

GUI Structure

Interfaces and Abstract Classes

1. `GUI` Interface

Declares methods:

- `menuOutput()`: To be implemented by classes that handle menu display.
- `boardOutput()`: To be implemented by classes that handle the game board display.

2. `initialization` Class

Responsible for initializing the game board with random elements:

- Uses `UniqueRandomNumbers` and `CreateRandom` for generating unique random numbers.
- Creates a 32x32 board, sets border walls, entrance, exit, and random obstacles/rewards.

3. `jframeblueprint` Abstract Class

- Extends `initialization`.
- Provides a method `image()` to add images to the GUI using `JLabel`.

4. `jframe` Class

- Extends `jframeblueprint`.
- Manages the graphical representation of the game board.
- Utilizes a grid layout to display cells with different colors based on the type of element (walls, rewards, obstacles).
- Implements button functionality to save the board to a file.
- Overrides `paintComponent()` for custom drawing.

Inner Classes in `jframe`:

- `TopHalfColoredPanel`, `BottomHalfColoredPanel`, `LeftHalfColoredPanel`, `RightHalfColoredPanel`: Custom `JPanel` classes for drawing colored halves.

5. `CreateRandom` Class

- Generates random numbers for the initialization process.

6. `UniqueRandomNumbers` Class

- Generates unique random numbers for the initialization process.

7. `MyGui` Abstract Class

- Extends `JFrame`.
- Implements the `GUI` interface methods.

8. `Output` Class

- Extends `MyGui`.

Main Application Classes

9. `BoardObject` Abstract Class

- Implements the `Drawable` interface.
- Represents common properties and methods for both obstacles and rewards.
- Contains attributes `positionX` and `positionY` to store the object's position on the board.

10. Concrete Classes

- `Obstacle` and `Reward` Abstract Classes:
 - Extend `BoardObject`.
 - Introduce additional properties (`damage` for obstacles, `value` for rewards).
- `Wall` Class:
 - Represents a specific type of obstacle, extending `Obstacle`.
 - Overrides the `draw` method to draw a black rectangle.
- `Coin` Class:
 - Represents a specific type of reward, extending `Reward`.
 - Overrides the `draw` method to draw a yellow oval.

11. `Board` Class

- Represents the game board.
- Extends `JFrameBlueprint` for GUI rendering and implements the `Serializable` interface for object serialization.
- Initializes the board with an entrance, exit, walls, and rewards.

- Provides methods like `setEntranceAndExit`, `placeWalls`, `placeRewards`.
- Supports saving (`saveBoard`) and loading (`loadBoard`) the board using object serialization.
- Overrides `paintComponent` for drawing the board on the GUI.

12. `BoardFrame` Class

- Represents the main frame of the GUI application.
- Extends `JFrame`.
- Provides a menu bar with options to design a new board and load an existing board.
- Utilizes a `JFileChooser` for loading boards.
- Updates the `Board` instance based on user actions.

13. `testclass1` Class

- Contains the `main` method to launch the Swing application.
- Creates an instance of `BoardFrame` and sets it visible on the Swing event dispatch thread.

Chart of the OO design implemented.

```
-----  
| initialization |  
-----  
| - uniqueRand: UniqueRandomNumbers |  
| - rando: CreateRandom |  
| - board: int[32][32] |  
-----  
| + initialization() |  
-----  
| jframeblueprint |  
-----  
| - image(String, int, int, int, int, JFrame): JLabel |  
-----  
| jframe |  
-----  
| - boardSize: int |  
| - squares: JPanel[][] |  
| - length: int | | - height: int |  
| - newBoard: boolean |  
| - frame: JFrame |  
-----  
| + BoardGame() |  
| + menu() |  
| - TopHalfColoredPanel |  
| - BottomHalfColoredPanel |  
| - LeftHalfColoredPanel |  
| - RightHalfColoredPanel |  
-----  
| MyGui |  
-----  
| + menuoutput(): void |  
| + Boardoutput(): void |  
-----  
| output |
```


| UniqueRandomNumbers |

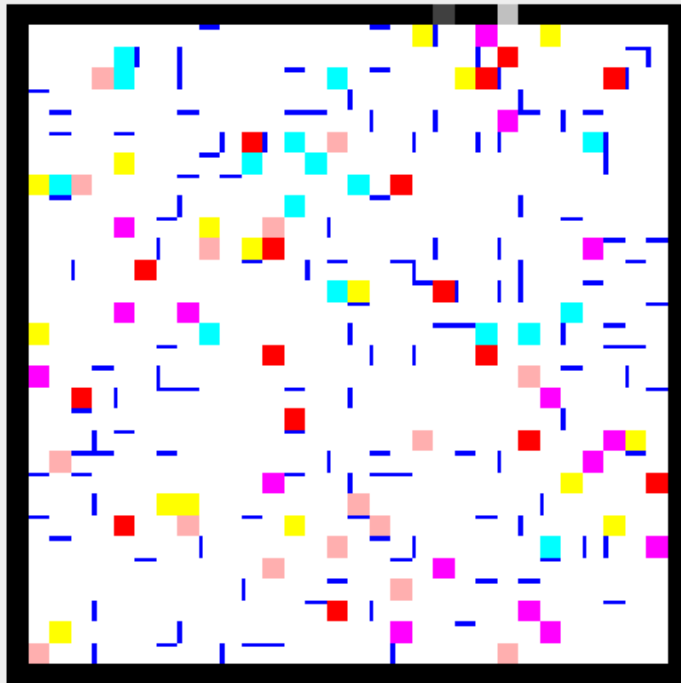
| - rando: Random
| - generatedNumbers: Set |

| + getUniqueRandomNumber(int): int |
| + getUniqueRandomNumber2(int): int |











| CreateRandom |

- rand: Random |

+ getRandomNumber(int): int



SAVE BOARD

	----> WALLS		----> BORDER WALLS
	----> ICECREAM		----> TROLLS
	----> ELIXIR		----> ENTER
	----> CAVE DWELLERS		----> EXIT
	----> COINS		----> BLANK SPACE

