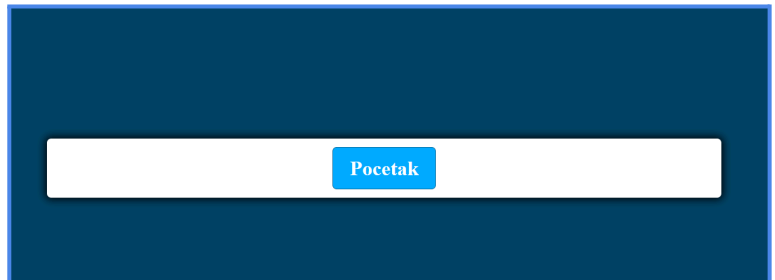


Igra 6

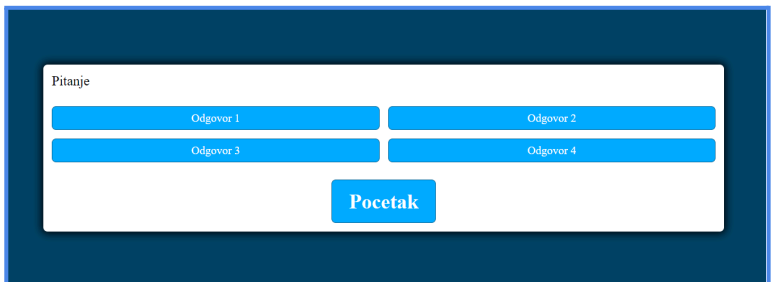
OBJAŠNJENJE KAKO RADI IGRICA:

- Igrač igra igricu tako što pogađa odgovore na data pitanja. Tačan odgovor je markiran u zelenu boju a netačni u crvenu.
- Može se dodati broj poena



IDEJA ZA IMPLEMENTACIJU:

- Deo1
 - ◆ Dodavanje u svoj projekat (novi dugmić, linkovi ažurirani)
 - ◆ Dodavanje HTML strukture i stil
 - ◆ Koristenje boja u hue notaciji
- Deo2
 - ◆ Dodavanje osnovnih promenljivih u javascriptu
 - ◆ Dodavanje 1 pitanja i strukture za pitanja
 - ◆ Dodavanje stila za tačno i netačno pitanje
- Deo3
 - ◆ Napraviti event listenere za dugmice next i za start
 - ◆ Prikaz pitanja
- Deo4
 - ◆ Provera tacnosti pitanja
 - ◆ Resetovanje prikaza
 - ◆ Dodavanje brojanja poena (TBA)



POSTUPAK:

HTML: osnovna struktura komponenti

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <link href="Cas_12.css" rel="stylesheet">
  <script defer src="Cas_12.js"></script>
```

```

<title>Quiz</title>
</head>
<body>
  <div class="container">
    <div id="question-container" >
      <div id="question">Pitanje</div>
      <div id="answer-buttons" class="btn-grid">
        <button class="btn">Odgovor 1</button>
        <button class="btn">Odgovor 2</button>
        <button class="btn">Odgovor 3</button>
        <button class="btn">Odgovor 4</button>
      </div>
    </div>
    <div class="controls">
      <button id="start-btn" class="start-btn btn">Pocetak</button>
      <button id="next-btn" class="next-btn btn hide">Sledece</button>
    </div>
  </div>
</body>
</html>

```

CSS: osnovni stil komponenti

```

*, *::before, *::after {
  box-sizing: border-box;
  font-family: Gotham Rounded;
}
:root {
  --hue-neutral: 200;
  --hue-wrong: 0;
  --hue-correct: 145;
}
body {
  --hue: var(--hue-neutral);
  padding: 0;
  margin: 0;
  display: flex;
  width: 100vw;
  height: 100vh;
  justify-content: center;
  align-items: center;
  background-color:
    hsl(var(--hue), 100%, 20%);
}
.container {
  width: 800px;
  max-width: 80%;
  background-color: white;
  border-radius: 5px;
  padding: 10px;
  box-shadow: 0 0 10px 2px;
}
.btn-grid {
  display: grid;
  grid-template-columns:
    repeat(2, auto);
  gap: 10px;
  margin: 20px 0;
}

```

```

.btn {
  padding: 10px 20px;
  --hue: var(--hue-neutral);
  border: 1px solid
hsl(var(--hue), 100%, 30%);
  background-color:
hsl(var(--hue), 100%, 50%);
  border-radius: 5px;
  padding: 5px 10px;
  color: white;
  outline: none;
}
.start-btn, .next-btn {
  font-size: 1.5rem;
  font-weight: bold;
}
.controls {
  display: flex;
  justify-content: center;
  align-items: center;
}
.hide {
  display: none;
}

```

OBJASNJENJE ZADATKA:

NAPOMENA 1: univerzalni selektor nije poznat možda deci. Rem sintaksa. Hue. Može da se radi i bez toga, da bude jednostavnije, pa kasnije ulepšavati.

NAPOMENA 2: može se staviti da bude npr sve vidljivo, dok se ne napravi sve, pa tek kasnije da se setuje display none

DEO 2:

- Korišćenje složenih tipova podataka u strukturama
- Classlist / add i remove klase

```

const startButton = document.getElementById('start-btn')
const nextButton = document.getElementById('next-btn')
const questionContainerElement =
document.getElementById('question-container')
const questionElement = document.getElementById('question')
const answerButtonsElement = document.getElementById('answer-buttons')

```

// struktura podataka za pitanja

```

const questions = [
  {
    question: 'Sta su celi brojevi u programiranju?',
    answers: [
      { text: 'int', correct: true },
      { text: 'float', correct: false }
    ]
  }
]

```

```

    ]
  },
  {
    question: 'Koja rec se koristi za definisanje funkcija u
javascriptu?',
    answers: [
      { text: 'def', correct: false },
      { text: 'main', correct: false },
      { text: 'function', correct: true },
      { text: 'func', correct: false }
    ]
  }
]

```

// setovanje stila za tačno i netačno pitanje

HTML: hajdovanje

```

<body>
  <div class="container">
    <div id="question-container" class="hide">
      <div id="question">Pitanje</div>
      <div id="answer-buttons" class="btn-grid">
        <button class="btn">Odgovor 1</button>
      </div>
    </div>
  </div>

```

CSS: dodavanje stila za body i dugmiće koji će se kroz JS dodati

```

body.correct {
  --hue: var(--hue-correct);
}
body.wrong {
  --hue: var(--hue-wrong);
}
.btn:hover {
  border-color: black;
}
.btn.correct {
  --hue: var(--hue-correct);
  color: black;
}
.btn.wrong {
  --hue: var(--hue-wrong);
}

```

JS: korak 1 - funkcije za setovanje i brisanje klasa stila

```
function setStatusClass(element, correct) {
  clearStatusClass(element)
  if (correct) {
    element.classList.add('correct')
  } else {
    element.classList.add('wrong')
  }
}

function clearStatusClass(element) {
  element.classList.remove('correct')
  element.classList.remove('wrong')
}
```

OBJASNJENJE ZADATKA:

NAPOMENA 1: Redosled radnji - osnovna struktura HTML fajla, dodavanje ostalih fajlova, dodavanje klasa za dugmice jedno po jedno

NAPOMENA 2: Dodati vise pitanja u listi

DEO 3:

- Inline definisanje funkcije kod event listenera
- Funkcija sort()
- Decimalni brojevi .5
- For each naredba

POSTUPAK:

JS: Event listeneri za start button i next button

```
let shuffledQuestions, currentQuestionIndex

startButton.addEventListener('click', startGame)
nextButton.addEventListener('click', () => {
  currentQuestionIndex++
  setNextQuestion()
})

function startGame() {
  startButton.classList.add('hide')
```

```

shuffledQuestions = questions.sort(() => Math.random() - .5)
currentQuestionIndex = 0
questionContainerElement.classList.remove('hide')
setNextQuestion()
}

```

JS: Prikaz pitanja

```

function showQuestion(question) {
  questionElement.innerHTML = question.question
  question.answers.forEach(answer => {
    const button = document.createElement('button')
    button.innerHTML = answer.text
    button.classList.add('btn')
    if (answer.correct) {
      button.dataset.correct = answer.correct
    }
    button.addEventListener('click', selectAnswer)
    answerButtonsElement.appendChild(button)
  })
}

```

OBJASNJENJE ZADATKA:

NAPOMENA 1: Random funkcija prisetiti se šta vraća

NAPOMENA 2: Cuvanje stanja da je nesto tacno ili ne unutar butona u datasetu

NAPOMENA 3: U show funkciji se prikazuje onoliko dugmica za odgovor koliko je odgovora setovano u strukturi postavljanja pitanja

DEO 4:

- Array.from()
- firstChild, removeChild
- Classlist - add i remove

POSTUPAK:

JS: Pravljenje funkcije next question

```

function setNextQuestion() {
  resetState()
  showQuestion(shuffledQuestions[currentQuestionIndex])
}

```

```
function resetState() {
  clearStatusClass(document.body)
  nextButton.classList.add('hide')
  while (answerButtonsElement.firstChild) {
    answerButtonsElement.removeChild(answerButtonsElement.firstChild)
  }
}
```

JS: Provera tačnosti odgovora

```
function selectAnswer(e) {
  const selectedButton = e.target
  const correct = selectedButton.dataset.correct
  setStatusClass(document.body, correct)
  Array.from(answerButtonsElement.children).forEach(button => {
    setStatusClass(button, button.dataset.correct)
  })
  if (shuffledQuestions.length > currentQuestionIndex + 1) {
    nextButton.classList.remove('hide')
  } else {
    startButton.innerText = 'Restart'
    startButton.classList.remove('hide')
  }
}
```