**READ ME AND PROJECT DESCRIPTION**

**Files**

HTML - **Index.html** – To start the game

CSS File – gameStyle.css

Script File - myScript.js

JQuery - jquery-1.11.3.js

**Preferred Browser:** Use Mozilla Firefox version 42 or 42+

The game works the best in this browser so use this browser for testing the game.

**Tertris Game Project**

**Introduction**

The Game is similar to a classic Tetris Game. With normal rules like new objects will appear and you need to make lines through the objects. The score is incremented as a line is formed.

Have applied shadow, transitions and animation effects. Hover on button will animate and increase font. Hove out will make it regain original font size.

Have made use of case statements, loops and functions

**Start Game:**

Objects start coming from top.

User can restart game and is alerted for restarting

**Joystick**:

Up Arrow – Rotate the object in counter-clockwise direction (flip object)

Down Arrow- Move the object in downward direction (reducing Y axis)

Left Arrow- Move the object in left direction (reducing X axis)

Right Arrow- Move the object in right direction (increasing X axis)

User can **select difficulty level** – For Beginner, Speed: Interval 270. Intermediate, speed: Interval 180. Expert, Speed: 90. Every time, after changing the level from drop-down, user should click on canvas so that canvas will respond to up, down, top, bottom keys to move the object respectively

On changing the level, if the user has not yet started the game then the game will get restarted automatically. Else user is in middle of a game then user will be prompt if he wants to restart or not.

If the objects added ultimately get filled and touches canvas top then user loses and is alerted game over. **Score** is incremented as user is able to fill line.

**The architecture of project**

Following approach is used in making of the game:

**Function StartGame**

In the canvas - #rows :30 , #columns: 15

For every new object that is created:

**Function newShape, makeMovingObjects**

* Stores the field as a 2-dimensional array(Dimension used are 4 X 4) - canvasFrame
* The falling Object as a similar is also stored in 2D array that's a 4x4 array.

Challenges faced: Checking that two object do not collide while making them move.

**Function isObjectAcceptable**

* Making sure that user can’t rotate object in such way that it would get stuck in the wall or outer canvas.

**Keypress of left arrow : Function makeMovingObjects**

* For the object, move it to the left by decreasing the X axis for the object, making sure it can go till left till the walls of the canvas.

**Keypress of Up arrow : Function makeMovingObjects, rotate**

* Rotate the falling block by rotating the contents of the according array (more or less means circularly swapping the array cell values around). Rotation happens

**Keypress of right arrow : Function makeMovingObjects**

* For the object, move it to the left by increasing the X axis for the object, making sure it can go till left till the walls of the canvas.

**Keypress of Down arrow : Function rotate**

* Rotate the falling block by rotating the contents of the array accordingly (i.e. swapping the array cell values around).

**Function stopObject**

* Pause the object & lay it sit onto the canvasFrame or on another object
* Also check if the object touches/reached the top of the canvas then list game over for the user.

Challenges faced: Iwasgetting uncaught exception for memory because sometimes the object didn’t used to stop. But, then fixed it by checking all the arrays made with undefined and made sure, when column count has reached 30 it may not move further down and stop at that location.

**Function clearLines**

* Clearing the lines by checking if all cells are filled in any given row. If that is so, shift the contents of all rows above the filled line downwards (overwriting it), and increment the score point accordingly.

Challenges faced: Calling this function the correct row wasn’t getting cleared earlier then fixed it and now the exact row is getting cleared.

**Function incrementScore**

Increment the score based on the difficulty level so if difficulty level chosen:

Beginner– levelOfDifficulty for interval - 270, increment score by 100 every time a line is cleared.

Intermediate – levelOfDifficulty for interval - 180, increment score by 200 every time a line is cleared.

Expert– levelOfDifficulty for interval - 90, increment score by 300 every time a line is cleared.