

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni diplomski studij

VIZUALIZACIJA PODATAKA

**Vizualizacija podataka o statistici rangiranih igrača
unutar igre Apex Legends**

Seminarski rad

Andreja Nađ

Osijek, 2022.

SADRŽAJ

<i>1. UVOD.....</i>	<i>1</i>
<i>2. KORIŠTENE TEHNOLOGIJE.....</i>	<i>2</i>
2.1. Biblioteka D3	2
2.2. SVG	2
2.3. JavaScript	2
<i>3. VIZUALIZACIJA PODATAKA</i>	<i>3</i>
<i>4. PROGRAMSKO RJEŠENJE.....</i>	<i>7</i>
<i>ZAKLJUČAK.....</i>	<i>12</i>
<i>LITERATURA.....</i>	<i>13</i>

1. UVOD

Izrada ovog projekta svodi se na prikaz podataka koji su sakupljeni sa web stranice „Apex Legends Status“ koja ima bazu od više od trinaest milijuna igrača. Podatci se skupljaju i izmjenjuju svake minute te su podatci korišteni u ovom radu uzeti sa iste stranice u malom skupu. Statistika se odnosi na prikaz rangiranih igrača i najčešće korištenih likova unutar rangiranih igara.

2. KORIŠTENE TEHNOLOGIJE

2.1. Biblioteka D3

D3.js je JavaScript biblioteka za manipulaciju dokumentima na temelju podataka. D3 omogućuje povezivanje proizvoljnih podataka s objektnim modelom dokumenta (DOM), te primijenjivanje transformacije vođene podacima na dokument.

2.2. SVG

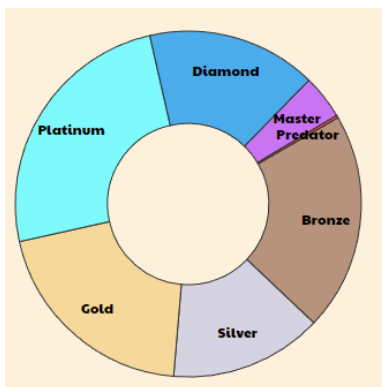
SVG je skraćenica od Scalable Vector Graphics. SVG definira vektorsku grafiku za web u XML formatu. Svaki element i svaki atribut u SVG datotekama može se animirati. SVG se integrira s W3C standardima kao što su DOM i XSL

2.3. JavaScript

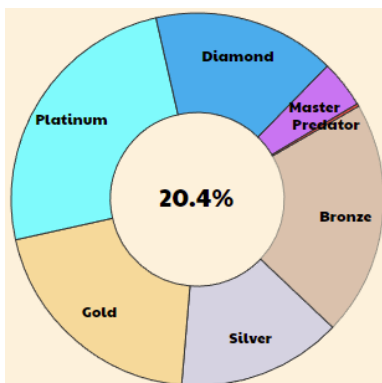
JavaScript je najpopularniji svjetski programski jezik weba. JavaScript je skriptni programski jezik, koji se izvršava u web pregledniku na strani korisnika. Napravljen je da bude sličan Javi, zbog lakšega korištenja, ali nije objektno orijentiran kao Java, već se temelji na prototipu i tu prestaje svaka povezanost s programskim jezikom Java. JavaScript je lako naučiti, ne mora se preuzimati i besplatan je.

3. VIZUALIZACIJA PODATAKA

Prvi prikaz je kružni graf koji je prikazan na slici 1, on predstavlja prikaz postotka rangiranih igrača u svakom rangu. Radi tako da kada se pređe pokazivačem preko elementa kruga, vrijednost dijela kruga će se prikazati u središtu kao na slici 2.

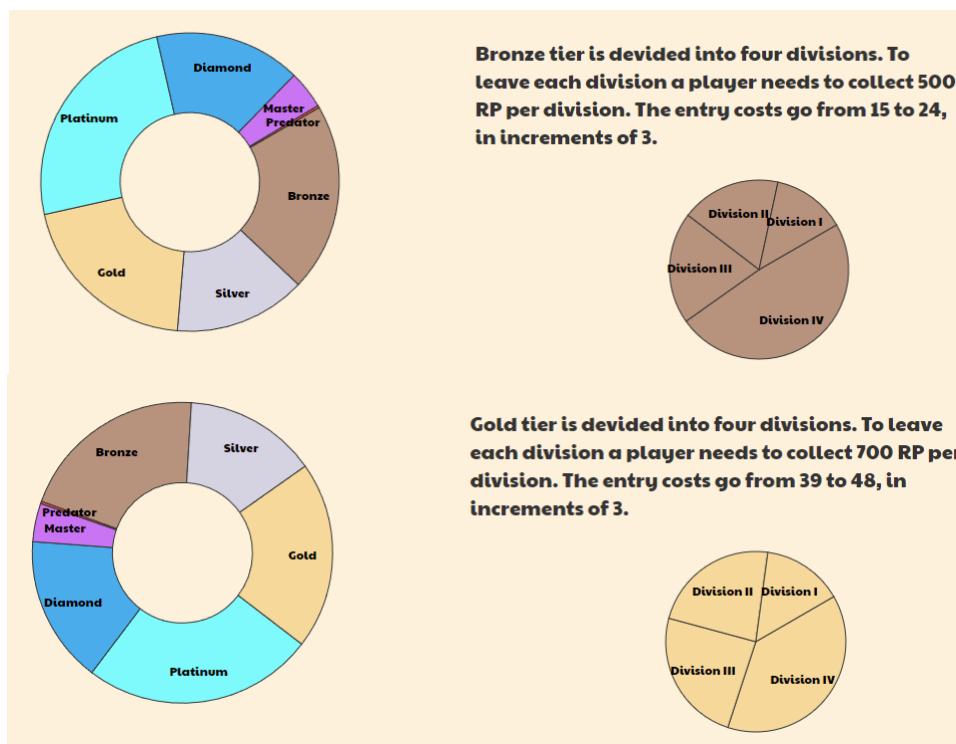


Slika 1. Kružni prikaz postotka rangiranih igrača sa rupom i korišteni podaci



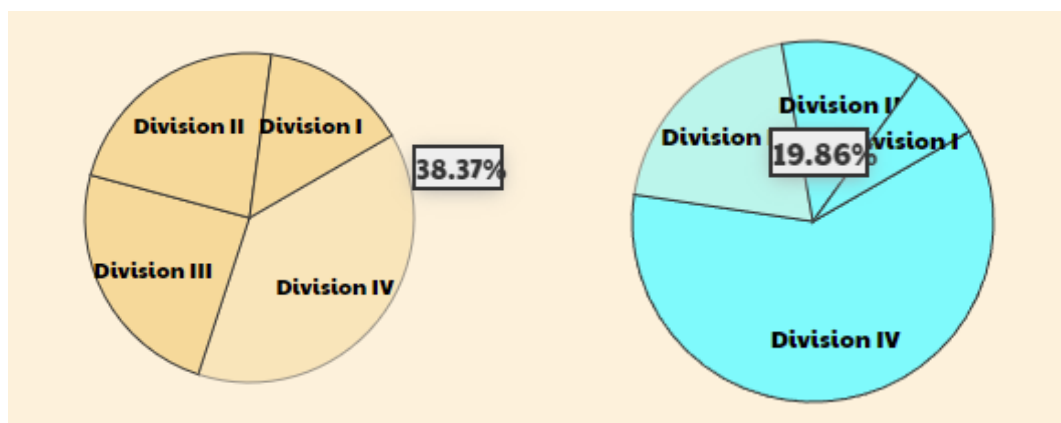
Slika 2. Prikaz vrijednost određenog dijela kružnog grafa

Druga funkcionalnost ovog grafa je povezana sa div elementom do njega. Kada se klikne na određeni dio kružnog grafa, kružni graf se rotira tako da na desnu stranu dolazi kliknuti dio te se prikaže opisni tekst vezan za dio kruga koji je pritisnut. Slika 3 prikazuje početno stanje stranice i stanje kada se klikne na „Gold“ dio u kružnom grafu.



Slika 3. Prikaz rada kružnog grafa

Ispod teksta nalazi još jedan kružni graf koji se mijenja ovisno o kliknutom elementu. Pošto skoro svaki rang unutar igrice ima četiri divizije, postotak igrača u svakoj od divizija je prikazan putem drugog kružnog grafa. Time, kružni graf ima više podataka koji se koriste te se podaci mijenjaju prema kliknutom elementu u prvom kružnom grafu. Prilikom klika na bilo koji dio prethodnog kružnog grafa, briše se prethodni mali kružni graf i crta se novi kako se grafovi ne bi gomilali jedan na drugoga. Ovaj graf ima dodatni alat za opis koji se aktivira kada pokazivač pređe preko kružnog grafa. Alat za opis koji se aktivira prikazuje postotak elementa te se pomiče za pokazivačem tako da ga prati. Slika 13 prikazuje funkcionalnost alata za opis koji prati pokazivač.



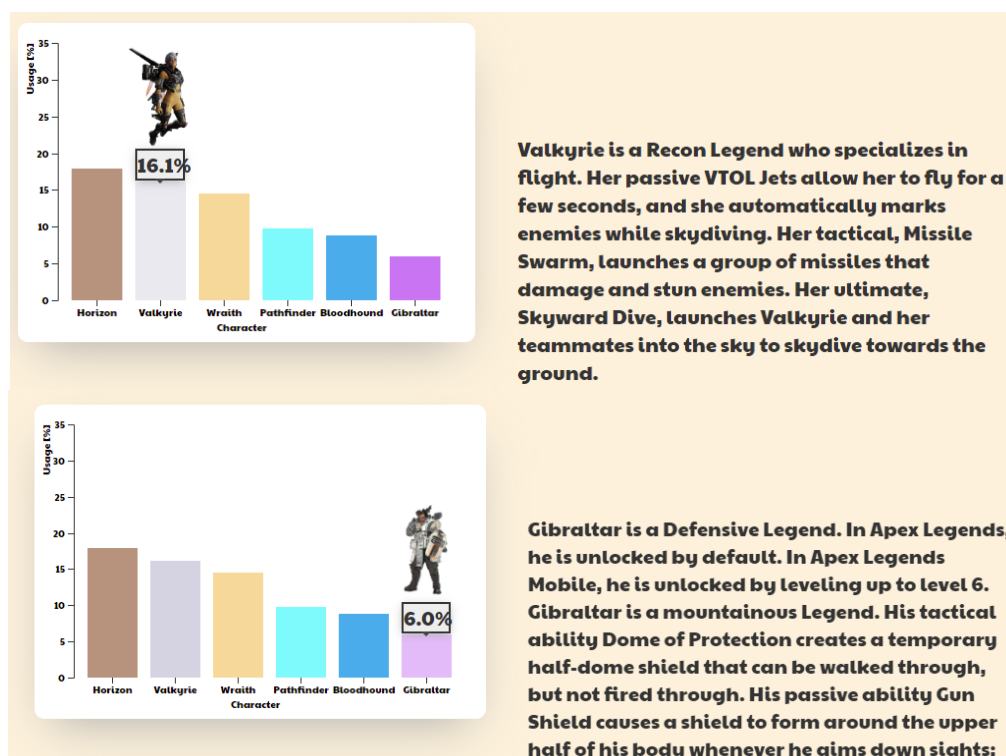
Slika 4. Prikaz rada alata za opis

Treći i zadnji graf je stupičasti graf koji prikazuje najpopularnije likove koji se koriste u rangiranim igrama.



Slika 5. Osnovni prikaz stupičastog grafa

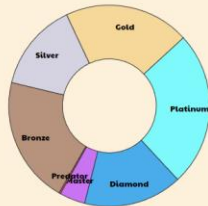
Iznad stupa na kojem je pokazivač, pokazuje se alat za opis sličan prethodnom. Te se iznad njega prikazuje opisna slika lika o kojem je riječ. Slika 20 prikazuje dva primjera alata za opis i slike koja je iznad njega. Slično kao kod prvog grafa, postoji tekstualni dio koje je vidljiv samo kada je pokazivač na nekom od stupova.



Slika 6. Primjer rada stupičastog grafa

Apex Legends Ranked Overview

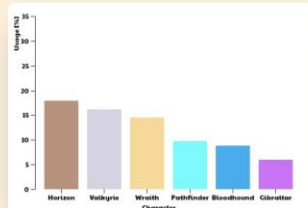
All the data collected count only the ranked games and only players who are not in the new Rookie category or are unranked.



Platinum tier is divided into four divisions. To leave each division a player needs to collect 700 RP per division. The entry costs go from 51 to 60, in increments of 3.



Looking at the ranked plays in the last few ranked seasons we can see the most popular characters which pros love to use. Seeing their skills we can tell that the pros love mobility of a character the most, since it gives the opportunity for an easy escape and fast engaging of enemies.



Slika 7. Prikaz gotove stranice

4. PROGRAMSKO RJEŠENJE

Varijabla *pieSvg* je vezana na svg element određen unutar *html*-a. Taj element već ima određene dimenzije te se one dobivaju koristeći funkciju *attr()* sa odgovarajućim nazivom atributa. Osim toga potreban je *radius* i *g* element koji predstavlja cijeli kružni graf sa početnom pozicijom rotacije. Unutar tog *g* elementa se nalaze drugi *g* elementi klase *arc* koji predstavljaju odvojene dijelove kružnog grafa. Njihove dimenzije ovise o danim podatcima te su automatski dodjeljene. Varijabla *percentageText* je varijabla na koju će se zapisivati trenutačni postupak dijela kružnog grafa kada se pređe preko njega putem pokazivača.

```
var pieSvg = d3.selectAll(".pieChart")
    width = +pieSvg.attr("width"),
    height = +pieSvg.attr("height"),
    radius = Math.min(width, height) / 2,
    g = pieSvg.append("g").attr("transform", "translate(" + width / 1.95 + ", "
+ height / 1.95 + ") rotate(" + 60 + ")")

var pie = d3.pie()
    .sort(null)
    .value(function(d) { return d.percentage; });

var path = d3.arc()
    .outerRadius(radius - 10)
    .innerRadius(radius - 90);

var label = d3.arc()
    .outerRadius(radius - 40)
    .innerRadius(radius - 40);

var arc = g.selectAll(".arc")
    .data(pie(pieData))
    .enter().append("g")
    .attr("class", "arc");

var percentageText = d3.selectAll(".pieChart").append("text")
    .text("%")
    .attr("x", radius + 45)
    .attr("y", radius + 10)
    .style("visibility", "hidden");
```

Slika 8. Kreiranje kružnog grafa 1/2

Prije spomenuti *g* elementi klase *arc* koji predstavljaju odvojene dijelove kružnog grafa su roditelji *path* elementa koji zapravo ima koordinate i dimenzije svakog dijela kružnog grafa. Gledajući da su *path* elementi isto što su *bar* elementi kod stupičastih grafova, onda na njih primjenjujemo elemente koje želimo da se prikazuju na određenom dijelu ili attribute za određeni dio kao što su boja ili slika na elementu. Varijabla *text* predstavlja naslov svakog ranga koji se pozicionira na dijelove kružnog prikaza.

```

arc.append("path")
  .attr("d", path)
  .attr("fill", function(d) { return d.data.color; });

var text = arc.append("text")
  .attr("transform", function(d) { return "translate(" + label.centroid(d)
+ ") rotate(" + (-60) + ")"; })
  .attr("dy", "0.35em")
  .text(function(d) { return d.data.rank; });

```

Slika 9. Kreiranje kružnog grafa 2/2

Kao prije spomenuto, na *path* element postavljamo attribute i funkcije vezane za te elemente. Neke od njih su funkcije *on()* koje se bave događajima kao na primjer kada pokazivač pređe preko elementa, klikne na element ili makne se sa elementa. Prijelaz preko elementa je prikazan na donjoj slici gdje se traži trenutni naziv elementa i putem toga se prikazuje postotak ranga elementa unutar prije spomenute varijable *percentageText*.

```

arc.append("path")
  .attr("d", path)
  .attr("fill", function(d) { return d.data.color; })
  .on("mouseover", function(d){
    var name = this.parentNode.lastChild.innerHTML;
    var text;
    console.log(pieData)
    pieData.forEach(element => {
      if(name === element['rank']){
        text = element['percentage'] + "%"
      }
    });
    percentageText.style("visibility", "visible").text(text);
  })
  .on("mouseout", function(d){percentageText.style("visibility", "hidden")})

```

Slika 10. Mouseover funkcionalnost na kružnom grafu

Funkcija *on()* koja se bavi događajem na klik je prikazana dolje. Putem nje se rotira kružni graf tako da se kliknuti dio kruga prikaže prema desno gdje se nalazi div na kojem se prikazuje opisni tekst ovisan o dijelu koji je pritisnut, te se ispod teksta nalazi još jedan kružni graf koji se mijenja ovisno o kliknutom elementu.

```

.on("click", function(d) {
    var name = this.parentNode.lastChild.innerHTML;
    var transf = this.parentNode.parentNode.getAttribute('transform');
    var rotation;

    if(transf.includes("rotate")){
        rotation = transf.split("rotate")[1];
        rotation = rotation.split("(")[1];
        rotation = rotation.split(")")[0];
    }
    else{
        rotation = 0;
    }
    pieData.forEach(element => {
        if(name === element['rank']){
            rankText.text(element['text']);
        }
    });

    if(name === "Bronze")
    {
        RotateThis(60)
        deleteChart()
        ChangeChart(bronzeData);
    }
}

```

Slika 11. Click funkcionalnost na kružnom grafu

Funkcija *RotateThis()* rotira kružni graf te sve labelle dijelova u zadanom broju.

```

function RotateThis(rotate){
    g.transition()
    .attr("transform", "translate(" + width / 1.7 + "," +
+ height / 1.58 + ") rotate(" + rotate + ")")
    .duration(1000);

    text.transition()
    .attr("transform", function(dd) {
        return "translate(" + label.centroid(dd) + ") rotate("
+ (-rotate) + ")"; })
    .duration(1000);
}

```

Slika 12. Funkcija *RotateThis()*

Važne funkcije za rad drugog kružnog grafa su funkcija *deleteChart()* koja je odgovorna za brisanje prethodnog grafa prije prikaza novog, drugog malog kružnog grafa. Ako se prethodni ne obriše, grafovi će se gomilati jedan na drugoga. Druga važna funkcija je *ChangeChart()* koja kreira kružni graf kao i prethodni. Ovaj graf ima dodatni alat za opis koji se aktivira kada pokazivač pređe preko kružnog grafa.

```

function deleteChart(){ divPieSvg.selectAll("g").remove();}

```

Slika 13. Funkcija *deleteChart()*

```

function ChangeChart(divPieData){
  currentDivPieData = divPieData;
  divPieSvg = d3.selectAll(".divPieChart").style("visibility", "visible")
  .....
  var divPath = d3.arc()
    .outerRadius(divPieRadius - 10)
    .innerRadius(0);
  .....
  var divArc = divG.selectAll(".arc")
    .data(pie(divPieData))
    .enter().append("g")
    .attr("class", "arc");

  divArc.append("path")
    .attr("d", divPath)
    .attr("fill", function(d) { return d.data.color; })

```

Slika 14. Prvi dio funkcije *ChangeChart()* za kreiranje kružnog prikaza prema predanim podacima

Alat za opis koji se aktivira prikazuje postotak elementa te se pomiče za pokazivačem tako da ga prati. Pozicija pokazivača se dobiva putem *event* varijable koja predstavlja objekt klase *MouseEvent* te sadrži attribute *clientX* i *clientY* koji predstavljaju koordinate pokazivača.

```

.on("mousemove", function(d){
  var colors = this.getAttribute('fill');
  var name = this.parentNode.lastChild.innerHTML;
  var text;
  currentDivPieData.forEach(element => {
    if(name === element['rank']){
      text = element['percentage']
    }
  });
  console.log(event)
  div.style("opacity", 1)
  div.html(text + "%")
    .style("left", event.clientX + 20 + "px")
    .style("top", event.clientY - 20 + "px");
})
.on("mouseout", function(d){
  div.transition()
    .duration(50)
    .style("opacity", 0);
});

```

Slika 15. Drugi dio funkcije *ChangeChart()* - funkcionalnost prelaska pokazivčala

```

var div = d3.select(".divPie").append("div")
  .attr("class", "tooltipPIE")
  .style("opacity", 0);

```

Slika 16. Osnovne postavke za kreiranje alata za opis

Kod stupičastog grafa, iznad stupa na kojem je pokazivač, pokazuje se alat za opis sličan prethodnom. Aktiviran je metodom *onMouseOver()* koja isto tako aktivira funkciju *imageAdd()* koja nadodaje sliku dimenzija 100x100 iznad alata za opis.

```

var tip = d3.select(".chart-container")
    .append("div")
    .attr("class", "tip")
    .style("position", "absolute")
    .style("z-index", "10")
    .style("visibility", "hidden");

function imageAdd(imageName, x, y){
    barImg = barSvg.append('image')
        .attr('xlink:href', imageName)
        .attr('width', 100)
        .attr('height', 100)
        .attr('x', x)
        .attr('y', y);
}

```

Slika 17. Alat za opis i funkcija imageAdd()

```

function onMouseOver(event){
    var percentage = (((258-event.target.y.baseVal.value)/258)*35).toFixed(1);
    var rank = (event.target.x.baseVal.value) + 12 + 56/2;
    var rankNumber = Math.floor((event.target.x.baseVal.value)/56);

    var imageName;
    charText.style("visibility", "visible");

    var i = 0;
    barData.forEach(element => {
        if(rankNumber == i){
            imageName = element['image'];
            charText.text(element['text']);
        }
        i++;
    });

    tip.style("visibility", "visible")
        .text(percentage + "%")
        .style("top", (y(percentage) - 13) + "px")
        .style("left", (rank) + "px");

    imageAdd(imageName, event.target.x.baseVal.value + 15,
event.target.y.baseVal.value - 115)
}

```

Slika 18. Funkcija onMouseOver()

ZAKLJUČAK

Najveće poteškoće izašle su iz manjka primjera programskog koda koji su rađeni na biblioteci D3.js verziji sedam. Dosta pronađenih primjera su rađeni u verziji četiri i tri koje koriste zastarjele funkcije. Teško je pronaći kako da se neke zastarjele stvari koriste u verziji sedam. To se najviše osjetilo u početnom kutu kružnog grafa, varijabla koju smo u prijašnjim verzijama mogli automatski dohvatiti na lagan način, no u ovoj verziji nisam uspješno našla lagan način pa sam morala improvizirati sa zadavanje za koliko se rotira umjesto računanjem.

Vizualizacija ovih podataka je bila zanimljiva za izraditi, iako ponekad sam morala odustat od nekih ideja ili zbog premalo vremena ili zbog ne pronalaska funkcija u novoj verziji biblioteke D3.js. Sve u svemu, D3 je jako moćan alat zbog kojeg je uopće bilo moguće napraviti ovakav projekt u kratkom vremenu bez previše programiranja.

Stranica je izrađena putem biblioteke *bootstrap* kako bi omogućili responzivnost dizajnu.

LITERATURA

1. <https://d3js.org/>
2. <https://github.com/d3/d3>
3. <https://github.com/d3/d3/blob/main/CHANGES.md>
4. <https://stackoverflow.com/>