# TETRIS

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER AND SCIENCE ENGINEERING**

**Submitted by**

**Mallikarjun Reddy (121910315001)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GITAM**

**(Deemed to be University)**

**VISAKHAPATNAM**

**September 2022**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned the efforts with success. It is a pleasant aspect that I now have the opportunity to express my gratitude for all of them.

The first person I would like to thank is my project guide Mr. ABC, who has given continuous critical suggestions and extension of a proper working atmosphere, abiding interest has finally evolved into this research work.

It is indeed with a great sense of pleasure and immense sense of guidance that I acknowledge the help and I am highly indebted to Prof. ABC, Principal, and School of Technology, for his support during the tenure of the internship.

I would like to express my sincere thanks to Prof. ABC, Head of the Department of Electronics and Communication engineering for providing the opportunity to Undertake this internship and encouragement in the completion of the project.

I am also thankful to all the staff members of the Electronics and communication Engineering Department for their valuable suggestions. I would like to thank my teammates and parents who extended their help, encouragement and moral support either directly or indirectly in this project.

                                                                                                  MALLIKARJUN REDDY

                                                                                   121910315001

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

**DECLARATION**

I solemnly declare that the project report is based on my own work carried out during the course of our study. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

1. The work contained in the report is original and has been done by me.
2. The work has not been submitted to any other institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.
4. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

    MALLIKARJUN REDDY

**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Chapter | Title |  | Page No |
| A | Abstract |  | 5 |
| I | Objective |  | 6 |
| II | Overview |  | 13 |
| III | Code |  | 19 |
| IV | Reference |  | 26 |

Abstract

Tetris is a very popular game that has served as an inspiration for many puzzle-style, tile matching video games we have. It is a classic. The fan base is growing every year because there are many variations of the game each with its own adjustments and different ways of gameplay. Tetris is a tile matching, line clearing, fun puzzle video game, it has different variations but all follow the major rules of players completing lines or rows by moving and rotating different shapes called tetrominoes which fall from the sky, the game board that shows the grid at which everything is cleared. This project seeks to provide a new variation of the Tetris game, a harder and more compelling version. Previous variations kept the hold and next feature which just makes the game too easy, especially for the Experts. This Tetris project will remove them and may only make them available by earning scores and levels for a temporary time. This project will also include an augmented reality integration to give its final edge. the Tetris industry provided another variation called Tetris effect that heavily focuses on effects and somewhat time manipulation which has helped the Tetris community experience another addition of the game. Each of these variations have concepts such as, speed increase, special effects, colors , world environments, game boards and number of tetrominoes /shapes possible to clear rows, for example DodecaTetris and OctoTetris.

**AIM AND OBJECTIVES:**

The aim of this project is to design an enhanced version of Tetris game by making it competitive, rewarding and adding new experiences to the game. To give experienced players more and to give new players ready to join the Tetris family something to look forward to by adding and removing features that can make this feasible.

* To increase player experience by addition of AR
* To make the game more challenging
* Removing the hold feature
* Removing the next feature
* Making bonus features derived from the removal of the hold and next features ▪ Addition of Augmented reality to the game

# Used languages:

**HTML:**

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation ([CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)) or functionality/behavior ([JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript)).

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as [<head>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head), [<title>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title), [<body>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/body), [<header>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header), [<footer>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer),

[<article>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article), [<section>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section), [<p>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p), [<div>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div), [<span>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span), [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img), [<aside>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/aside), [<audio>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio), [<canvas>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas), [<datalist>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist),

[<details>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/details), [<embed>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/embed), [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav), [<output>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/output), [<progress>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress), [<video>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video), [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul), [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol), [<li>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/li) and many others.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Create Web site - You can create a website or customize an existing web template if you know HTML well.

Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.

Understand web - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield the best results.

Learn other languages - Once you understand the basics of HTML then other related

technologies like JavaScript, PHP, or angular are becoming easier to understand.

**Applications of HTML:**

* As mentioned before, HTML is one of the most widely used languages over the web. I'm going to list few of them here:
* Web pages development - HTML is used to create pages that are rendered over the web.
* Almost every page of the web has html tags in it to render its details in the browser.
* Internet Navigation - HTML provides tags which are used to navigate from one page to
* another and is heavily used in internet navigation.
* Responsive UI - HTML pages now-a-days works well on all platforms, mobile, tabs,
* desktop or laptops owing to responsive design strategy.
* Offline support HTML pages once loaded can be made available offline on the machine without any need of the internet.
* Game development- HTML5 has native support for rich experience and is now useful in the gaming development arena as well.

**SIMPLE EXAMPLE:**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1> Heading</h1>

<p> paragraph.</p>

</body>

</html>

# CSS:

Cascading Style Sheets (CSS) is a [stylesheet](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) language used to describe the presentation of a document written in [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) or [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

Like HTML, CSS is not a programming language. It's not a markup language either. CSS is a style sheet language. CSS is what you use to selectively style HTML elements.

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

Create Stunning Website - CSS handles the look and feel part of a web page.

Using CSS, you can control the colour of the text, the style of fonts, the spacing

between paragraphs, how columns are sized and laid out, what background images

or colours are used, layout designs, variations in display for different devices and

screen sizes as well as a variety of other effects.

Become a web designer - If you want to start a career as a professional web

designer, HTML and CSS designing is a must skill.

Control web - CSS is easy to learn and understand but it provides powerful control

over the presentation of an HTML document. Most commonly, CSS is combined

with the markup languages HTML or XHTML.

**Applications of CSS:**

As mentioned before, CSS is one of the most widely used style languages over the web.

CSS saves time - we can reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

Pages load faster - If you are using CSS, you do not need to write HTML tag

attributes every time. Just write one CSS rule of a tag and apply it to all the

occurrences of that tag. So less code means faster download times.

Easy maintenance - To make a global change, simply change the style, and all

elements in all the web pages will be updated automatically.

Superior styles to HTML - CSS has a much wider array of attributes than HTML,

so you can give a far better look to your HTML page in comparison to HTML

attributes.

Multiple Device Compatibility - Style sheets allow content to be optimized for

more than one type of device. By using the same HTML document, different

versions of a website can be presented for handheld devices such as PDAs and cell

phones or for printing.

Global web standards - Now HTML attributes are being deprecated and it is being

recommended to use CSS. So, it's a good idea to start using CSS in all the HTML

pages to make them compatible with future browsers.

# JavaScript:

JavaScript (JS) is a lightweight, interpreted, or [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) compiled programming language with [first-class functions](https://developer.mozilla.org/en-US/docs/Glossary/First-class_Function). While it is most well-known as the scripting language for Web pages, [many non-browser environments](https://en.wikipedia.org/wiki/JavaScript#Other_usage) also use it, such as [Node.js](https://developer.mozilla.org/en-US/docs/Glossary/Node.js), [Apache CouchDB](https://couchdb.apache.org/) and [Adobe](https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/) [Acrobat](https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/).

JavaScript is a [prototype-based](https://developer.mozilla.org/en-US/docs/Glossary/Prototype-based_programming), multi-paradigm, [single-threaded](https://developer.mozilla.org/en-US/docs/Glossary/Thread), [dynamic](https://developer.mozilla.org/en-US/docs/Glossary/Dynamic_typing) language, supporting object-oriented, imperative, and declarative styles. Read more [about JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript).

Javascript is a MUST for students and working professionals to become a great Software

Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Javascript:

Javascript is the most popular programming language in the world and that makes

it a great programmer's choice. Once you learn Javascript, it helps you develop great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.

Javascript is everywhere, it comes installed on every modern web browser and so

to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox, Safari and every browser you know as of today, supports Javascript.

Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience (GUI).

JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as a

Javascript Programmer.

Due to high demand, there is tons of job growth and high pay for those who know

JavaScript. You can navigate over to different job sites to see what having

JavaScript skills look like in the job market.

Great thing about Javascript is that you will find tons of frameworks and Libraries

already developed which can be used directly in your software development to

reduce your time to market.

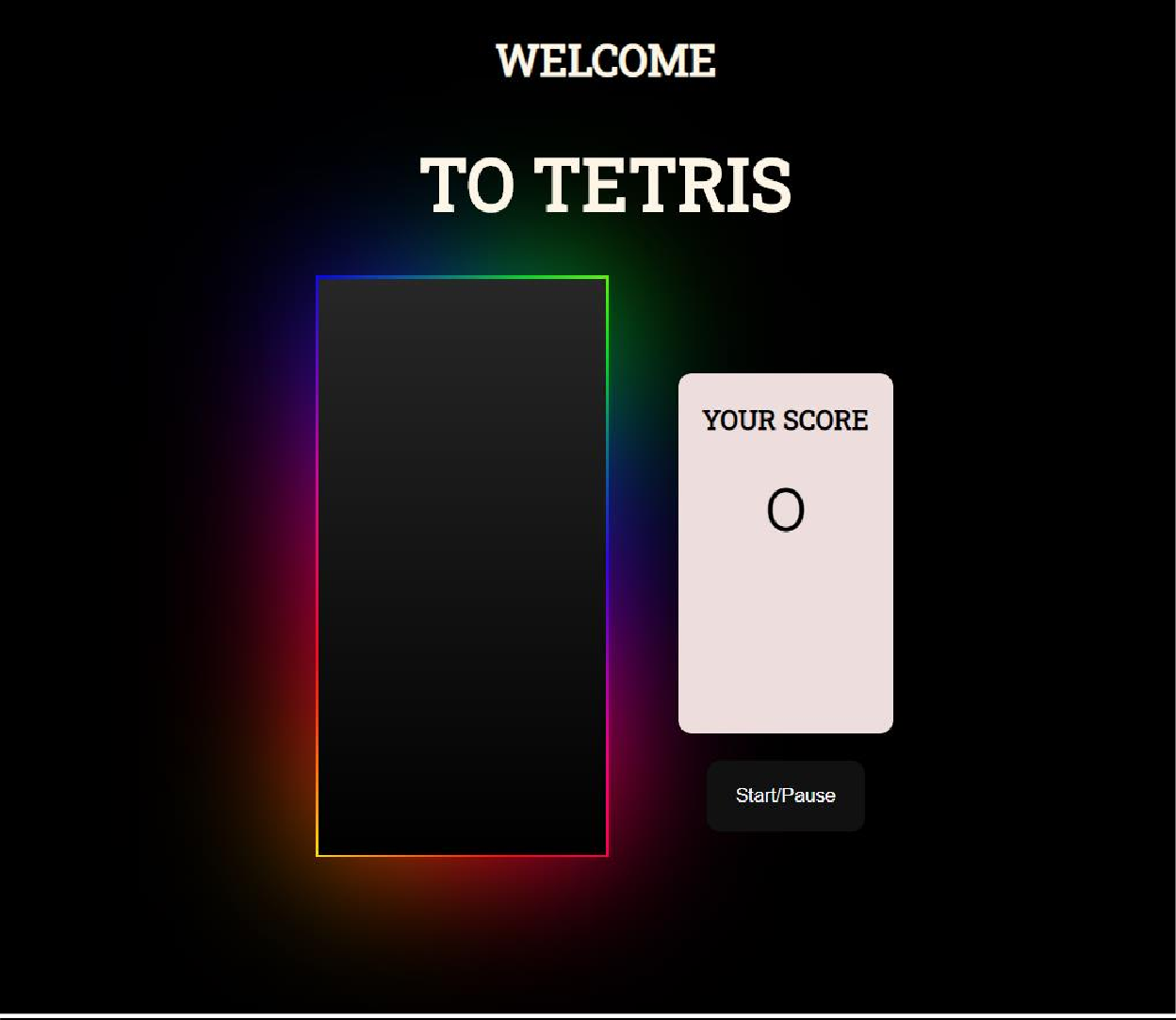
There could be 1000s of good reasons to learn Javascript Programming. But one thing for sure, to learn any programming language, not only Javascript, you just need to code, and code and finally.

# Overview

# Step 1: Design Overview

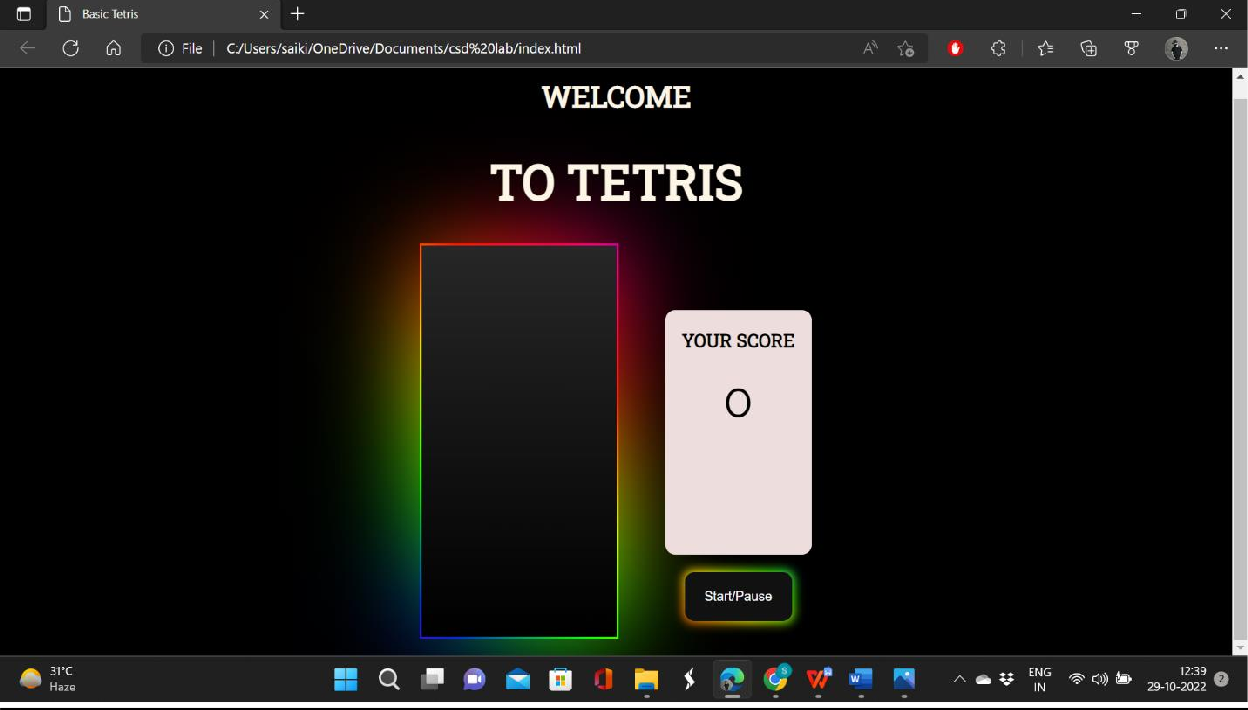
* + These notes assume that you are familiar with the game of Tetris. If not, we suggest that you first read the on Tetris and then play the game a bit.
  + These notes also assume that you are familiar with the JComponentWithEvents class in our 15-100 course.
  + Here is Tetris as we will write it: applet: tetris.html

jar file: tetris.jar



# Step 2: Creating and Drawing the Board

* + For our purposes, a "board" is a 2-dimensional array of Color objects. Our goal here is to allocate the board in a constructor, and then to paint the board in our paint methods.
  + To help us test our code, we will add a few lines in the constructor to pre-load a few cells in the board with some colors. This code will be removed after this step. In particular, we will paint the top-left cell red, the top-right cell white, the bottom-left cell green, and bottom-right cell gray.



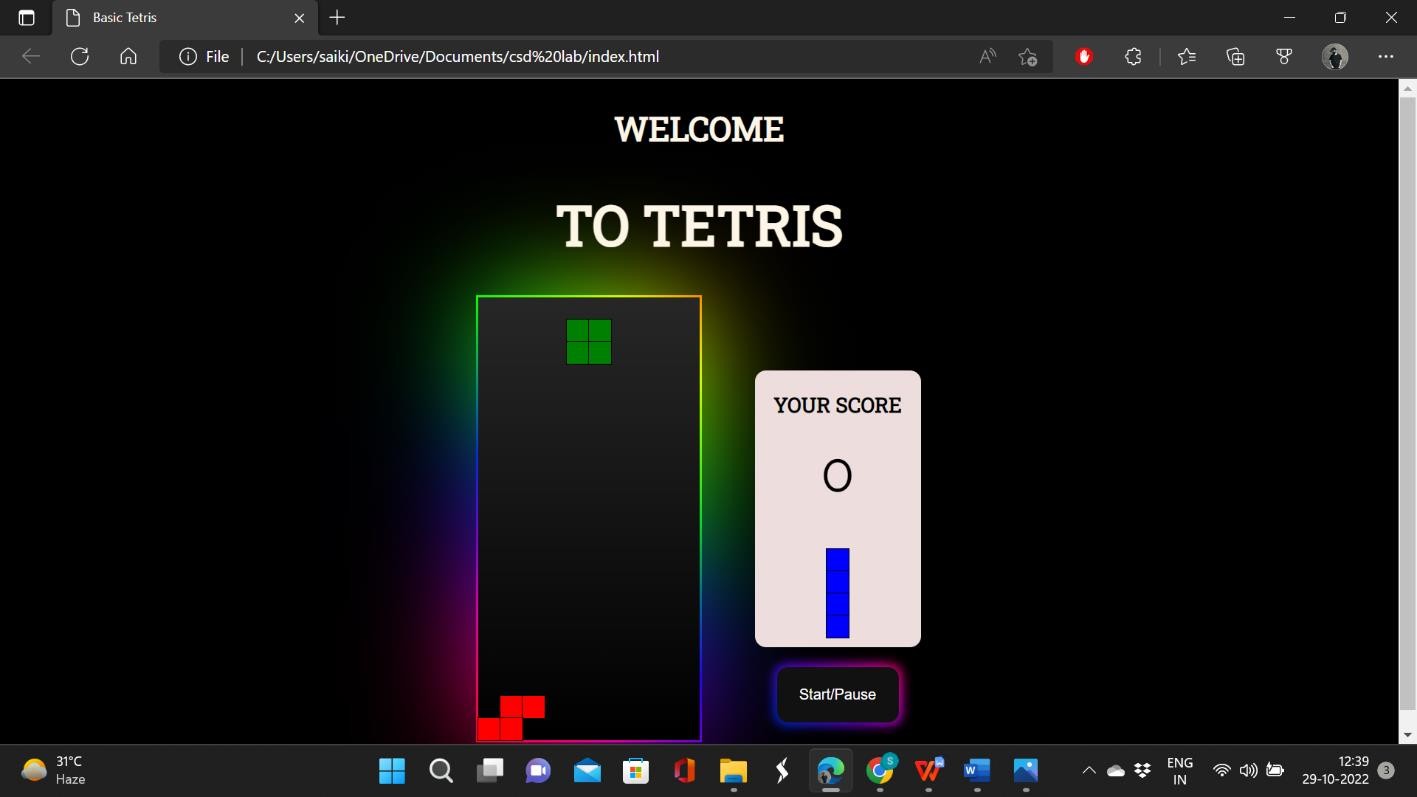
# Step 3: Moving the falling Piece left/right/down

* + In this step, we have our falling piece respond to left-arrow, right-arrow, and down- arrow key presses by moving in the given direction. Also, for testing purposes, we will add temporary code that changes the falling piece whenever any other key is

pressed. This code will be removed after this step.

* + Actually, we only want to move in a given direction if it is legal to do so. There are two reasons why it may not be legal: the falling piece may wind up off of the board, or a part of the falling piece may collide with a non-empty cell on the board. In either case, we should not move the falling piece.
  + Writing key Pressed:

We modify the key Pressed handler to call moveFallingPiece to move left, right, or down in response to left-arrow, right-arrow, or down-arrow key presses (and to change the falling piece in response to any other key press for now).



# step 4: Rotating the falling Piece

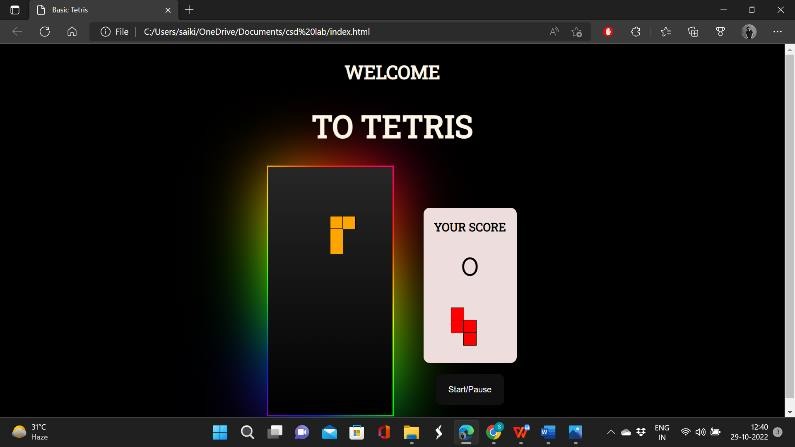
This is generally the most difficult part of the project: in response to up-arrow key presses, we should rotate the falling piece 90 degrees counterclockwise. We do this in the same way we handled other changes to the falling piece: we make the change, test if it is legal, and if not, we unmake the change.

* + Explaining rotateFallingPiece:

As noted, this method is similar to moveFallingPiece, in that it makes the rotation and then calls fallingPieceIsLegal (the same method used by moveFallingPiece) and undoes any illegal changes. As for the actual rotation, this is accomplished by changing the two- dimensional array of booleans that represent the falling piece. A new 2d array is created, and the old array is mapped to the new array according to a 90-degree counterclockwise rotation.

* + Updating keyPressed: We modify the keyPressed handler to call rotateFallingPiece in response to an up-arrow key press

# Before:

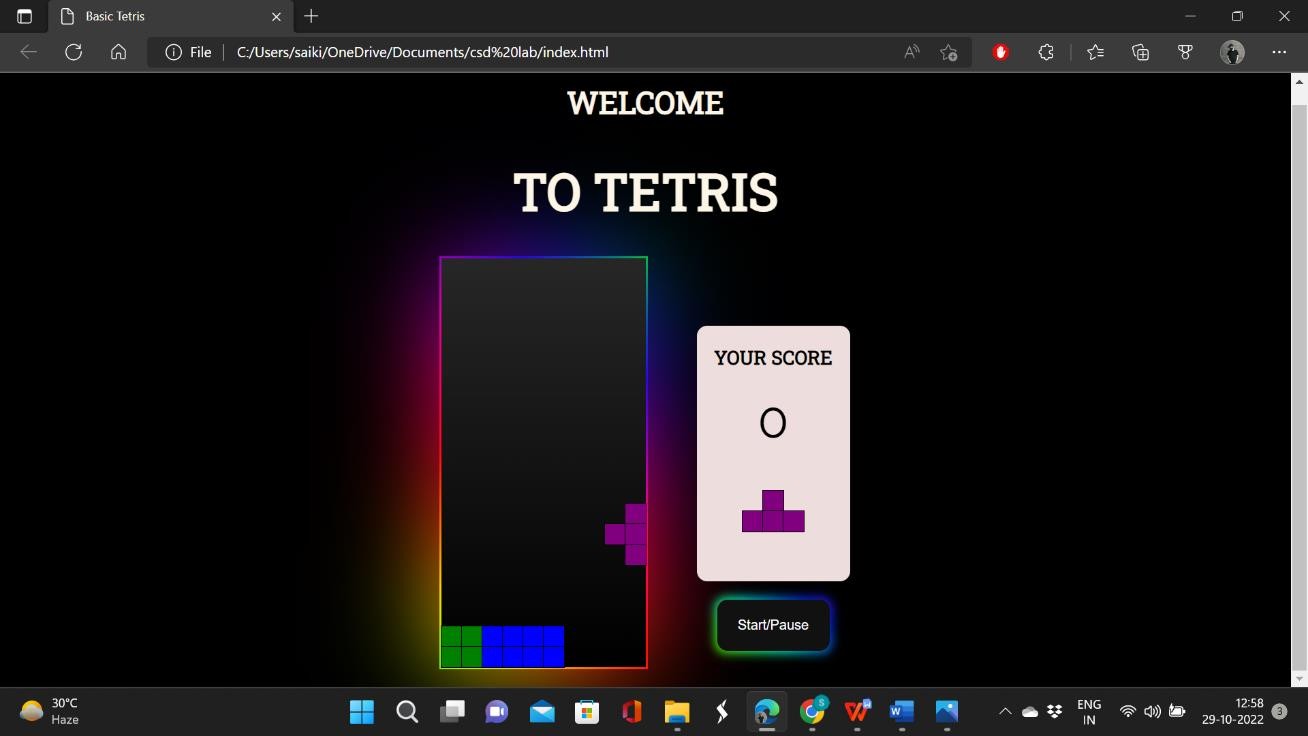
 

# After:



**Step 5: Dropping and Placing the falling Piece on the Board**

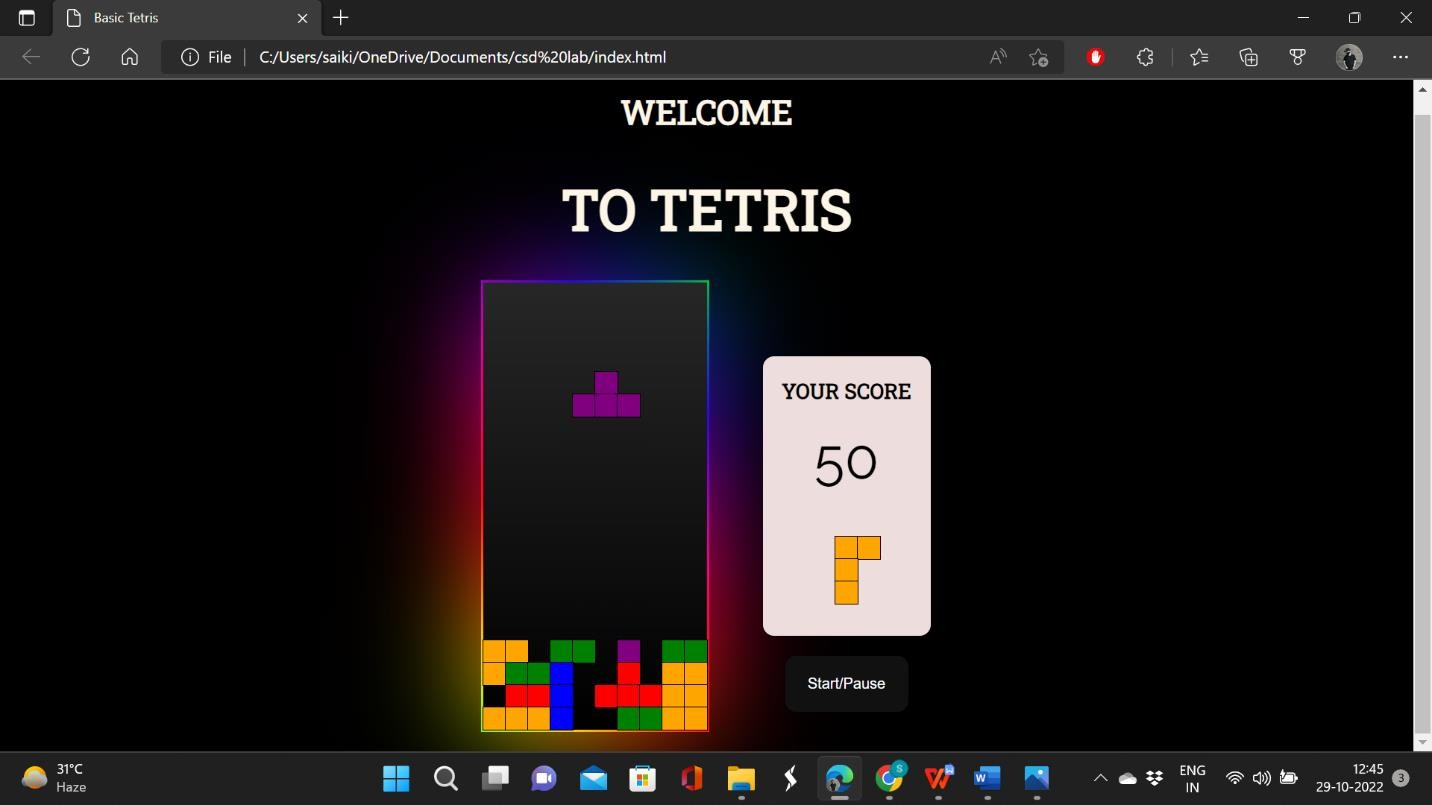
* + We are now ready to have the falling piece respond to timer events by dropping one row each time the timer fires.
  + A naive implementation would respond to timer events by calling moveFallingPiece(+1,0), just as we do in response to a down-arrow key press.
  + Actually, it's not a bad start, but this does point out two obvious problems. First, the piece is falling too fast. And second, when it gets to the bottom, it just sits there (until we hit a non-arrow key to artificially reset the falling piece).

 

# Step 6: Removing Full Rows and Keeping Score (and playing MIDI)

* + Our game is close to being complete -- or at least complete enough that it can be fun to play. All that remains is removing full rows (since filling rows is the object of Tetris) and keeping score.
  + Writing removeFullRows:

This method will iterate a variable, oldRow, from the *bottom to the top* of the board (that is, from this. rows-1 to 0). It will also start another variable, new Row, at the bottom of the board. Each oldRow that is not full is copied to the new Row, which is then decremented by one. Each oldRow that is full is not copied, and the full Rows counter is incremented. At the end, the rows on top are cleared (set to empty Color) as necessary

 

**Code:**

document.addEventListener('DOMContentLoaded', () => {

const grid = document.querySelector('.grid')

let squares = Array.from(document.querySelectorAll('.grid div'))

const scoreDisplay = document.querySelector('#score')

const startBtn = document.querySelector('#start-button')

const width = 10

let nextRandom = 0

let timerId

let score = 0

const colors = [

'orange',

'red',

'purple',

'green',

'blue'

]

//The Tetrominoes

const lTetromino = [

[1, width+1, width\*2+1, 2],

[width, width+1, width+2, width\*2+2],

[1, width+1, width\*2+1, width\*2],

[width, width\*2, width\*2+1, width\*2+2]

]

const zTetromino = [

[0,width,width+1,width\*2+1],

[width+1, width+2,width\*2,width\*2+1],

[0,width,width+1,width\*2+1],

[width+1, width+2,width\*2,width\*2+1]

]

const tTetromino = [

[1,width,width+1,width+2],

[1,width+1,width+2,width\*2+1],

[width,width+1,width+2,width\*2+1],

[1,width,width+1,width\*2+1]

]

const oTetromino = [

[0,1,width,width+1],

[0,1,width,width+1],

[0,1,width,width+1],

[0,1,width,width+1]

]

const iTetromino = [

[1,width+1,width\*2+1,width\*3+1],

[width,width+1,width+2,width+3],

[1,width+1,width\*2+1,width\*3+1],

[width,width+1,width+2,width+3]

]

const theTetrominoes = [lTetromino, zTetromino, tTetromino, oTetromino, iTetromino]

let currentPosition = 4

let currentRotation = 0

console.log(theTetrominoes[0][0])

//randomly select a Tetromino and its first rotation

let random = Math.floor(Math.random()\*theTetrominoes.length)

let current = theTetrominoes[random][currentRotation]

//draw the Tetromino

function draw() {

current.forEach(index => {

squares[currentPosition + index].classList.add('tetromino')

squares[currentPosition + index].style.backgroundColor = colors[random]

})

}

//undraw the Tetromino

function undraw() {

current.forEach(index => {

squares[currentPosition + index].classList.remove('tetromino')

squares[currentPosition + index].style.backgroundColor = ''

})

}

//assign functions to keyCodes

function control(e) {

if(e.keyCode === 37) {

moveLeft()

} else if (e.keyCode === 38) {

rotate()

} else if (e.keyCode === 39) {

moveRight()

} else if (e.keyCode === 40) {

moveDown()

}

}

document.addEventListener('keyup', control)

//move down function

function moveDown() {

undraw()

currentPosition += width

draw()

freeze()

}

//freeze function

function freeze() {

if(current.some(index => squares[currentPosition + index + width].classList.contains('taken'))) {

current.forEach(index => squares[currentPosition + index].classList.add('taken'))

//start a new tetromino falling

random = nextRandom

nextRandom = Math.floor(Math.random() \* theTetrominoes.length)

current = theTetrominoes[random][currentRotation]

currentPosition = 4

draw()

displayShape()

addScore()

gameOver()

}

}

//move the tetromino left, unless is at the edge or there is a blockage

function moveLeft() {

undraw()

const isAtLeftEdge = current.some(index => (currentPosition + index) % width === 0)

if(!isAtLeftEdge) currentPosition -=1

if(current.some(index => squares[currentPosition + index].classList.contains('taken'))) {

currentPosition +=1

}

draw()

}

//move the tetromino right, unless is at the edge or there is a blockage

function moveRight() {

undraw()

const isAtRightEdge = current.some(index => (currentPosition + index) % width === width -1)

if(!isAtRightEdge) currentPosition +=1

if(current.some(index => squares[currentPosition + index].classList.contains('taken'))) {

currentPosition -=1

}

draw()

}

//rotate the tetromino

function rotate() {

undraw()

currentRotation ++

if(currentRotation === current.length) { //if the current roation gets to 4, make it go back to 0

currentRotation = 0

}

current = theTetrominoes[random][currentRotation]

draw()

}

//show up-next tetromino in mini-grid display

const displaySquares = document.querySelectorAll('.mini-grid div')

const displayWidth = 4

const displayIndex = 0

//the Tetrominos without rotations

const upNextTetrominoes = [

[1, displayWidth+1, displayWidth\*2+1, 2], //lTetromino

[0, displayWidth, displayWidth+1, displayWidth\*2+1], //zTetromino

[1, displayWidth, displayWidth+1, displayWidth+2], //tTetromino

[0, 1, displayWidth, displayWidth+1], //oTetromino

[1, displayWidth+1, displayWidth\*2+1, displayWidth\*3+1] //iTetromino

]

//display the shape in the mini-grid display

function displayShape() {

//remove any trace of a tetromino form the entire grid

displaySquares.forEach(square => {

square.classList.remove('tetromino')

square.style.backgroundColor = ''

})

upNextTetrominoes[nextRandom].forEach( index => {

displaySquares[displayIndex + index].classList.add('tetromino')

displaySquares[displayIndex + index].style.backgroundColor = colors[nextRandom]

})

}

//add functionality to the button

startBtn.addEventListener('click', () => {

if (timerId) {

clearInterval(timerId)

timerId = null

} else {

draw()

timerId = setInterval(moveDown, 1000)

nextRandom = Math.floor(Math.random()\*theTetrominoes.length)

displayShape()

}

})

//add score

function addScore() {

for (let i = 0; i < 199; i +=width) {

const row = [i, i+1, i+2, i+3, i+4, i+5, i+6, i+7, i+8, i+9]

if(row.every(index => squares[index].classList.contains('taken'))) {

score +=10

scoreDisplay.innerHTML = score

row.forEach(index => {

squares[index].classList.remove('taken')

squares[index].classList.remove('tetromino')

squares[index].style.backgroundColor = ''

})

const squaresRemoved = squares.splice(i, width)

squares = squaresRemoved.concat(squares)

squares.forEach(cell => grid.appendChild(cell))

}

}

}

//game over

function gameOver() {

if(current.some(index => squares[currentPosition + index].classList.contains('taken'))) {

scoreDisplay.innerHTML = 'end'

clearInterval(timerId)

}}})

**References:**

* <https://www.w3schools.com/html/>
* <https://www.w3schools.com/css/>
* <https://www.w3schools.com/js/>
* <https://stackoverflow.com/>
* <https://www.tutorialspoint.com/css/index.htm>

# Drive Link: