<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>My Snake Game</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

<body>

<ngx-snake>Loading...</ngx-snake>

</body>

</html>

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { enableProdMode } from '@angular/core';

import { AppModule } from './app/app.module';

enableProdMode();

platformBrowserDynamic().bootstrapModule(AppModule);

<div class="game-container">

<div class="game-header">

<h3 class="logo">My Snake Game</h3>

<div class="start-button" [ngClass]="{'disable-clicks': gameStarted}" (click)="showMenu()">Start Game</div>

</div>

<div class="row" \*ngFor="let column of board; let i = index;">

<div class="column" [ngStyle]="{'background-color': setColors(i, j)}" \*ngFor="let row of column; let j = index"></div>

</div>

<div class="new-game-menu" \*ngIf="showMenuChecker">

<div class="start-button new-game-button" (click)="newGame(mode)" \*ngFor="let mode of getKeys(all\_modes)">{{all\_modes[mode]}}</div>

</div>

</div>

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

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import { BestScoreManager } from './app.storage.service';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule

],

providers: [

BestScoreManager

],

bootstrap: [

AppComponent

]

})

export class AppModule { }

.game-header {

color: #fff;

padding: 5px 15px 5px 0px;

position: relative;

}

.game-header>.score-block {

display: inline-block;

}

.score-block>.score {

position: absolute;

right: 8px;

top: -5px;

}

.score-block>.best-score {

position: absolute;

right: 8px;

margin-top: 5px;

font-size: 12px;

}

.game-header>.logo {

display: inline-block;

padding-left: 15px;

}

.game-container {

min-height: 100vh;

height: auto;

margin: 0;

}

.row {

height: 26px;

}

.column {

border: 1px solid rgba(97, 131, 138, .1);

width: 24px;

height: 24px;

display: inline-block;

}

.start-button {

padding: 15px;

text-align: center;

background-color: #47565A;

color: white;

border-radius: 5px;

}

.start-button:hover {

opacity: 0.65;

cursor: pointer;

}

.start-button.new-game-button {

margin: 0 105px 4px 105px;

}

.disable-clicks {

pointer-events: none;

}

.new-game-menu {

position: absolute;

top: 0;

width: 100%;

bottom: 0;

padding: 40% 0;

text-align: center;

background: rgba(0, 0, 0, 0.15);

}

.new-game-menu-label {

font-size: 1.17em;

margin-bottom: 13px;

display: block;

color: #fff;

}

.new-best-score {

animation: glow .5s infinite alternate;

}

@keyframes glow {

to {

text-shadow: 0 0 15px #fff000;

}

}

@media screen and (max-width: 480px) {

.game-container {

width: 100%;

}

.column {

width: 5.55%;

height: 0;

padding-bottom: 5.06%;

box-sizing: border-box;

}

.row {

display: -webkit-box;

display: -moz-box;

display: -ms-flexbox;

display: -webkit-flex;

display: flex;

height: 0;

padding-bottom: 5.5%;

}

.new-game-menu {

width: 100%;

top: 8px;

padding-top: 30%;

font-size: 80%;

}

.start-button.new-game-button {

margin: 0 15% 1% 15%;

}

@-moz-document url-prefix() {

.column {

min-height: 5.5vw;

}

}

}

image{

height:45em; background-size:cover; width:100%;

background-position:50% 50%;

}

.jumbotron {

margin-bottom: 0px;

background-position: 0% 25%;

background-size: cover;

background-repeat: repeat-x;

color: white;

text-shadow: black 0.3em 0.3em 0.3em;

}

<!--The content below is only a placeholder and can be replaced.-->

<title>TwitterSearch</title>

<meta charset="utf-8">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://d3js.org/d3.v3.min.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.1.1/angular.min.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<body class="image">

<div class="jumbotron" align="center">

<h1 >Google Search</h1>

</div>

<div class="col-lg-12 " align="center">

<input placeholder="Enter a text" type="text" class="form-control" #text/>

<br>

<button class="btn btn-lg btn-success btn-block" (click)=search()>Search</button>

<br>

</div>

<p><h1><a><{{ url }}><</b></h1>

</body>

<router-outlet></router-outlet>

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 'google-search'`, () => {

const fixture = TestBed.createComponent(AppComponent);

const app = fixture.debugElement.componentInstance;

expect(app.title).toEqual('google-search');

});

it('should render title in a h1 tag', () => {

const fixture = TestBed.createComponent(AppComponent);

fixture.detectChanges();

const compiled = fixture.debugElement.nativeElement;

expect(compiled.querySelector('h1').textContent).toContain('Welcome to google-search!');

});

});

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

import { NgModule } from '@angular/core';

import { HttpClientModule} from '@angular/common/http';

import { FormsModule} from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }