

## Angular assignment- Basic Game Design

Following are all the source code files:

app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'myapp';
}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { SquareComponent } from './square/square.component';
import { BoardComponent } from './board/board.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { NbThemeModule, NbLayoutModule, NbButtonModule } from '@nebular/theme';
import { NbEvaIconsModule } from '@nebular/eva-icons';
import { ServiceWorkerModule } from '@angular/service-worker';
import { environment } from '../environments/environment';

@NgModule({
  declarations: [
    AppComponent,
    SquareComponent,
    BoardComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserAnimationsModule,
    NbThemeModule.forRoot({ name: 'cosmic' }),
    NbLayoutModule,
    NbEvaIconsModule,
    NbButtonModule,
    ServiceWorkerModule.register(environment.serviceWorkerPath)
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

```

    ServiceWorkerModule.register('ngsw-worker.js', { enabled:
environment.production })
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

## app.component.spec.ts

```

import { TestBed, async } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async(() => {
    TestBed.configureTestingModule({
      imports: [
        RouterTestingModule
      ],
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  }));

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'myapp'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.debugElement.componentInstance;
    expect(app.title).toEqual('myapp');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.debugElement.nativeElement;
    expect(compiled.querySelector('.content span').textContent).toContain('myapp
app is running!');
  });
});

```

## app.component.html

```
<nb-layout>

  <nb-layout-header fixed>
    <!-- Insert header here -->
  </nb-layout-header>

  <nb-layout-column>

    <app-board></app-board>

    <router-outlet></router-outlet>
  </nb-layout-column>

  <nb-layout-footer fixed>
    <!-- Insert footer here -->
  </nb-layout-footer>

</nb-layout>
```

## styles.css

```
@import 'themes';

@import '~@nebular/theme/styles/globals';

@include nb-install() {
  @include nb-theme-global();
};

/* You can add global styles to this file, and also import other style files */
```

## board.component.html

```
<h1>Current Player: {{ player }} </h1>

<button nbButton outline status="danger" (click)="newGame()">Start new
Game</button>

<h2 *ngIf="winner">
  Player {{ winner }} won the game!
</h2>
```

```

<main>
  <app-square
    *ngFor="let val of squares; let i = index"
    [value]="val"
    (click)="makeMove(i)">

  </app-square>
</main>

```

## board.component.scss

```

main {
  display: grid;
  grid-template-columns: 200px 200px 200px;
  grid-gap: 0px;
}

app-square {
  border: 1px gray solid;
  height: 200px;
}

```

## Board.component.ts

```

import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-board',
  templateUrl: './board.component.html',
  styleUrls: ['./board.component.scss']
})
export class BoardComponent implements OnInit {
  squares: string[];
  xIsNext: boolean;
  winner: string;

  constructor() {}

  ngOnInit() {
    this.newGame();
  }

  newGame() {
    this.squares = Array(9).fill(null);
    this.winner = null;
    this.xIsNext = true;
  }

  get player() {

```

```

    return this.xIsNext ? 'X' : 'O';
  }

  makeMove(idx: number) {
    if (!this.squares[idx]) {
      this.squares.splice(idx, 1, this.player);
      this.xIsNext = !this.xIsNext;
    }

    this.winner = this.calculateWinner();
  }

  calculateWinner() {
    const lines = [
      [0, 1, 2],
      [3, 4, 5],
      [6, 7, 8],
      [0, 3, 6],
      [1, 4, 7],
      [2, 5, 8],
      [0, 4, 8],
      [2, 4, 6]
    ];
    for (let i = 0; i < lines.length; i++) {
      const [a, b, c] = lines[i];
      if (
        this.squares[a] &&
        this.squares[a] === this.squares[b] &&
        this.squares[a] === this.squares[c]
      ) {
        return this.squares[a];
      }
    }
    return null;
  }
}

```

Output:

