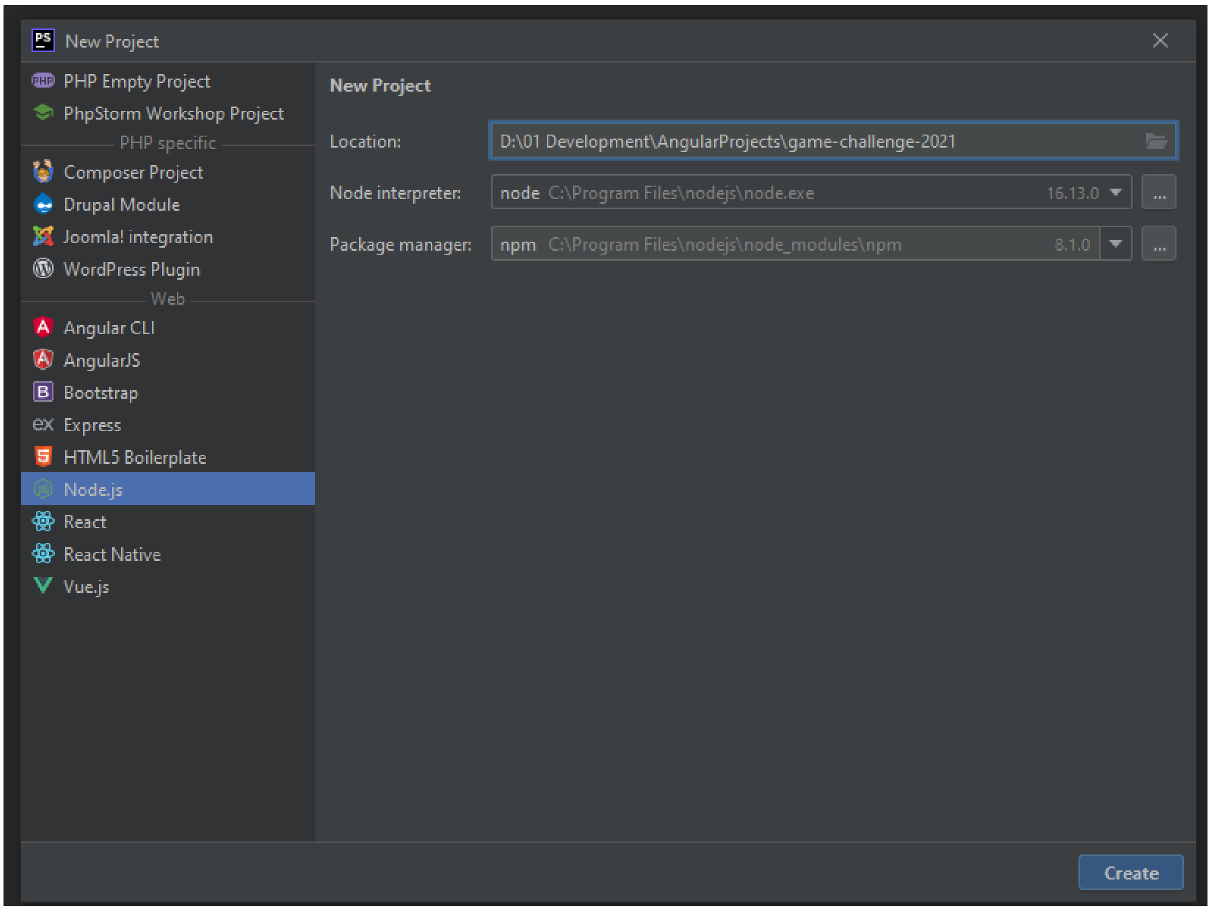
1. Install phpstorm
2. Install nodeJS
3. Create a new nodeJS project (in phpStorm



New Project

Location: D:\01 Development\AngularProjects\game-challenge-2021

Node interpreter: node C:\Program Files\nodejs\node.exe 16.13.0

Package manager: npm C:\Program Files\nodejs\node_modules\npm 8.1.0

4. Create a directory build
5. Create a directory src
6. Create tsconfig.json

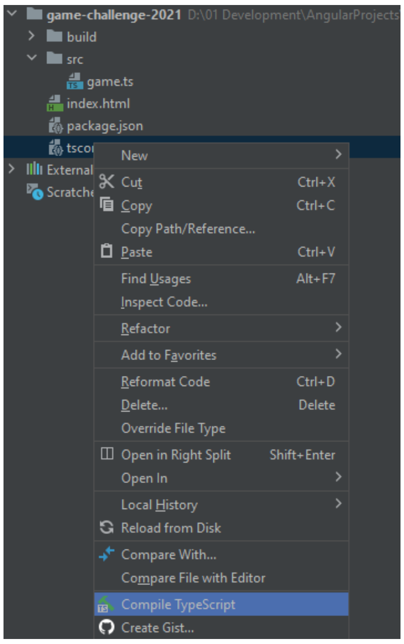
```
{
  "compilerOptions": {
    "target": "ES6",
    "sourceMap": true,
    "noImplicitAny": true,
    "noImplicitReturns": true,
    "removeComments": true,
    "out": "build/app.js",
    "noEmitOnError": true,
    "newLine": "lf"
  },
  "include": [
    "src/**/*"
  ],
  "exclude": [
    "node_modules",
    "**/*.spec.ts"
  ]
}
```

7. Create the src\game.ts main file

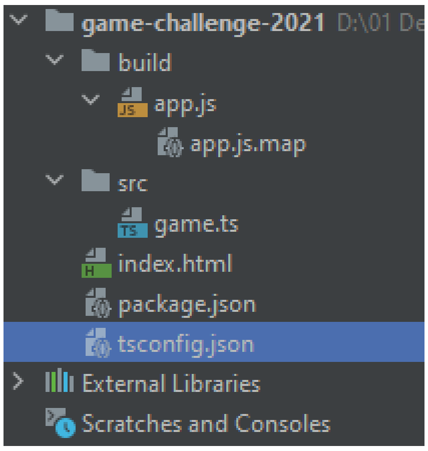
8. Create the index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link href="styles/default.css" rel="stylesheet" type="text/css">
    <title>My Awesome Game</title>
</head>

<body>
    <canvas id="canvas"></canvas>
    <script src="./build/app.js"></script>
</body>
</html>
```

9. Right mouse click tsconfig.json -> compile Typescript
10. Compile all typescript files



11. Check the output



Check output is in build. App.js.

```
//# sourceMappingURL=app.js.map
```

App.js.map

```
{"version":3,"file":"app.js","sourceRoot":"","sources":["../src/game.ts"],"names":[],"mappings":""}
```

Create the following

Create the file “src\base\ViewBase.ts”

```
abstract class ViewBase {

    protected readonly d_canvasHelper : CanvasHelper;

    protected constructor(canvasId: HTMLCanvasElement) {
        //construct all canvas
        this.d_canvasHelper = CanvasHelper.Instance(canvasId);
    }

    public abstract Render() : void;

}
```

Create the file “src\helpers\CanvasHelper.ts”

```
class CanvasHelper {

    private readonly d_canvas : HTMLCanvasElement;
    private readonly d_context : CanvasRenderingContext2D;

    private static s_instance: CanvasHelper = null;

    public static Instance(aCanvas: HTMLCanvasElement = null): CanvasHelper {
        if (this.s_instance == null)
        {
            if (aCanvas == null)
            {
                throw new DOMException("The first time the instance is created a Canvas must be given.");
            }
            this.s_instance = new CanvasHelper(aCanvas);
        }
        return this.s_instance;
    }

    private constructor(aCanvas: HTMLCanvasElement) {
        this.d_canvas = aCanvas;
        this.d_canvas.width = window.innerWidth; // add /2 to allow two games side-by-side horizontal
        this.d_canvas.height = window.innerHeight;// add /2 to allow two games side-by-side vertical

        this.d_context = this.d_canvas.getContext('2d');
    }

    public writeTextToCanvas(
        text : string,
        fontSize : number,
        xCoordinate : number,
        yCoordinate : number,
        color : string = "white",
        alignment : CanvasTextAlign = "center"
    )  {
        this.d_context.font = `${fontSize}px Tahoma`;
        this.d_context.fillStyle = color;
        this.d_context.textAlign = alignment;
        this.d_context.fillText(text,xCoordinate,yCoordinate);
    }

    public writeTextCenterToCanvas(
        text : string,
        fontSize : number,
        color : string = "white",
        alignment : CanvasTextAlign = "center"
    )  {
        const horizontalCenter = this.GetWidth() / 2;
        const verticalCenter = this.GetHeight() / 2;

        this.d_context.font = `${fontSize}px Tahoma`;
        this.d_context.fillStyle = color;
        this.d_context.textAlign = alignment;
        this.d_context.fillText(text,horizontalCenter,verticalCenter);
    }

    /**
     * Clear
     * @AccessModifier {public}
     * Clears the canvas
     */
    public Clear(): void {
        // clear the screen
        this.d_context.clearRect(0, 0, this.GetWidth(), this.GetHeight());
    }

    /**
     * GetHeight
     * @AccessModifier {public}
     * returns Height of the canvas
     */
    public GetHeight() : number {
        // return the height of the canvas
        return this.d_canvas.height;
    }

    /**
     * GetWidth
     * @AccessModifier {public}
     * returns the Width of the canvas
     */
    public GetWidth() : number {
        // return the width of the canvas
        return this.d_canvas.width;
    }


}
```

Create the file “src\views\StartScreen.ts”

```
class StartScreen extends ViewBase {

    public constructor(aCanvas: HTMLCanvasElement) {
        super(aCanvas);
    }

    public Render() : void
    {
        this.d_canvasHelper.Clear();
        this.d_canvasHelper.writeTextCenterToCanvas("StartScreen",40);
    }
}
```

create the file “src\views\SecondScreen.ts”

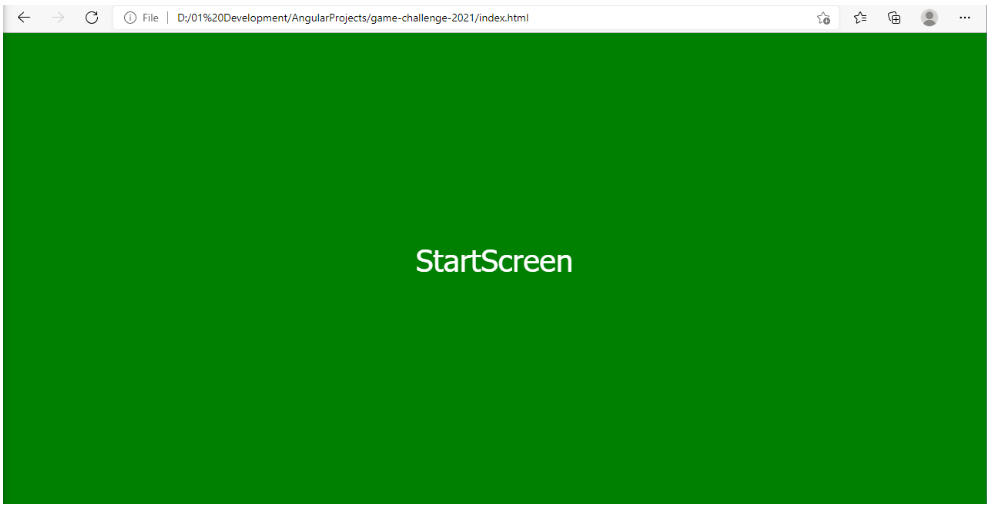
```
class SecondScreen extends ViewBase {

    public constructor(aCanvas: HTMLCanvasElement) {
        super(aCanvas);
    }

    public Render() : void
    {
        this.d_canvasHelper.Clear();
        this.d_canvasHelper.writeTextCenterToCanvas("SecondScreen",40);
    }
}
```

Create the file “styles/default.css”

```
*, body {
    margin: 0;
    padding: 0;
    overflow: hidden;
}

canvas {
    background-color: green;
    height: 100vh;
    width: 100vw;
}
```

Compile the tsconfig.json

Open the html in the browser


File
D:/01%20Development/AngularProjects/game-challenge-2021/index.html
StartScreen