

Practical Software Engineering I.

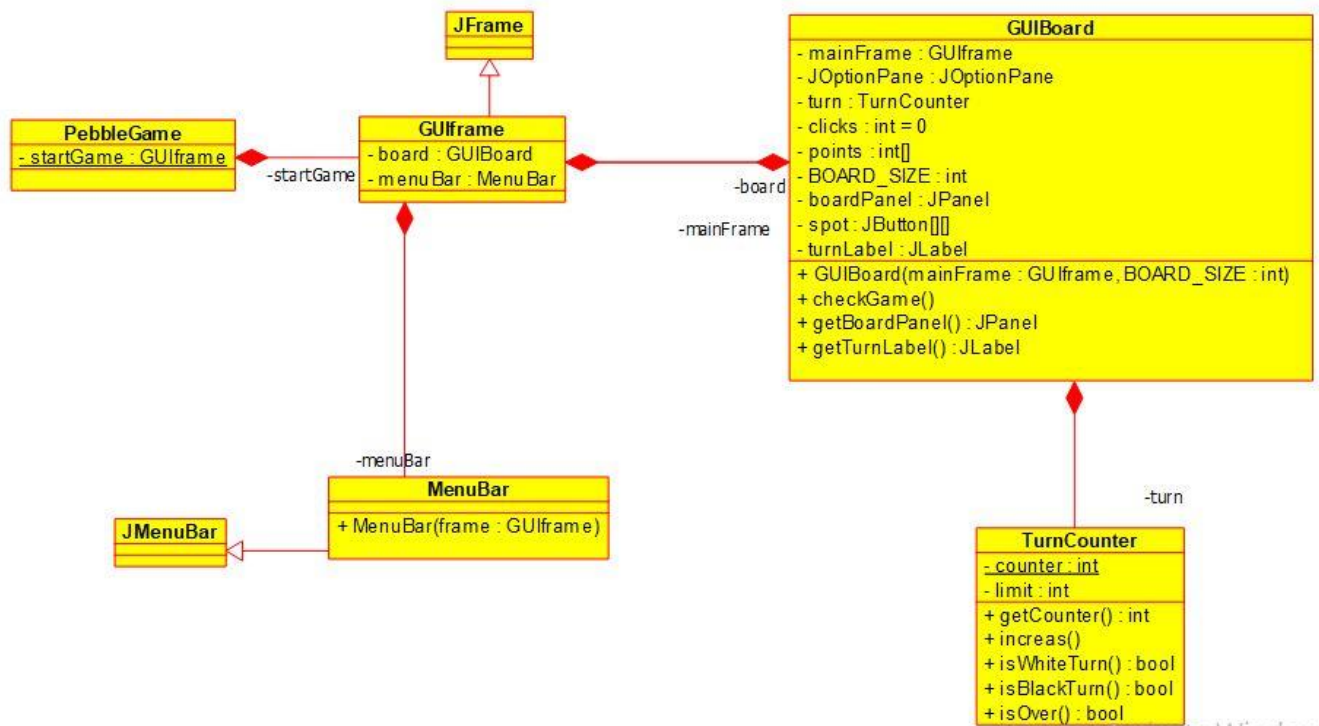
Assigment_2

Exercise description:

2. Pebble

Pebble is a two-player game, played on a board consists of $n \times n$ fields. Initially, n white and n black pebbles are placed on the board randomly. Each color belongs to only one player. The players take turns choosing one of its pebble, and then move it horizontally or vertically. The movement also affects the neighboring pebbles in the direction (the pebble on the edge falls off). The objective of the game is to push out as much pebbles of the opponent from the board as we can, within a given number of turns ($5n$). A player wins, if he has more pebbles on the board at the end than his opponent. The game is draw, if they have the same number of pebbles on the board .

Implement this game, and let the board size be selectable (3x3, 4x4, 6x6 → turns are 15, 20, 30). The game should recognize if it is ended, and it has to show the name of the winner in a message box (if the game is not ended with draw), and automatically begin a new game .

Class diagram :*Program description:*

It is a two-player game, the winner is one who drop the most of other opponent's pebbles of the edge.

If there is still pebbles on the field the winner is the one holding more pebbles.

description of each methods

GUIBoard class includes :

- **ButtonListener class** which include :

-**public void move (Int x , int y , int nextX , int nextY)** : is the **main algorithm** to move pebbles based on coordinates from first buttons clicked to determine which pebble we want to move and the second button which is the movement direction

-implement the **ActionListener** which use the actionPerformed method to select the pebble for first click and move it to the chosen location based on wide range of movement shields (logical conditions) .

- **public void checkGame()** : to check the state of the game and it will notify and start new game .
- **JPanel getBoardPanel()** : return JPanel that contain the buttons .
- **JLabel getTurnLabel()** : return JLabel that show the turn to be put on the frame .

TurnCounter class :

- **int getCounter()** : return the turn counter .
- **void increas()** : increases the turn counter .
- **Boolean isWhiteTurn()** : checks if it is White 's turn .
- **Boolean isBlackTurn()** : checks if it is Black 's turn .
- **Boolean isOver()** : checks if the the number of turns allowed is reached .

Also there is MenuBar class and the GUIframe class ...