (Easy, Normal, Hard). These buttons are added to the menu panel.

GamePanel.java (View)

The GamePanel class is part of the view in the MVC architecture for the Minesweeper game and extends JPanel. It's designed to display the game's information and board to the user.

public GamePanel(): The constructor sets up the layout of the panel and initializes key components such as the infoPanel for displaying time, and a grid of buttons representing the game cells.

public JPanel createPanel(String mode, Game game): The createPanel method takes a game mode (such as "easy", "normal", or "hard") and a Game object to set up the game board accordingly. It adjusts the game panel's layout based on the selected mode and populates it with instances of Blocks, which are custom buttons that represent the individual cells on the game board.

public void update(String result): Additionally, the class implements the Observer interface, indicating it's designed to be notified of changes in the game state. This method is used to change the display based on the results of the game, such as showing the number of mines or announcing the game's outcome.