

Vizualizacija podataka

World of tanks statistics

PROJEKTNI ZADATAK

Izradio:

Josip Rizner

Osijek, 2022.

Sadržaj

1. UVOD	1
2. KORIŠTENE TEHNOLOGIJE	2
2.1. HTML	2
2.2. CSS	3
2.3. Bootstrap	3
2.4. JavaScript	4
2.5. D3	4
3. GRAFOVI	5
3.1. Stupčasti grafikon	5
3.1.1. Programski kod – Stupčasti grafikon	6
3.2. Radar chart	8
3.2.1. Programski kod - Radar chart	9
3.3. Animacije	12
3.3.1. Programski kod - animacije	13
4. ZAKLJUČAK	18
5. LITERATURA	19
6. PRILOZI	20

1. UVOD

Ljudima je vrlo teško izvući bilo kakav zaključak iz nekakve nakupine podataka, raznih tablica ili sličnog izvora podataka. Iz tog razloga je vizualizacija podataka vrlo bitna, dobrom vizualizacijom može se vrlo lako prenijeti bit prikazanih podataka. Što su podaci bolje i jednostavnije prikazani, lakše se prenosi korisna informacija. Isto tako, bitno je odabrati grafove koji će biti adekvatni za prikazivanje pojedinog problema. S druge strane, cilj vizualizacije podataka ne mora biti samo prikazati nekakav podatak, nego i ispričati priču. U ovom radu koriste se dva grafa i dvije animacije kako bi se inače vrlo dosadne karakteristike izlistane u tablicama prikazale na zanimljiviji način i tako olakšale razumijevanje istih.

2. KORIŠTENE TEHNOLOGIJE

2.1. HTML

U [1] i [2] je navedeno kako je HTML je kratica od engleskog pojma HyperText Markup Language što je naziv jezika za izradu međusobno povezanih dokumenata. HTML nije programski jezik već je skup pravila kako renderirati stranicu. Sastoji se od tekstualnih oznaka (engl. tags) koje pregledniku daju upute o strukturi i rasporedu elemenata na stranici koju je potrebno prikazati. Elementi stranice sastoje se od oznake početka elementa, sadržaja elementa i oznake kraja elementa. HTML oznake pišu se unutar znakova < i >, a unutar kojih, osim samog imena oznake, mogu biti navedeni različiti atributi koji dodatno opisuju specifična svojstva pojedinog elementa. Svi HTML elementi mogu imati svojstva. Neka od najčešće upotrebljivanih svojstava su id, name i class, a njihova namjena je omogućiti odabir određenog elementa prema jedinstvenom identifikatoru, prema imenu, odnosno definirati pripadnost određenoj klasi elemenata.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>

</body>

</html>
```

2.2. CSS

Kao što je opisano u izvoru [3], CSS (engl. *Cascading Style Sheet*) stilski je jezik. Služi sa oblikovanje sadržaja koji je prethodno smješten u HTML-u. Sve elemente HTML dokumenta može se urediti koristeći CSS. Elementi HTML-a prikazuju se na stranici onako kako su definirani u CSS-u. Svako pravilo koje se koristi sastoji se od selektora (engl. *Selector*) i deklaracijskog bloka. Selektor označava dio na koji se primjenjuje stil. Selektor može biti *id* što predstavlja jedinstveni element ili *class* koji obuhvaća više od jednog elementa. Deklaracijski blok predstavljaju vitičaste zagrade unutar kojih je deklaracija. Deklaracija se sastoji od svojstva koje se uređuje, dvotočke i vrijednosti koje se pridodaju svojstvu. Između uzastopnih deklaracija mora se nalaziti točka-zarez.

2.3. Bootstrap

Bootstrap izrađen 2011. godine od strane Marka Otta i Jacoba Thorntona, web developera koji rade u Twitteru. Njihov glavni cilj je bila konzistentnost i lako održavanje koda. Bootstrap 1.0.0 je izašao 2011. godine sa samo HTML i CSS komponentama. Javascript pluginovi nisu bili uključeni sve do verzije 1.3.0 koja je također bila i kompatibilna sa IE7 i IE8. Bootstrap danas je vrlo popularan razvojni okvir koji nam omogućava brzo i jednostavno kreiranje web. Ima jako puno ugrađenih izgleda pojedinih komponenti, što nam znatno olakšava rad s elementima i njihovim pozicioniranjem.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.
css" rel="stylesheet"
  integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfFspd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">

  <title>Document</title>
</head>
<body>
</body>
</html>
```

2.4. JavaScript

Kako je opisano u izvoru [5], 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. Kako bi mogli koristiti .js datoteku, potrebnu ju je također uključiti da je web sučelje prepozna. U priloženom kodu prikazano je uključivanje .js datoteke u HTML.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.
css" rel="stylesheet"
integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">
    <script src="script.js"></script>
    <title>Document</title>
</head>
<body>
</body>
</html>
```

2.5. D3

D3 je JavaScript biblioteka za manipulaciju dokumenata koji se baziraju na podacima. Korištenjem HTML-a, SVG-a i CSS-a vrši manipulaciju nad podacima kako bi se mogle izraditi vizualizacije koje će prikladno, odnosno točno, fleksibilno, koncizno, precizno i što jednostavnije prikazati podatke iz kojih se mogu što lakše iščitati informacije. Zbog svih tih odlika, D3 biblioteka je korištena u ovom radu [3]. Uključivanje d3 prikazano je kodom:

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

3. GRAFOVI

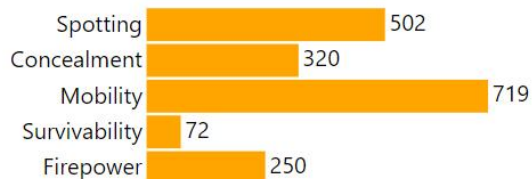
3.1. Stupčasti grafikon

Stupčasti grafovi (engl. *bar chart*) sastoje se od pravokutnih oblika, čije su duljine proporcionalne vrijednostima koje oni predstavljaju. Stupčani grafikoni koriste se za uspoređivanje dvije ili više vrijednosti. Ti pravokutni oblici mogu biti okomito ili vodoravno orijentirani. Njihova implementacija pomoću biblioteke D3 je jednostavna jer se realiziraju pomoću `<rect>` elemenata koji su pravilno razmaknuti jedan od drugoga, imaju jednaku širinu, a visinu ovisno o podacima u slučaju vertikalnih stupaca ili jednaku visinu, a širinu ovisnu o podacima u slučaju horizontalnih stupaca. Pomoću stupčanih grafikona možemo dobiti vizualni prikaz usporedbe skupina podataka ili učestalosti pojavljivanja različitih osobina podataka [2].



Cromwell

Tier VI Sniper Medium Tank



[Details](#)

Slika 3.1. Stupčasti dijagram – karakteristike tenka

3.1.1. Programski kod – Stupčasti grafikon

Stupčani grafikon u ovome projektu je korišten za prikaz različitih karakteristika tenka. Prikazuju se bodovi za pet različitih karakteristika, a to su: izviđanje (engl. spotting), kamuflaža (engl. concealment), pokretljivost (engl. Mobility), otpornost (engl. survivability) i vatrena moć (engl. Firepower). Bodovi za svaku karakteristiku su preuzeti iz *World of Tanks* igrice.

```
var data = [{
  "name": "Firepower",
  "value": 250,
},
{
  "name": "Survivability",
  "value": 72,
},
{
  "name": "Mobility",
  "value": 719,
},
{
  "name": "Concealment",
  "value": 320,
},
{
  "name": "Spotting",
  "value": 502,
}
];

var margin = {
  top: 5,
  right: 40,
  bottom: 15,
  left: 95
};

var width = 375 - margin.left - margin.right,
    height = 150 - margin.top - margin.bottom;

var svg = d3.select(".card-stats-cromwell").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var x = d3.scale.linear()
  .range([0, width])
  .domain([0, d3.max(data, function (d) {
    return d.value;
  })]);
```



```

    }));

var y = d3.scale.ordinal()
    .rangeRoundBands([height, 0], .1)
    .domain(data.map(function (d) {
        return d.name;
    }));

var yAxis = d3.svg.axis()
    .scale(y)
    .tickSize(0)
    .orient("left");

var gy = svg.append("g")
    .attr("class", "y axis")
    .call(yAxis)

var bars = svg.selectAll(".bar")
    .data(data)
    .enter()
    .append("g")

bars.append("rect")
    .attr("class", "bar")
    .attr("y", function (d) {
        return y(d.name);
    })
    .attr("height", y.rangeBand())
    .attr("x", 0)
    .attr("fill", "orange")
    .attr("width", function (d) {
        return x(d.value);
    });

bars.append("text")
    .attr("class", "label")
    .attr("y", function (d) {
        return y(d.name) + y.rangeBand() / 2 + 4;
    })
    .attr("x", function (d) {
        return x(d.value) + 3;
    })
    .text(function (d) {
        return d.value;
    });

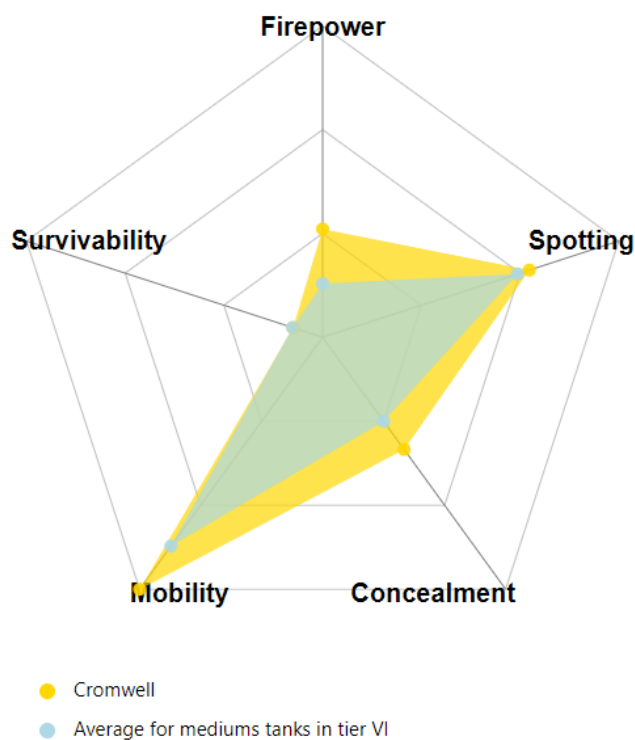
```

U varijablu data su bodovi i ime za svaku karakteristiku tenka. Na početku definiramo varijable za margine, širinu i visinu koje će se koristiti dalje u kodu. Pomoću funkcije *select()* odabiremo html element u kojem će se graf nalaziti. U ovom slučaju to je `<div>` elementu kojem pristupamo preko njegove klase. Kada je kreiran canvas, postavljaju se osi i kreiraju se `<rect>`

elementi pomoću kojih prikazujemo bodove za određenu karakteristiku tenka. Na kraju se dodaju labele kako bi prikazani podaci dobili značenje. Ovaj graf preuzet je i prilagođen s izvora [7].

3.2. Radar chart

Radar chart je dvodimenzionalni tip grafikona dizajniran za iscertavanje jedne ili više grupa vrijednosti preko više kvantitativnih varijabli. Svaka varijabla ima svoju os, a sve osi su spojene u središtu slike. Ova vrsta grafikona ima nekoliko nedostataka i treba se pažljivo koristiti.



Slika 3.2. Radar chart – karakteristike tenka

3.2.1. Programski kod - Radar chart

```
RadarChart.defaultConfig.color = function () { };
RadarChart.defaultConfig.radius = 5;
RadarChart.defaultConfig.w = 500;
RadarChart.defaultConfig.h = 500;

var data = [
  {
    className: 'specific',
    axes: [
      { axis: "Firepower", value: 250 },
      { axis: "Survivability", value: 72 },
      { axis: "Mobility", value: 719 },
      { axis: "Concealment", value: 320 },
      { axis: "Spotting", value: 502 }
    ]
  },
  {
    className: 'average',
    axes: [
      { axis: "Firepower", value: 124 },
      { axis: "Survivability", value: 73 },
      { axis: "Mobility", value: 596 },
      { axis: "Concealment", value: 240 },
      { axis: "Spotting", value: 473 }
    ]
  }
];

var chart = RadarChart.chart();
var cfg = chart.config(); // retrieve default config
var svg = d3.select('#radar_chart').append('svg')
  .attr('width', cfg.w + cfg.w)
  .attr('height', cfg.h + cfg.h / 4);

svg.append('g').classed('single', 1).datum(data).call(chart);

svg.append("circle").attr("cx", 40).attr("cy", cfg.h + 20).attr("r",
6).style("fill", "#FFD700")
svg.append("circle").attr("cx", 40).attr("cy", cfg.h + 50).attr("r",
6).style("fill", "#ADD8E6")
svg.append("text").attr("x", 60).attr("y", cfg.h +
20).text("Cromwell").style("font-size", "15px").attr("alignment-baseline",
"middle")
svg.append("text").attr("x", 60).attr("y", cfg.h + 50).text("Average for
mediums tanks in tier VI").style("font-size", "15px").attr("alignment-
baseline", "middle")
```

U programskom kodu iznad prikazani su podatci koji se prikazuju pomoću *Radar chart-a*. Podaci su jednaki kao i na stupčastom dijagramu, ali u ovom slučaju još se prikazuju bodovi za prosječnog tenka tog tipa i te razine. Isto tako može se vidjeti kreiranje kanvasa i legende koja se nalazi ispod grafikona. Za iscrtavanje se još dohvaća .js datoteka preuzeta s [9]. Grafikon reagira na prelazak miša tako što ističe karakteristike za specifični tenk za koji smo otvorili stranicu, odnosno za prosjek tenkova istog tipa i iste razine. Isto tako, na projelaz miša ispisuju se brojčane vrijednosti za karakteristiku iznad koje se strelica miša trenutno nalazi.

```
.radar-chart .level {
  stroke: grey;
  stroke-width: 0.5;
}

.radar-chart .axis line {
  stroke: grey;
  stroke-width: 1;
}

.radar-chart .axis .legend {
  font-family: sans-serif;
  font-size: 20px;
  font-weight: bold;
}

.radar-chart .axis .legend.top {
  dy: 1em;
}

.radar-chart .axis .legend.left {
  text-anchor: start;
}

.radar-chart .axis .legend.middle {
  text-anchor: middle;
}

.radar-chart .axis .legend.right {
  text-anchor: end;
}

.radar-chart .tooltip {
  font-family: sans-serif;
  font-size: 20px;
  transition: opacity 200ms;
  opacity: 0;
}

.radar-chart .tooltip.visible {
  opacity: 1;
}

/* area transition when hovering */
.radar-chart .area {
  stroke-width: 2;
```

```

    fill-opacity: 0.5;
}
.radar-chart.focus .area {
    fill-opacity: 0.1;
}
.radar-chart.focus .area.focused {
    fill-opacity: 0.7;
}

.radar-chart .circle {
    fill-opacity: 0.9;
}

/* transitions */
.radar-chart .area, .radar-chart .circle {
    transition: opacity 300ms, fill-opacity 200ms;
    opacity: 1;
}
.radar-chart .d3-enter, .radar-chart .d3-exit {
    opacity: 0;
}

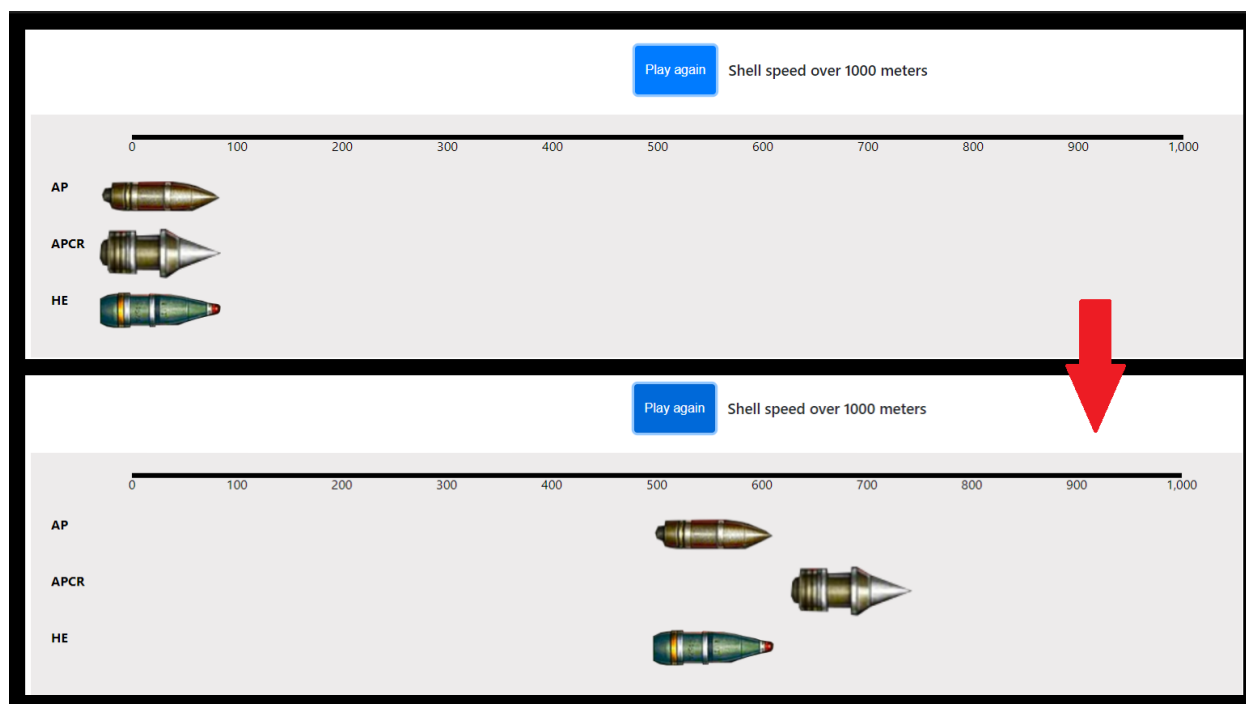

.radar-chart .area {
    fill-opacity: 0.7;
}
.radar-chart.focus .area {
    fill-opacity: 0.3;
}
.radar-chart.focus .area.focused {
    fill-opacity: 0.9;
}
}
.area.specific, .specific .circle {
    fill: #FFD700;
    stroke: none;
}
}
.area.average, .average .circle {
    fill: #ADD8E6;
    stroke: none;
}
}

```

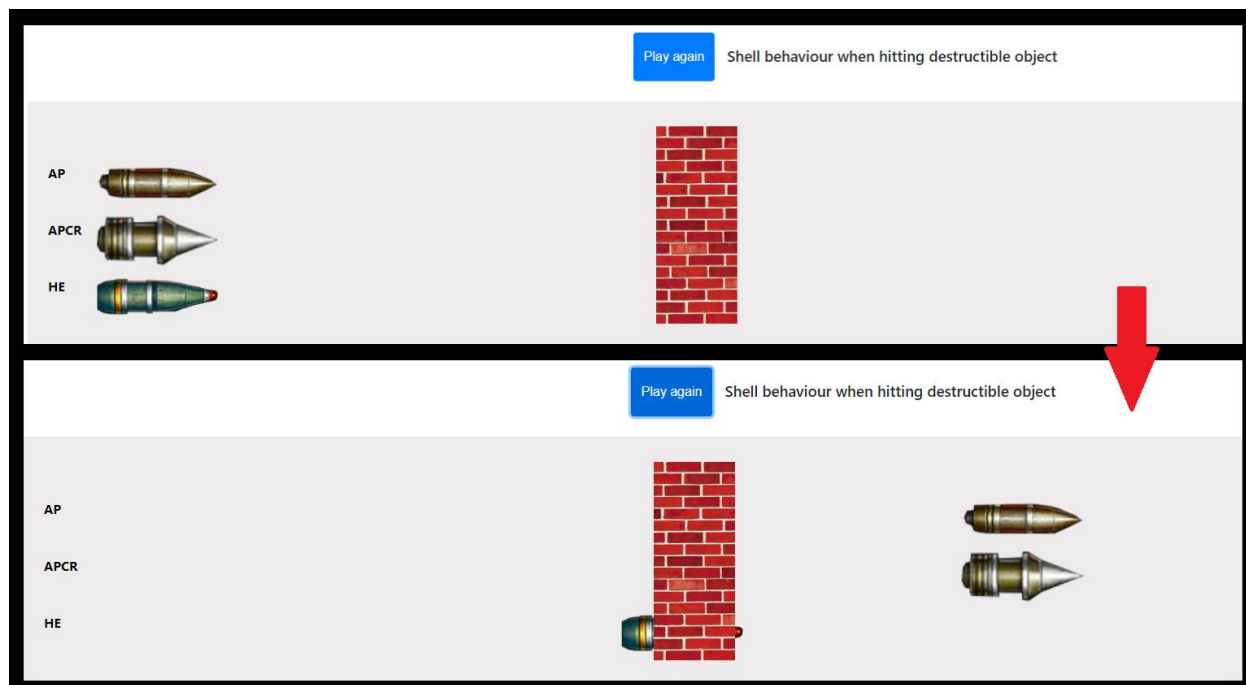
Boja, font, prozirnost reakcija na postavljanje miša iznad se još dodatno podešava pomoću CSS-a. Kod CSS datoteke kojim se podešava izgled može se vidjeti iznad. Ovaj *radar chart* preuzet je i prilagođen iz izvora [8].

3.3. Animacije

Osim svih karakteristika tenkova, vrlo bitan dio igre *World of Tanks* predstavljaju granate. Svaki tenk u igri ima do tri različita tipa granata. Granate se razlikuju po tipu, ponašanju pri koliziji s različitim objektima, prosječnoj penetraciji i prosječnoj šteti. Prosječna penetracija i šteta spadaju pod kategoriju vatrene moći na prethodna dva grafikona, a brzina i ponašanje pri koliziji s uništivim objektima prikazano je pomoću dvije animacije. Na prvoj animaciji prikazane su brzine svake od granata na tisuću metara. Na drugoj animaciji prikazano je kako će granata reagirati kada dođe do kolizije s uništivim objektom. Kada dođe do kolizije, ovisno o tipu, granate će ili eksplodirati ili proći kroz objekt i nastaviti letjeti.



Slika 3.3. Prva animacija – Brzina različitih vrsta granata na 1000 metara



Slika 3.4.. Druga animacija – Ponašanje granate pri koliziji s uništivim objektom

3.3.1. Programski kod - animacije

Kao što je opisano u poglavlju iznad, projekt sadrži dvije animacije. Prva prikazuje brzinu granata na tisuću metara i programski kod ove animacije može se vidjeti ispod.

```
var shell_data = [
  {
    "type": "AP",
    "speed": "785"
  },
  {
    "type": "APCR",
    "speed": "981"
  },
  {
    "type": "HE",
    "speed": "785"
  }
];

function translate_x(d) {
  return "translate(" + d + ", " + 0 + ")";
}

function translate_back(d) {
  return "translate(" + 0 + ", " + 0 + ")";
}
```

```

var width = 1500;
var height = 300;

var svg_speed = d3.select('#shell_speed').append('svg')
  .attr('width', width)
  .attr('height', height)
  .style("background", "#edebeb")
  .style("margin", "25px")
  .style("padding", "25px");

var scale = d3.scale.linear()
  .domain([0, 1000])
  .range([100, 1400]);

var x_axis = d3.svg.axis()
  .scale(scale);

svg_speed.append("g")
  .call(x_axis)
  .attr("x", 50)
  .attr("y", 100);

var shells = svg_speed.selectAll("image.shells")
  .data(shell_data)
  .enter()
  .append("svg:image")
  .attr("x", function (d) { return 60 })
  .attr("y", function (d, i) { return 1 + (i) * 70 })
  .attr('width', 150)
  .attr('height', 150)
  .attr("xlink:href", function (d) { return
  ../static/images/${d.type}.png });

var shellTypes = svg_speed.selectAll("h1")
  .data(shell_data)
  .enter()
  .append("svg:text")
  .style("font-weight", "bold")
  .text(function (d) { return ${d.type} })
  .attr("x", function (d) { return 0 })
  .attr("y", function (d, i) { return 70 + (i) * 70 });

d3.select("#replay_shell_speed_animation").on("click", function () {
  shells.transition()
    .duration(function (d, i) { return
  calculateTimeToTravelFullDistance(d.speed) * 1000 })
    .ease("linear")
    .attr("transform", translate_x(1300))
    .transition()

```



```

        .duration(0)
        .attr("transform", translate_back)
    });

    function calculateTimeToTravelFullDistance (speed) {
        return 1000 / speed;
    }

```

U programsko kodu iznad mogu se vidjeti podaci koji se koriste kod obje animacije. Isto tako vidi se kako se prvo kreira kanvas. Na kanvas se dodaje x-os koja predstavlja udaljenost koju je granata prošla, slike granata s obzirom na tip granata te ime, tj. tip granate na y-os. Pritiskom na gumb *Play again* pokreće se animacija, kada animacija završi automatski se resetira. Kako bi se odredilo vrijeme koje je granati potrebno da preleti tisuću metara, koristi se jednostavna funkcija *calculateTimeToTravelFullDistance()*. Funkcija prima brzinu granate te vraća vrijeme u sekundama koje određuje koliko će granati trebati da preleti s lijeve strane kanvasa na desnu.

Programski kod za drugu animaciju prikazan je ispod.

```

var width = 1500;
var height = 300;

var svg_behaviour = d3.select('#shell_behaviour').append('svg')
    .attr('width', width)
    .attr('height', height)
    .style("background", "#edebeb")
    .style("margin", "25px")
    .style("padding", "25px");

var shells_behaviour = svg_behaviour.selectAll("image.shells")
    .data(shell_data)
    .enter()
    .append("svg:image")
    .attr("class", function (d) { if (d.type == "AP" || d.type == "APCR") { return "penetrates_dest_obj" } return "explodes_on_impact" })
    .attr("x", function (d) { return 60 })
    .attr("y", function (d, i) { return 1 + (i) * 70 })
    .attr('width', 150)
    .attr('height', 150)
    .attr("xlink:href", function (d) { return
    ../static/images/${d.type}.png });

var shellTypes_behaviour = svg_behaviour.selectAll("h1")
    .data(shell_data)
    .enter()

```

```

.append("svg:text")
.style("font-weight", "bold")
.text(function (d) { return "${d.type}" })
.attr("x", function (d) { return 0 })
.attr("y", function (d, i) { return 70 + (i) * 70 });

var wall = svg_behaviour.append('image')
.attr('xlink:href', '../static/images/wall.png')
.attr('width', 100)
.attr('height', height)
.attr('x', width / 2);

var behaviour1Shells = svg_behaviour.selectAll("image.penetrates_dest_obj");
var behaviour2Shells = svg_behaviour.selectAll("image.explodes_on_impact");

d3.select("#replay_shell_behaviour_animation").on("click", function () {

    behaviour1Shells.transition()
        .duration(function (d, i) { return 1500 })
        .ease("linear")
        .attr("transform", translate_x(1300))
        .transition()
        .duration(0)
        .attr("transform", translate_back)
        .delay(1500)

    behaviour2Shells.transition()
        .duration(function (d, i) { return 750 })
        .ease("linear")
        .attr("transform", translate_x(width / 2 - 100))
        .transition()
        .delay(1500)
        .duration(0)
        .delay(1500)
        .attr("transform", translate_back)
        .delay(1500)

});

```

Kao što je već spomenuto, obje animacije koriste iste podatke. Isto tako, obje animacije koriste funkcije *translate_x()* koja se koristi za određivanje do kuda će granata letjeti i funkciju *translate_back()* koja se poziva kada se animacija resetira. Animacija koja prikazuje ponašanje granate prilikom kolizije s uništivim objektom također ima svoj *Play again* gumb kojim se pokreće. Kada dođe do kolizije, može se dogoditi jedan od dva slučaja, granata će ili zapeti u zidu ili proletjeti kroz zid i nastaviti letjeti. Što će se dogoditi, odlučuje tip granate. Prilikom dodavanja

slika na canvas, *image* elementu dodijeljena je jedna klasa s obzirom na tip granate koji je na slici. Kasnije se *image* elementi dohvaćaju s obzirom na tu klasu te se prema klasi prikazuje odgovarajuća animacija.

4. ZAKLJUČAK

D3.js biblioteka se pokazala jako učinkovitom prilikom uzimanja dosadnih podataka iz tablica i prikazivanjem istih na razne načine. Isto tako, omogućava jednostavnu i brzu izradu različitih vrsta grafova i animacija kojom se lako može prenijeti informacija ili nekakva priča. Potreno je dosta vremena kako bi se svladale osnove d3.js biblioteke, vježbom mogu se stvoriti jako zanimljive vizualizacije. Promatranjem grafova u ovom radu može se razmjerno dobro zaključiti koja informacija se pokušava prenijeti. Naravno, vizualizacije imaju mnogo prostora za poboljšanja, poput kreiranja responzivnog dizajna te dodavanja dodatnih interakcija sa korisnikom.

5. LITERATURA

[1] HTML For Beginners The Easy Way: Start Learning HTML & CSS Today, dostupno na:

<https://html.com/> Datum zadnje posjete: 17.6.2022.

[2] J. Job, Č. Livada, Priručnik za laboratorijske vježbe iz kolegija Vizualizacija podataka

[3] W3Schools What is CSS?, dostupno na:

https://www.w3schools.com/whatis/whatis_css.asp Datum zadnje posjete: 17.06.2022.

[4] Bootstrap, Get started with Bootstrap, dostupno na:

<https://getbootstrap.com/docs/5.2/getting-started/introduction/> Datum zadnje posjete: 17.06.2022.

[5] Wikipedija, JavaScript, dostupno na: [JavaScript – Wikipedija \(wikipedia.org\)](https://en.wikipedia.org/wiki/JavaScript),

Datum zadnje posjete: 17.6.2022.

[6] D3js, Data-Driven Document, dostupno na: [D3.js - Data-Driven Documents \(d3js.org\)](https://d3js.org/)

Datum zadnje posjete: 17.6.2022.

[7] Simple horizontal bar chart, dostupno na:

<https://bl.ocks.org/hrecht/f84012ee860cb4da66331f18d588eee3>

[8] Alangrafu, Radar Chart D3, dostupno na: <https://github.com/alangrafu/radar-chart-d3>

[9] Radar Cart D3 Config, dostupno na: <https://rawgit.com/tpreusse/radar-chart-d3/master/src/radar-chart.js>

6. PRILOZI

Link na GitHub repozitorij projekta: <https://github.com/Josip-Rizner/Wot-VP-Projekt>