**Blokus Game – User’s & Developer Guide**

Isfandyar V., Neerajen S., Sally H.

**Users Guide**

**Background:**

Blokus is an abstract strategy board game for two to four players, invented by Bernard Tavitian and first released in 2000 by Sekkoïa, a French company. The game is played on a square board divided into 20 rows and 20 columns for a total of 400 squares. There are total of 84 game tiles organized into 21 shapes in each of four colors: blue, red, yellow and green. The objective of the game is each player has to ﬁt as many of his/her 21 pieces on the board as possible. Each player chooses a color and places that set of 21 pieces in front of his/her side on the board, each new piece which the player places must touch at least one other piece of the same color but without contact between two sides. The game ends for a player when the player is blocked and can no longer put down any pieces or that player run out of pieces.

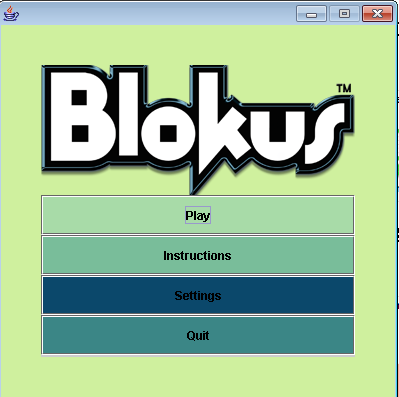
**Getting Started:**

The Phase Game has been coded with Eclipse & Ready to program. The program runs in both Eclipse and Ready to program Java and both applications can be downloaded for free online. Within the .zip file, you will find several files. Open and run the file that is labelled *MainMenu.java*; you may need to compile it first. Once you have the file running, you get acquainted to the main menu where you can select the option to Play the game, see the instructions, set the settings or quit the game. The game is controlled with the arrow keys to move the tiles around the board and mouse is used to select the tiles, rotate and place them.

**Features of the Program:**

**MAIN MENU**

*Buttons* used to select between options



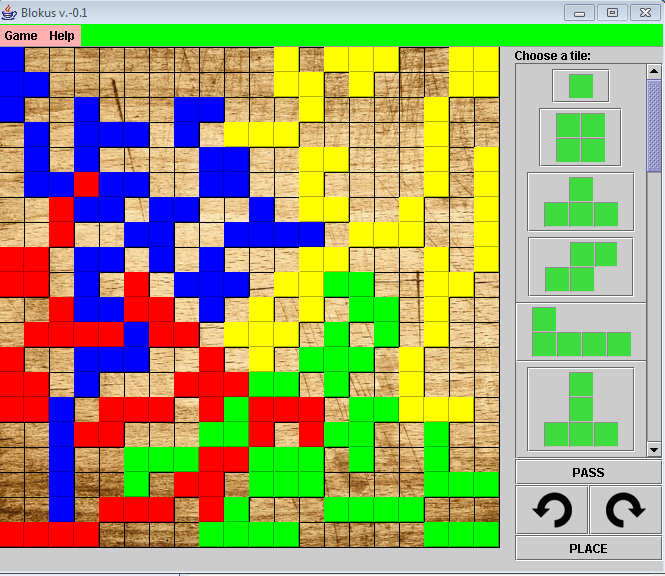
Tiles are places on this board by the user from the scroll bar. Users use their arrow keys to navigate through the board to place their tiles.

The buttons are used here so that the user can pass his turn, rotate the tiles and place them.

Menu bar is used here so that user can restart the game, read instructions, view game statics, change player info, etc.

User can use this scroll bar to select the tiles they want to place. Every time user uses a tile, that color tile won’t be available in the scroll bar. Also, after a user places his/her tile, then the scroll bar’s tiles changes to the next users color.

**GAME SCREEN**



**POP-UP ERROR MESSAGES**

Every time a user makes an illegal move, then an error message pop’s up. These pop-ups are implemented so that the user can know their move is illegal with a reasonable answer.



**Limitations and Bugs of the Program:**

The limitations with the program are that the game requires 4 players to play this game. If the program undergoes with future development, then an intelligent NPC with difficulty system regarding its AI can be added to the program so that a user can play alone.

**Developer’s Guide**

**Overview:**

There are four main components that the Blokus Game requires: the Board class, the TileCollection class, the Player class, the GameScreen class. Also there are three sub components used in this game: the Settings class, the MainMenu class and the Instructions class.

**Sections of each Feature of the Program:**

**Board Class**

This is a very important class, with many subclasses. The board class creates a board to play Blokus. It also adds tiles to the board, checks if the tile is going to be placed on a valid spot, gives out pop-op messages for illegal moves, uses keys to navigate the tiles through the board, determines the final score of each player, and draws the board’s grid. This class implements ActionListener to track keystrokes.

**Important Methods:**

* validateTile: checks if the placement of the tile object is valid with the rules of Blokus and returns a Boolean variable.
* addSelectedTile: void method which adds the tile object to a temporary array, used for moving the tile object around the grid before placement.
* placeTile: void method which adds the tile object to boardTracker [] [] and resets all coordinates for the next player’s turn.
* determineScore: determines the score of all the players and returns it in an array of integers.

**Tile Class**

The Tile Class is used to create a tile object, which is composed of a rectangular Boolean grid that has true values to determine the shape of the tile.

**Important Methods:**

* rotate: rotates the tile object ninety degrees about the origin, changing the tile’s orientation
* flipHorizontal: flips the tile object horizontally
* flipVertical: flips the tile object vertically

**Note:** Although there are not options for flipHorizontal() and flipVertical () on the Graphical User Interface, this is because the rules of Blokus state that tiles are not to be flipped. However, these methods were made to further development if desired.

**TileCollection Class**

The TileCollection Class creates 21 different tile objects required to play Blokus and stores them in an array for the program to select from.

**Player Class**

The Player Class contains the name of the player, and the tiles. This makes it possible for the game to differentiate between the players and making it possible to replace the tiles as the player switches.

**GameScreen Class**

GameScreen Class is the class with all the main GUI of the game. It includes a Menu bar so that user can restart the game, check instructions, change appearance and information, and do much more. Also, the class includes a scroll bar which keeps all the tiles in them, so that user can select tiles from it. Also, it includes buttons for skipping turn, placing and rotating. It also includes a lot of thinking methods of the program. For example: It will declare winner when all players pass.

**MainMenu Class**

The MainMenu Class is used as the startup face of the game. It is used to display an amazing menu for the game, with JButtons for the user to choose between the options to start the game, check the Instructions and setup the Settings, and Quitting the game.

**Settings Class**

Settings Class is used so that the Users can setup their name change the theme and background color. It isn’t an import class. Its purpose is so that users have an option to change their names, and the color scheme of the program.

**Instructions Class**

Instructions Class is used to display the instructions about the program. This is not an import class, but it is used in the main menu class. The objective of this class is that the users can learn how to play the game.

**Suggestions for Improvement:**

We could have created various game modes for the game so that user can play with different versions. Example: Time Mode, where a user have to do a move in a certain amount of time. We could also have created a way so that user could play with the computer. Taking it even further, we could have added difficulty to that computer, so that the computer’s AI gets better as the user desires. We also could have added a better user friendly GUI which might include the use of mouse to place the tiles, or making it possible for the user to drag and drop his selected tile to the board. Also, the game only ends when all four players pass. A method could be created that determines if any more moves are possible and then acts accordingly.