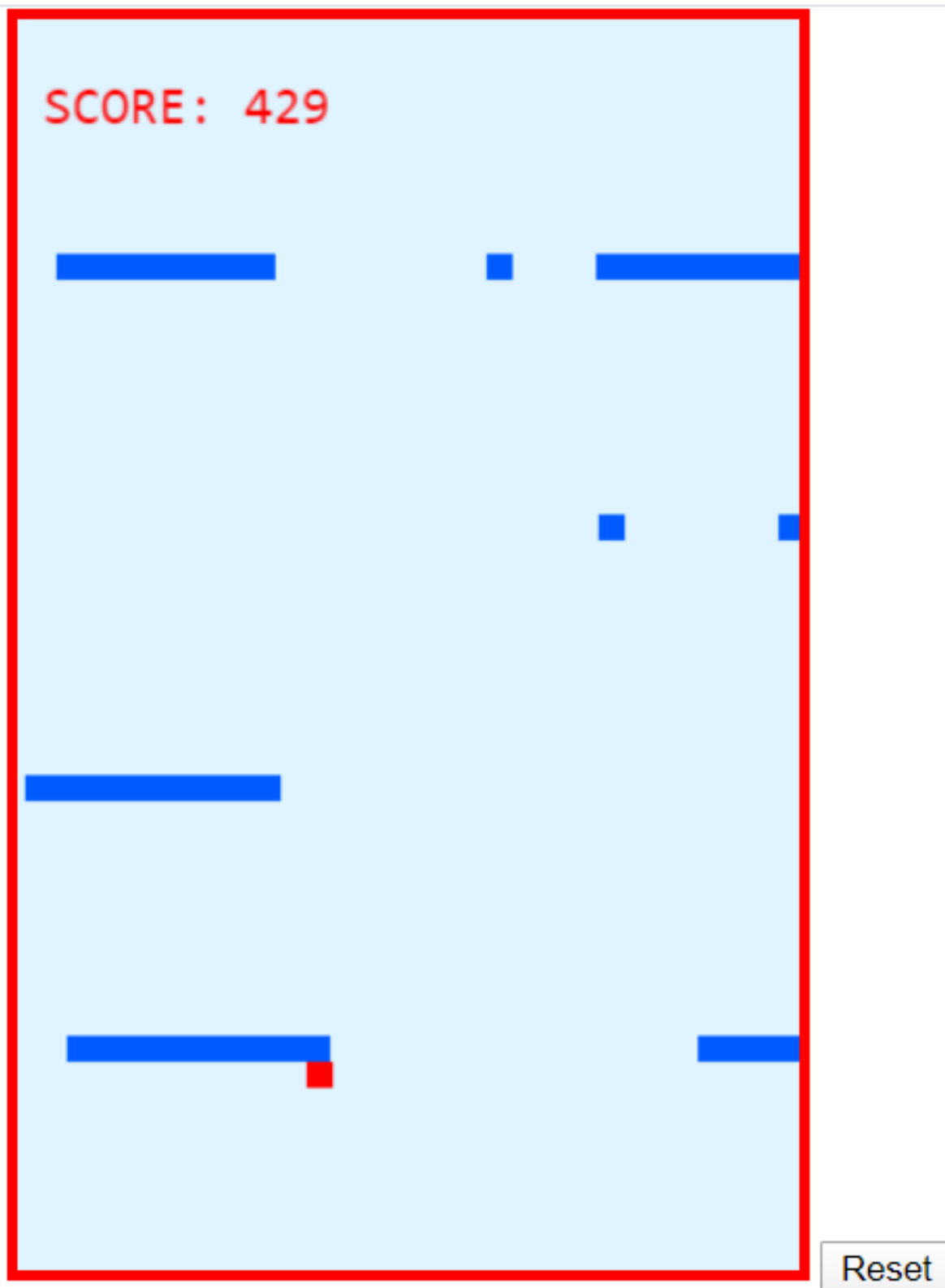


DOKUMENTACIJA PROJEKTA



PROJEKT: MiniPiccino

GRUPA: JAVASCRIPT

Dora Baričević,

Lara Smolić-Ročak,

Paula Šainčić,

Ana Tirić,

Antonio Tokić,

Rene Zbašnik - predstavnik

GAM,

Rijeka, 8.5.2020.

SADRŽAJ

1. <u>UVOD</u>	4
2. <u>JAVASCRIPT</u>	5
3. <u>DETALJAN OPIS RADA</u>	6
4. <u>TEHNIČKE INFORMACIJE</u>	11

1. UVOD

IDEJA PROJEKTA I AUTORI

Naša je grupa koja se sastoji od šest članova (Dore, Lare, Paule, Ane, Antonija i Renea) za zadatak imala napraviti rad pomoću programskog jezika JavaScript. Za inspiraciju nam je poslužila igrica „Flappy Bird“ koja je 2013. pokorila svijet. Za razliku od njezinog horizontalnog dizajna našu igricu odlikuje vertikalni i to s obrnutom gravitacijom. Osim toga u našoj igrici ulogu ptice preuzima kvadrat.

Naime, cilj je igrača izbjegavati prepreke te na taj način sakupljati bodove kako bi u konačnici ostvario što bolji rezultat i time dokazao da je upravo njegov kvadrat onaj najbolji. Rad je na igrici započeo Rene napravivši „kostur“ programa čime je dao odličnu bazu Dori i Ani da nastave s radom na samom programu i oblikuju njegov minimalistički, ali nadasve očaravajući dizajn. Uloga Antonija, Lare i Paule ostvarila se u konzultaciji s ostatkom tima oko samog programa te završnim detaljima poput izrade dokumentacije.

2. JAVASCRIPT

O PROGRAMU, DETALJI

JavaScript skriptni je programski jezik koji se izvršava u web-pregledniku. Skriptni jezici su programski jezici manjih mogućnosti koji se sastoje od izvršnog računalnog koda (obično ugrađenog u HTML stranice). Najpopularniji je skriptni programski jezik na Internetu i podržavaju ga svi poznati preglednici (Chrome, Mozilla, Opera, Internet Explorer, Netscape). Iako napravljen da, zbog lakšeg korištenja, bude sličan Javi, nije objektno orijentiran kao Java. JavaScript temelji se na prototipu zbog čega nestaje svaka povezanost s programskim jezikom Java. JS izvorno je razvijen od strane tvrtke Netscape (www.netscape.com). Primjena je ECMAScript standarda. JavaScript s Asynchronous JavaScript and XML (AJAX) tehnikom web-stranicama omogućuje komunikaciju sa serverskim programom, što ga čini interaktivnijom i nadasve lakšom za korištenje. JavaScript je interpreter, što znači da se skripta izvršava odmah naredbu po naredbu, bez prethodnog prevođenja (kompiliranja) cijelog programa i kreiranja izvršne datoteke.

JavaScript omogućava programiranje u okviru HTML stranica, pretvaranje dinamičkog teksta u HTML stranicu (ako se neka varijabla pod nazivom "name" želi ispisati u okviru HTML stranice, može se napisati skriptna naredba), reagiranje na događaje – moguće je postaviti da se skripta izvršava kada se dogodi neki događaj (kada se stranica učitava ili kada korisnik klikne na određeno dugme ili drugi HTML element), čitanje i pisanje HTML elemenata – JavaScript može pročitati i promijeniti sadržaj nekog HTML elementa, validiranje (provjeru ispravnosti i vjerodostojnosti) podataka – JavaScript može validirati podatke prije nego se pošalju na server, čime se server oslobađa dodatne obrade, detektiranje preglednika kojeg korisnik upotrebljava (na osnovu prepoznavanja preglednika JavaScript može učitati drugačiju stranicu ovisno o pregledniku tako da se učitava stranica koja je posebno dizajnirana za isti), kreiranje "kolačića" (cookies) – JavaScript može pohraniti i učitati informacije o korisnikovom računalu i još mnogo toga.

3. DETALJAN OPIS RADA

NAMJENA, FUNKCIJA I OSOBINE PROJEKTA

Namjena našeg projekta bila je prvenstvo za zabavu i rekreaciju. Kako je u uvodu navedeno, inspiracija je došla od igrice „Flappy Bird“ koja je kasnije bila maknuta s tržišta. Htjeli smo napraviti rekreaciju i pružiti korisnicima koliko toliko slično iskustvo. Kasnije smo, igrajući samu igricu, vidjeli da ona ne mora biti samo za zabavu; vrlo dobro može poslužiti i za relaksaciju i rješavanja stresa, a zbog svoje jednostavnosti je dostupna svima; od onih najmanjih, do onih najvećih. Još jedna odlika naše igrice je da je ona prilagođena svim osobama bez obzira na dob i vrlo lagana za igranje.

Igrica funkcionira na način da smo mi crveni kvadrat i moramo izbjegavati prepreke koje nam dolaze ususret. Krećemo se pomoću tipkovnice iako je prvotna ideja bila pokretanje pomoću „button“-a LIJEVO/DESNO, ali smo kasnije shvatili da bi ovako bilo lakše. Kraj igrice je kada naš kvadrat ne uspije izbjeći prepreku te se zabije u nju. Pa krenimo s objašnjavanjem koda.

Kako bismo uopće započeli s radom, trebali smo definirati program, napraviti kostur (uvesti html komponente) te smo kao dodatak uredili naš okvir igrice (dodali smo crveni obrub i plavu kao boju pozadine). Također smo dodali i gumb kojim ćemo kasnije moći ponovo pokrenuti igricu.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
5    <style>
6      canvas {
7        border:4px solid red;
8        background-color: #DFF4FF;
9      }
10   </style>
11   </head>
12   <body onload="startGame()">
13     <button onclick="myButton()"> Reset </button>
14
15   <script>
16
```

Za početak smo trebali definirati varijable koje označavaju naš rezultat, prepreke i naš kvadratić.

```
22  var myGamePiece;
23  var myObstacles = [];
24  var myScore;
25
```

Nakon toga smo krenuli s uređivanjem svake od varijabli; prvo smo definirali kvadrat (boja, veličina) i smjestili score u okvir igrice.

```
26 function startGame() {
27     myGamePiece = new component(10, 10, "red", 150, 400); // oblikujemo predmet
28     myScore = new component("20px", "Consolas", "red", 10, 40, "text"); // oblikujemo gdje stoji rezultat
29     myGameArea.start();
30 }
31
```

Kada smo to sve odredili, krećemo sa stvaranjem i uređivanjem okvira u kojem će se igrati igrice (određujemo širinu, visinu i značajke okvira). Također dodajemo i funkciju clear kojom brišemo cijeli canvas (kako se krećemo, želimo da se ono što smo uspješno savladali makne).

```
32 var myGameArea = { // započinjemo postavljanjem prozora gdje će se sve odvijati
33     canvas : document.createElement("canvas"),
34     start : function() {
35         this.canvas.width = 300;
36         this.canvas.height = 480;
37         this.context = this.canvas.getContext("2d");
38         document.body.insertBefore(this.canvas, document.body.childNodes[0]);
39         this.frameNo = 0;
40         this.interval = setInterval(updateGameArea, 15);
41     },
42     clear : function() {
43         this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
44     }
45 }
```

Sljedeći dio koda bio je urediti gravitaciju – da se naš kvadrat giba prema gore cijelo vrijeme (obrnuto od normalne gravitacije).

```
47 function component(width, height, color, x, y, type) { //funkcija gdje određujemo gravitaciju u prozoru, početni score i boje
48     this.type = type;
49     this.score = 0;
50     this.width = width;
51     this.height = height;
52     this.speedX = 0;
53     this.speedY = 0;
54     this.x = x;
55     this.y = y;
56     this.update = function() {
57         ctx = myGameArea.context;
58         if (this.type == "text") {
59             ctx.font = this.width + " " + this.height;
60             ctx.fillStyle = color;
61             ctx.fillText(this.text, this.x, this.y);
62         } else {
63             ctx.fillStyle = color;
64             ctx.fillRect(this.x, this.y, this.width, this.height);
65         }
66     }
67 }
```

Nakon toga dolazimo do problema kako će se naš kvadrat kretati (bez da ostavlja trag iza sebe). To smo riješili na sljedeći način:

```
67     this.newPos = function() { //mjenjanje pozicije u prozoru
68         this.x += this.speedX;
69         this.y += this.speedY;
70     }
```

Također se pojavio problem kako napraviti da se naš kvadrat zabije u prepreku, tj. da se prestane kretati kad dodirne prepreku, a ne da samo preleti preko nje.

```
71     this.crashWith = function(otherobj) { // kad se zabijemo u predmet (prestaje se kretati)
72         var myleft = this.x;
73         var myright = this.x + (this.width);
74         var mytop = this.y;
75         var mybottom = this.y + (this.height);
76         var otherleft = otherobj.x;
77         var otherright = otherobj.x + (otherobj.width);
78         var othertop = otherobj.y;
79         var otherbottom = otherobj.y + (otherobj.height);
80         var crash = true;
81         if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft) || (myleft > otherright)) {
82             crash = false;
83         }
84         return crash;
85     }
86 }
87
88
```

Nakon što smo definirali prepreke, kretanje kvadrata i sam sudar, moramo postaviti naš kvadrat u labirint s preprekama.

```
90     function updateGameArea() { //funkcija za postavljanje predmeta u labirintu,
91         var x, height, gap, minHeight, maxHeight, minGap, maxGap;
92         for (i = 0; i < myObstacles.length; i += 1) {
93             if (myGamePiece.crashWith(myObstacles[i])) {
94                 return ;
95             }
96         }
```

Ostaje nam još definirati prepreke, točnije njihovu učestalost, kretanje i boju. Stavili smo da se pojavljuju na „random“ kako bi igrice bila nepredvidljiva.


```

97     myGameArea.clear(); // oblikujemo prepreke, učestalost, boju; na random
98     myGameArea.frameNo += 1;
99     if (myGameArea.frameNo == 1 || everyinterval(100)) {
100         x1 = Math.floor(Math.random()*100);
101         x2 = Math.floor(Math.random()*100);
102         x3 = Math.floor(Math.random()*100);
103         x11 = Math.floor(Math.random()*300);
104         x21 = Math.floor(Math.random()*300);
105         x31 = Math.floor(Math.random()*300);
106         gap = 60
107         myObstacles.push(new component(x1, 10, "#005BFF", x11, gap));
108         myObstacles.push(new component(x2, 10, "#005BFF", x21, gap));
109         myObstacles.push(new component(x3, 10, "#005BFF", x31, gap));
110     }
111

```

Sljedeći dio koda govori o brzini i tome da se prepreke stalno pojavljuju.

```

112     for (i = 0; i < myObstacles.length; i += 1) {
113         myObstacles[i].y += 1;
114         myObstacles[i].update();
115     }

```

Ostalo nam je još kako namjestiti da se score ažurira svako malo, tj. da se on povećava što mi duže igramo.

```

116     myScore.text="SCORE: " + myGameArea.frameNo; //ažuriranje scorea
117     myScore.update();
118     myGamePiece.newPos();
119     myGamePiece.update();
120 }

```

Ovaj dio nam je bitan jer omogućuje pojavljivanje prepreka nakon određenog intervala. Onemogućava da se dogodi previše prepreka. (dijeli se frame kroz broj intervala).

```

121
122     function everyinterval(n) {
123         if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
124         return false;
125     }

```

Ostalo je još napraviti kretanje samog kvadrata. Prije smo mu omogućili mijenjanje pozicije, a sada moramo napraviti da se naš kvadratić pomakne lijevo/desno za određenu duljinu. I to ćemo sve na kraju napraviti tako da možemo upravljati tipkovnicom.

```
126
127 function desno () { //funkcija za kretanje udesno
128     myGamePiece.x +=3;
129 }
130
131 function lijevo(){ //funkcija za kretanje ulijevo
132     myGamePiece.x -=3;
133 }
134
135 document.onkeydown = function(event) { //funkcija da se koriste tipke na tipkovnici za kretanje
136     event = event || window.event;
137     switch (event.keyCode) {
138         case 37:
139             lijevo();
140             break;
141         case 39:
142             desno();
143             break;
144     }
145 }
146
147 </script>
148
149 </body>
150
```

Te nam je za kraj još ostalo napraviti da se naša igrice resetira/ponovo pokrene pritiskom na gumb „Reset“ koji smo odmah na početku napravili.

```
17
18 function myButton() { //resetira igricu
19     location.reload();
20 }
21
```

I to bi ukratko bilo to.

4. TEHNIČKE INFORMACIJE

KORIŠTENE TEHNOLOGIJE, POTREBNA KONFIGURACIJSKA STRUKTURA

Za izradu ove igrice koristili smo se programskim jezikom JavaScript koji je najpopularniji skriptni jezik na Internetu i kojeg podržavaju svi poznatiji preglednici kao npr. Opera, Firefox, Mozilla, Internet Explorer itd. JavaScript nam omogućuje programiranje u HTML-u (uvodimo oznaku `<script>`). Razlog zbog kojeg se ova naša igrica uspješno pokreće u raznim preglednicima je upravo u JS, naime, način na koji JavaScript funkcionira je zanimljiv; unutar normalne web stranice stavite neki JS kod, kada preglednik učitava stranicu pokreće se naš kod jer preglednik ima ugrađeni interpreter koji čita JS kod koji pronalazi na stranici i pokreće ga. Što se tiče naše igrice, nismo mogli dobiti bolji programski jezik jer je JS poznata za uporabu u interaktivne svrhe (u što spada i izrada igrice).

Igrica je napravljena na način da smo mi crveni kvadrat i krećemo se između prepreka pomoću tipkovnice. Cijelo njezino igranje se temelji na HTML i JS kodu.

Za pokretanje igrice nema baš mnogo preduvjeta osim pristupa Microsoft Windows 7 operativnom sustavu ili novijem. Nama je za izradu same bio potreban Notepad++ koji smo mogli besplatno instalirati, a čija je svrha editiranje programskog koda (s podrškom za četrdesetak jezika). Korištena je 7.6.6. verzija Notepad++ iako je mogla i novija, a i nešto starija.