**TETRIS GAME**

Member 1 : Vineet Kaur Sethi

Member 2 : Snoviya Dcunha

Member 3 : Varsha Kulkarni

Duration of Project: 9/5/2017- 17/7/2017

**Project Name : TETRIS**

# Abstract

INTRODUTION:

Tetris is a popular video game. Tetriminos are geometric shapes of four (tetra) square blocks each and hence the name “tetris”. A random sequence of tetriminos fall down in the playing field. The objective of the game is to manipulate these tetriminos by moving them one sideways and rotating them by 90 degrees, with the aim to creating horizontal line without gaps .When such line is formed, it disappears, and any block created above will fall. When certain number of lines are cleared, the tetriminos fall faster, and the game ends when the stack of tetriminos reaches the top of the playing field.

OVERVIEW:

At the start of the game, a dialog box appears asking for the player’s name. On entering the name, the game begins with the count of 3 2 1.. The tetriminos of random shapes and colours fall in the playing filed. They can be manipulated by moving them sideways or rotating them by 90 degrees to create a horizontal line with no gaps. Each horizontal line formed with no gaps is cleared, giving you 100 points. As game proceeds, the tetriminos move faster. To pause or exit the game, options are provided on the right of the playing field.

**Completion status:**

Project is successfully completed.

# Softwares used

* List of software used : Blue J, NetBeans.
* Detail of software: Blue j 3.1.5 download link
* Detail of software: NetBeans IDE 8.2 download link

# Software and Code

[Github link](http://www.github.com/) for the repository of code: https://github.com/VineetKaur/inheritance-

**BRIEF EXPLAINATION OF THE CODE:**

***Constructor****:* To initialize all the variables with default values and to read the previous high score from file.

***Void init():*** To set the different properties of the main JFrame such as resizable, focusable, etc.

***Void setPolygon():*** A random shape is selected from five shapes (that are L-Shape, T-Shape, Z-Shape, Line and Square block) and the area is stored in the array.

***Void setRandColor():*** A random color is created having random values of RGB. This color is stored in the array.

***class KeyManager:*** This inner class listens for key events of left, right, up and down arrow key and changes the respective values.

***void move() :*** This moves the current falling shape on basis of the keys pressed after checking the limits.

***boolean intersectsShape(Area shapeA, Area shapeB):*** Checks if the shapes are intersecting.

Returns true if they will intersect else false.

***boolean collisionWithSideWalls(Area p) :*** Checks if the shape collides with the side walls of the playing field.

***boolean collisionWithBottomWall(Area p):*** Checks if the area of shape collides with the bottom wall of the playing field.

***void getName():*** Displays the dialogue box to get the input of the player name.

***class DrawPanel1 extends JPanel :*** Draws the playing arena along with all the shapes that have fallen and displays the details such as player name, level, score etc. It also displays the buttons to pause or exit the game. It also rotates the shape if the up arrow key is pressed.

***class PauseList implements ActionListener:*** This class acts a listener for the pause button. Another frame is made visible and the user is given the choice to continue the game or exit. Value of the paused variable is changed to true.

***class ContListener implements ActionListener:*** This class acts as listener to the continue button. The game is made visible again and the paused frame is removed. Value of paused variable is changed to false.

***class ExitListener implements ActionListener:*** This class acts a listener for the exit button. Another frame is made visible and the user is given the choice to either exit the game or not. Value of the paused variable is changed to true.

***class Exit implements ActionListener:*** acts as a listener to the yes button on exit screen. The game is closed, frames are disposed off.

***class MoveThread implements Runnable:*** it moves the shape down after a definite time delay depending on the level.

***public void run():*** All the main logic is present in this method. Infinite loop is set up. Random shape with random color is created. Another infinite is loop is set up. If the current shape intersects with another shape or collides with the bottom wall one loop is exited.

It is checked if any line is completed. If it is, the area of that line is deleted and the shapes above it are brought down.

It is checked Whether the last block that came down touches the top wall when it has stopped moving. If it is so the game is over and the infinite loop is exited.

If the score is higher than the previous high score, it is printed in the file.

***public static void main():*** Call the run method and displays the animation using the animg class object.

***Class animg:*** contains the starting animation and the game over animation.

***Class Sound*** : It plays the background sound through out the game.

# Use and Demo

Screenshots of the project

**Instructions for user:**

1. Use left and right arrow keys to move tetriminos to the left and right respectively.
2. Use down arrow to move the tetriminos faster .
3. Use up arrow key to rotate the shape 90 degrees clockwise.
4. Player’s name, game level and score are mentioned at top right corner.
5. Each time, on clearing 5 rows, level and speed of game increases.

# Future Work

Tetris with new levels and some modification.

# Bug report and Challenges

Row deletion, rotation of the shapes and score display were few problems faced.

Area array was made for shapes to make it easy for rotation and deletion. Score display was tackled by changing panel to draw panel.

# Bibliography

* Tutorial: [https://docmss.djangoproject.com/en/1.11/](https://docs.djangoproject.com/en/1.11/)
* Documentation: https://docs.djangoproject.com/en/1.11/