

Igra 3

IGRA VESALO

U ovoj igri igrač pogadja rec koju mu je racunar zadao iz random liste reci koju korisnik moze da azurira. Moze da pogresi 5 puta pri pogadjanju date reci.

U ovoj verziji ne crtamo hangmana.

Po zavrsetku igre se ispisuje pobeda ili poraz, kao i trazena tajna rec.

Igra moze da se reloaduje na dugme nova igra kao i u prethodnim igricama.



POSTUPAK IMPLEMENTACIJE:

(Igrica se pravi 3 casa)

DEO 1

- **Osnovna HTML struktura stranice**
- **Uvezati sa ostatkom projekta i napraviti neophodne linkove sa drugih strana**
- **CSS i JS fajlovi da su uvezani (defer za js)**
- **Dohvatanje queri selektorima sve sto nam treba**

DEO 2

- Dinamicko pravljenje slova
- Init funkcija
- Pogadjanje reci
- Pravljenje reci za prikaz

DEO 3

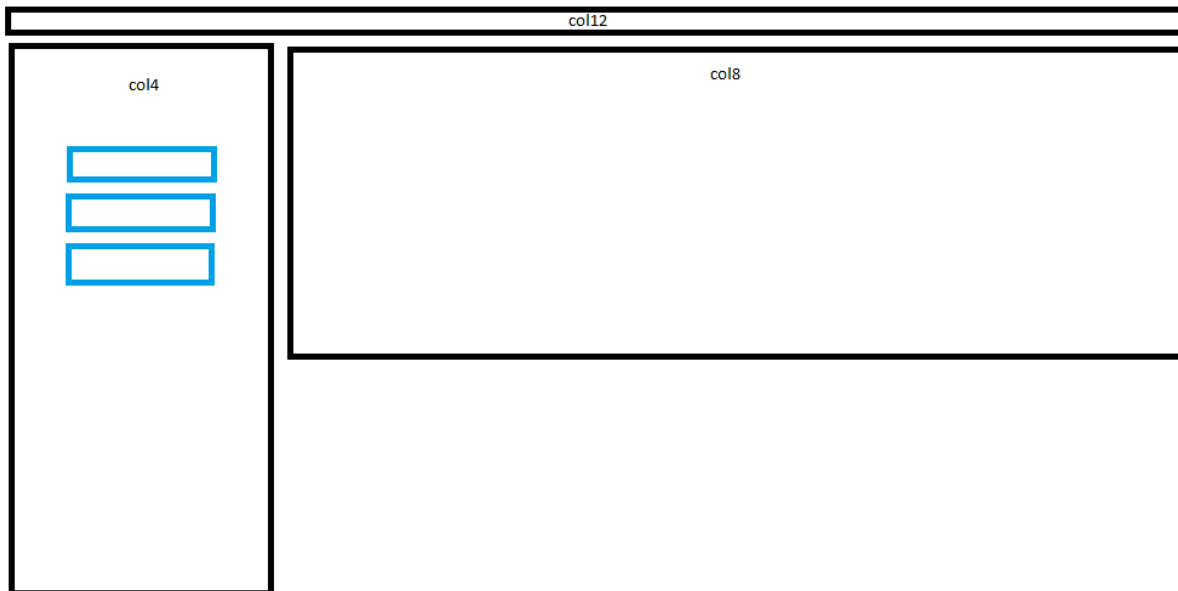
- Detekcija pogotka
- Notifikacije za pobedu i poraz
- Brojanje poena
- Bonus (da se sa strane prikazuju i slike hangmana - moze da im se da ideja pa da sami)

DEO 1 MPLEMENTACIJA:

HTML fajl

Korak 1: DODATAK U TRENUTNI PROJEKAT

- Iskopirati strukturu sa neke prethodne stranice
- Potrebno je dodati div koji ce imati col8 klasu kao sadrzaj
- Dodati da se ta stranica vidi na ostalim stranicama koje imaju igrice



Korak 2: DODATI OSNOVNI SETAP STRANICE

- Dodati title stranice da bude Hangman game
- Napraviti js fajl i dodati ga u script tag
- Script fajl mora biti defer `<script defer src="script.js"></script>`

Korak 3 - HTML STUKTURA ZA IGRU

```

<div class="col-8 sadrzaj">
  <div class="igra">
    <h1>Igra Vesalo</h1>
    <button id="igrajIspocetka" onclick="novaIgra()">Nova
Igra</button>
    <h3>Izaberi slovo</h3>
    <div class="slovo-div"></div>
    <div class="rec-div"></div>
    <h3>Preostalo pokusaja: <span class="zivoti">5</span></h3>
  </div>
  <div class="poruka hidden">
    <h2>KRAJ IGRE</h2>
    <h3 class="poruka-content">Pobedio si!</h3>
    <h4>Tacna Rec: <span class="poruka-span"></span></h4>
  </div>
</div>

```

Korak 4 - STIL (div za igru, dugmici za slovca i njegova stanja)

```

.igra {
  width: fit-content;
  margin: auto;
}

```

```

.slovce {
  width: 2rem;
  height: 2rem;
  background: #ffe5b4;
  border: none;
  border-radius: 4px;
  margin: 3px;
  cursor: pointer;
}
.slovce:hover {
  background-color: #e75480;
  border: none;
}
.disabled {
  pointer-events: none;
  background-color: darkgray;
}

```

OBJASNJENJA:

- Nisu radili ovo iz CSSa fit-content
- Nisu radili ove rem velicine, mi sve koristimo piksele, pa mozemo to proizvoljno
- Nisu radili pointer-events do sada

Korak 4 - JS queries

```

const slovaDiv = document.querySelector('.slovo-div');
const zivotiSpan = document.querySelector('.zivoti');
const recDiv = document.querySelector('.rec-div');
const porukaDiv = document.querySelector('.poruka');
const porukaSadrzaj = document.querySelector('.poruka-content');
const porukaTacnaRec = document.querySelector('.poruka-span');

```

OBJASNJENJA:

- **NOVO** - Dohvatanje se vrši preko klase, zato ide sa tackicom (.slovo-div), radjeno je preko data- atributa
- Objasnjenje za sta koji sluzi:
 - slovaDiv cemo koristiti da bismo dinamički u njega ubacili dugmice slova
 - zivotiSpan cemo koristiti za display zivota
 - recDiv cemo koristiti za div u kome treba dinamički prikazati zadatu rec i stalno je azurirati kada se neko slovo pogodi
 - porukaDiv cemo koristi za prikaz i skrivanje diva koji ce nam sluziti za ispis poruke po zavrsetku igre
 - PorukaSadrzaj moze biti da je korisnik pobedio ili izgubio
 - porukaTacnaRec prikazuje trenutnu rec koju smo pogadjali

DEO 2 MPLEMENTACIJA:

Korak 1: LISTA RECI ZA BIRANJE I FUNKCIJA ZA BIRANJE

```
const listaReci=['vrat', 'stalak', 'trn', 'žaba', 'otorinolaringolog' ]

// fja za dohvatanje random reci
function randomRec(list) {
  return list[Math.floor(Math.random() * listaReci.length)];
};
```

OBJASNJENJE:

- Tu je sve poznato za random, to bi mogli sami da naprave

Korak 2: INIT FUNKCIJA I FUNKCIONALNOSTI U NJOJ

DEO 1: sta sve treba da radi init()

```
function init() {
  recDiv.innerHTML = ' ';
  // pravljenje dugmica

  // podesavamo rec i zivote

  // dohvatamo queri za sva slova

  // popunjavamo koliko imamo zivota

  // pravimo crtice za tajnu rec i prikazujemo ih

};

// Pozivamo init() funkciju
init();
```

DEO 2: pravljanje dugmica unutar te funkcije

```
// pravljenje dugmica
var tekst = 'abcčćddefghijklmnoprsštuvzž'
for (var i of tekst) {
  const html = `<button
class="slovce">${i.toUpperCase()}</button>`;
  slovaDiv.insertAdjacentHTML('beforeend', html);
}
```

OBJASNJENJE:

- Objasniti insertAdjacent i parametre
https://www.w3schools.com/jsref/met_node_insertadjacenthtml.asp
- Insertovanje stringa u string sa \$ znakom

DEO 3: pravljanje tajne reci i zivota

```
// van init fje
var tajnaRec;
var zivoti;

// podesavamo rec i zivote u init funkciji
tajnaRec = randomRec(listaReci);
zivoti = 5;
```

OBJASNJENJE:

- Zgodnije je u initu da se podese svi ovi bitni parametri nego van inita

DEO 4: pravljenje crtica

```
var slova;

function init() {
  recDiv.innerHTML = '';
  // pravljenje dugmica
  ...
  // dohvatamo queri za sva slova
  slova = document.querySelectorAll('.slovce');

  // popunjavamo koliko imamo zivota
  zivotiSpan.textContent = zivoti;

  // pravimo crtice za tajnu rec
  for (let i = 0; i < tajnaRec.length; i++) {
    const html = `<span class="rec">_</span>`;
    recDiv.insertAdjacentHTML('beforeend', html);
  }
};
// initializing the page
init();
```

OBJASNJENJE:

- U P tagu zeza sto idu svi u novi red, pa SPAN bolje radi

DEO 5: Dodati CSS za crtice

```
.rec {
  width: 20px;
  margin: 5px;
}
```

Korak 3: Osluskivanje klika dugmica

```
// Osluskivanje klika dugmica
slova.forEach(btn => {
  btn.addEventListener('click', slovoKliknuto);
});

function slovoKliknuto(){
  const slovo = this.textContent.toLowerCase();
  // ispisivati za sada slovo koje je kliknuto
  console.log(slovo)

  // ako je slovo u reci, azurirati testContent od svakog spana

  // ako slovo nije u reci, smanjiti zivote
}
```

OBJASNJENJE:

- Property `textContent` je novo

DEO 3 MPLEMENTACIJA:

Korak 1: Funkcija za proveru pobede - kao u vesalu iz Pythona, da nema `_` u reci

```
// proverava da li je negde ostala _
const proveriRec = function () {
  let kraj = true;
  for (let i = 0; i < recDiv.children.length; i++) {
    if (recDiv.children[i].textContent === '_') {
      kraj = false;
    }
  }
  return kraj;
};
```

Korak 2: Smanjenje zivota i detekcija kraja igre

```
function smanjiZivote() {
  zivoti--;
  console.log(zivoti)
  zivotiSpan.textContent = zivoti;
}
```

```

    if (zivoti === 0) {
        prikaziPoruku("Izgubio si");
    }
}

function prikaziPoruku(txt) {
    porukaDiv.classList.remove('hidden');
    porukaTacnaRec.textContent = tajnaRec;
    porukaSadrzaj.textContent = txt;
}

```

Korak 2a : Dodati css za hidden

```

.rec {
    width: 20px;
    margin: 5px;
}

.hidden {
    display: none;
}

</style>

```

Korak 3 : dopuna funkcije od proslog puta - slovo kliknuto

Pomocna funkcija:

```

// provera indeksa koja su slova pogodjena datim unosom
function sviIndeksi(slovo) {
    let svi = [];
    [...tajnaRec].forEach((val, i) => { // operator ... objasniti
        if (val === slovo) {
            const index = i;
            svi.push(index);
        }
    });
    // console.log(svi);
    return svi;
};

```

// poziv te funkcije u funkciji slovo kliknuto

```

function slovoKliknuto(){
const slovo = this.textContent.toLowerCase();
// ispisivati za sada slovo koje je kliknuto
console.log(slovo)
}

```

```

// ako je slovo u reci, azurirati testContent od svakog spana
if (tajnaRec.includes(slovo)) {
    const indeksi = sviIndeksi(slovo);
    indeksi.forEach((val, i) => {
        recDiv.children[val].textContent = this.textContent;
    });
    if (proveriRec()) {
        prikaziPoruku("Pobedio si")
    }
} else {
// ako slovo nije u reci, smanjiti zivote
smanjiZivote();
}
this.classList.add('disabled'); // disejblujemo slovo za dalji klik
}

```

Korak 4: Testirati

- Disejblovat slova nakon kraja igre bilo pobeda bilo gubitak
- Dugme Novalgra ne radi

Korak 5: Dodavanje funkcije novalgra()

- Ne moze da se pozove samo init() jer se dupliraju onda slova
- Nacin 1
 - Potrebno je da se resetuju sve promenljive
 - Potrebno je da se dugmicima skine disable klasa
- Nacin 2
 - Reload the page

```

function novalgra() {
    location.reload();
}

```

Korak 6: Disejbl slova nakon kraja igre

- Nacin 1
 - Dodamo fleg kojim radimo razgranavanje da li su slova klikabilna

Korak 6a:

```

// dodajemo promenljivu
var tajnaRec;
var zivoti;
var slova;
var krajIgre = false;

```

Korak 6b:

// setujemo promenljivu na true kada se ispise pobeda ili poraz

```
function prikaziPoruku(txt) {  
    porukaDiv.classList.remove('hidden');  
    porukaTacnaRec.textContent = tajnaRec;  
    porukaSadrzaj.textContent = txt;  
    krajIgre = true;  
}
```

Korak 6c:

// dodajemo da se klikovi mogu desiti samo kada je kraj igre false

```
function slovoKliknuto(){  
    if (krajIgre === false) {  
        const slovo = this.textContent.toLowerCase();  
        // ispisivati za sada slovo koje je kliknuto  
        console.log(slovo)  
  
        // ako je slovo u reci, azurirati testContent od svakog spana  
        if (tajnaRec.includes(slovo)) {  
            const indeksi = sviIndeksi(slovo);  
            indeksi.forEach((val, i) => {  
                recDiv.children[val].textContent = this.textContent;  
            });  
            if (proveriRec()) {  
                prikaziPoruku("Pobedio si")  
            }  
        } else {  
            // ako slovo nije u reci, smanjiti zivote  
            smanjiZivote();  
        }  
        this.classList.add('disabled'); // disejblujemo slovo  
    }  
}
```