

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3161

**Implementacija igre za
uvježbavanje kibernetičkog
ratovanja na strateškoj razini**

Filip Todorić

Zagreb, lipanj 2023.

DIPLOMSKI ZADATAK br. 3161

Pristupnik: **Filip Todoric (0036506267)**

Studij: Računarstvo

Profil: Računarska znanost

Mentor: izv. prof. dr. sc. Stjepan Groš

Zadatak: **Implementacija igre za uvježbavanje kibernetičkog ratovanja na strateškoj razini**

Opis zadatka:

Kibernetičke vježbe bitan su dio izgradnje sposobnosti obrane tvrtki. Svrha kibernetičkih vježbi je uvježbavanje timova koji brane neki informacijski sustav kako bi u slučaju stvarnih napada reagirali što bolje i na taj način umanjili štetu koja nastaje organizaciji. Kibernetičke vježbe mogu se provoditi na tehničkoj, taktičkoj, operativnoj i strateškoj razini. Za svaku razinu koriste se različite metode provođenja vježbi. Jedan od pristupa na operativnoj i strateškoj razini su igre koje podsjećaju na igre s kartama, kao što je monopoli. U sklopu ovog diplomskog rada potrebno je implementirati vježbu koja je opisana u doktorskoj disertaciji Haggman, Andreas. Cyber wargaming: finding, designing, and playing wargames for cyber security education. Diss. Royal Holloway, University of London, 2019. Vježba treba biti u formi Web aplikacije koja omogućava pristup više korisnika, praćenje uspjeha korisnika, spremanje stanja i slično. U diplomskom radu jasno naznačiti koji elementi sustava su nadograđeni, odnosno nisu opisani u disertaciji. Provesti eksperimente s aplikacijom. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 23. lipnja 2023.

SADRŽAJ

1. Uvod	1
1.1. Motivacija	1
1.2. Postojeće edukacijske igre i igre ratovanja	2
2. Pregled igre	4
2.1. Začetak igre	4
2.2. Reprzentacija igre	5
2.3. Sastavnice igre	5
2.3.1. Entiteti	5
2.3.2. Potezi i akcije	9
2.3.3. Sredstva	13
2.3.4. Događaji	14
2.3.5. Ciljevi i kraj igre	14
3. Implementacija web aplikacije	17
3.1. Arhitektura i korištene tehnologije	17
3.2. Backend aplikacija	19
3.2.1. Moduli, kontroleri, servisi i repozitoriji	19
3.2.2. Shema baze podataka	20
3.2.3. REST API	22
3.2.4. Autentikacija	27
3.2.5. WebSocket protokol	28
3.2.6. Obavljanje akcija i završavanje poteza	31
3.2.7. Aktiviranje efekata sredstava i karti događaja	32
3.2.8. Zapisnik igre	32
3.3. Aplikacija klijenta	33
3.3.1. Stranice aplikacije	33
3.3.2. WebSocket komunikacija	35

3.3.3. Interaktivnost i korisničko iskustvo	37
3.4. Daljnja poboljšanja	38
4. Testiranje aplikacije	41
4.1. Korisničko iskustvo	41
4.2. Iskustva odigranih igara	42
4.3. Nedostatci online igara	42
5. Zaključak	43
Literatura	44

1. Uvod

U ovom radu objasnit će se motivacija i proces razvoja igre za uvježbavanje kibernetičkog ratovanja. U uvodnom poglavlju opisat će se inicijalno razmatranje i trenutno stanje igara kibernetičkog ratovanja i njihove svrhe. U 2. poglavlju opisat će se elementi i pravila obrađene igre, kao i njezin cilj. U 3. poglavlju opisat će se metodologija i arhitektura praktičnog ostvarenja igre u obliku web aplikacije. U 4. poglavlju razmotrit će se početni dojam i iskustva korištenja aplikacije i igranja igre.

1.1. Motivacija

Razvojem računalne tehnologije i računalne povezanosti u svijetu, održavanje informatičke sigurnosti sustava zahtijeva sve više truda. Porast kompleksnosti u računalnom svijetu, također, povećava i količinu potrebnog znanja da bi se vršila dovoljno dobra zaštita sustava. Osim individualnih prijetnji na sustave, postoje i opasnosti na nacionalnim razinama, točnije kibernetičkom ratovanju. Preuzimanje kontrole i oštećivanja kritičnih sustava na nacionalnoj razini vrijedan je korak u dobivanju nadmoći i diktiranju uvjeta [1]. Posjedovanje specijaliziranog tima za protekciju informatičkih sustava unutar pojedinih kompanija ili ostalih tijela na višim razinama omogućuje smanjivanje ili prevenciju štete od bilo kakvih vanjskih učinaka i napadača. Kibernetika nije samo ograničena na informatički svijet, ali se obrađena igra i aplikacija u ovom radu posvećuju upravo njemu. Također treba naglasiti da je u kibernetičkom ratovanju bitna stavka i povratna informacija rezultata djelovanja da se sljedeći postupci pokušaju poboljšati. Tako se pokušavaju pronaći i primijeniti najbolje strategije za ostvarenje cilja zaštite ili napada.

U cilju uvježbavanja dinamike kibernetičkog ratovanja, ovaj rad će razmotriti to u obliku igre. Simulacijske igre korištene su već neko vrijeme u svrhu podučavanja korisnika [2]. Pažljiva gamifikacija određenog područja, s pomoću stvaranja znatiželje i zadavanja izazova, potiče korisnika da kritički razmišlja i uči najbolje strategije koje bi ga vodile ultimativnom cilju, pobjedi [7]. Naravno, igra kao takva mora sadržava-

vati kvalitetne elemente ciljnog područja i točke učenja da potakne znatiželju igrača i njegovo dugoročno zadržavanje.

Osim samog poticanja na učenje, koncept igre je vrlo sličan konfliktima u stvarnom svijetu. Konflikt u kojima se dvije strane nadmeću i u kojima koriste razne strategije, je prikladna percepcija kibernetičkog ratovanja u stvarnom svijetu [4].

1.2. Postojeće edukacijske igre i igre ratovanja

Postoje brojne igre na ploči koje imaju za cilj simulirati upravo ratovanje kibernetičke naravi. Međutim, skoro se sve postojeće igre igraju na ploči. Sljedeće izdvojene igre su dostupne u računalnom obliku.

- OverTheWire
- CyberCIEGE (2014.)
- Cyber Wargame (2020.)

OverTheWire je edukacijska igra koja trenira vještine i koncepte računalne sigurnosti. Razvijana je od strane skupine dobrovoljnih programera tijekom duljeg niza godina [6]. Primjer je igre koja se ne može u potpunosti klasificirati kao igra ratovanja, jer se sastoji od skupine izazova i nivoa u kojima se od igrača traži da rukuje s Linux OS i Bash komandnim jezikom. Često takve igre sadrže zadatke nalik *CTF* (eng. capture the flag) natjecanjima. Postoje brojne edukacijske igre slične *OverTheWire* koje su tehničke razine, a takva vrsta igre se neće razmatrati u ovom radu.

CyberCiege je edukacijska igra s interaktivnom grafikom, koja služi za uvježbavanja koncepta računalne i web sigurnosti [5]. Igra sadržava računalne sustave i mreže koje korisnik pokušava održavati i štiti. U ovoj igri nema elemenata strategije, već sadrži seriju interaktivnih edukacijskih izazova također tehničke i taktičke naravi.

Cyber Wargame je kartaška igra, dostupna i u obliku online aplikacije, koja, također, uči koncepte sigurnosti i zaštite sustava. Razvijena od strane programerske kuće *GamePartners* 2020. godine, u suradnji s tvrtkom *Holiseum*, koja specijalizira u području savjetovanja, usluga i obuke kibernetičke sigurnosti [3]. Igra obučava igrače o povjerljivosti podataka, ranjivosti sustava na višoj razini, i sličnija je igri koju ćemo razmatrati od prijašnje navedenih igara. Međutim, i dalje joj nedostaje faktor strategije i ratovanja gdje se traže različiti putovi kroz igru i na taj način uče korisnika različitim načinima upravljanja sustava, resursa i podataka.

U ovom se radu želi predstaviti igra u obliku web aplikacije koja sadrži sve elemente i pravila obrađene igre na ploči. Web aplikacijom se želi postići veća pristupačnost, veći broj igrača i olakšati akcije korisnicima. Također, prednost igre kao aplikacije je da se mogu automatizirati određeni procesi. Rezultat razvoja aplikacije je i dostupni *REST API* (eng. representational state transfer application programming interface) koji je pogodan, osim za zahtjeve klijentskih preglednika, i za prihvatanje zahtjeva raznih skripti i dubinsku analizu podataka. Tako, različiti modeli mogu analizom podataka olakšati pronalazak pravih strategija i dati uvid u određeni način razmišljanja. Osim API-ja, aplikacija poslužitelja također sadrži i rukovanje konekcijom *WebSocket* protokola za komunikaciju u stvarnom vremenu, što će isto biti opisano u radu.

2. Pregled igre

2.1. Začetak igre

U veljači 2019., Andreas Haggman je, u svojoj doktorskoj disertaciji "Cyber Wargaming: Finding, Designing, and Playing Wargames for Cyber Security Education", na sveučilištu u Londonu, opisao svoju novu igru kibernetičkog ratovanja [4]. U radu su objašnjeni motivacija, ključni pojmovi o igrama ratovanja, elementi novonastale igre i njezinih pravila. Igru je dizajnirao u svrhu edukacije u području kibernetičkog ratovanja. U radu se pomno istražuje prednosti igara ratovanja i posebno se obraća pozornost na spontanost i razigranost kao vrijedne aspekte igre koji privlače igrače. Izražava se da igrači vole dovoljno malu količinu nepredvidivosti i kaosa u smislu razigranosti. U izradi igre, navode se i limitacije igara ratovanja koji potencijalno narušavaju ili edukaciju igrača ili kvalitetu reprezentacije igre:

- *apstrakcija* - žrtvovanje apstrakcije radi smanjenja kompleksnosti analize igre i opsežnosti pravila
- *protivništvo* - oslanjanje na igrače koji igraju "antagonista" da dovoljno dobro simuliraju napadača
- *ishod igre* - prevelika pozornost igrača na ishod uzrokovana kompetitivnošću, zanemarujući suptilnije točke učenja kroz igru
- *politika reprezentacije* - borbeni "mi ili oni" način razmišljanja

Nadalje, osim navedenih faktora igara ratovanja, određuju se i najbitniji elementi koji dobro održavaju kibernetički prostor u stvarnom svijetu. U stvaranju pravila se vodi pažnja o konfliktu realističnosti igre i opsežnosti pravila. Navodi se da realističnost igre povećava broj parametara na kojima se igra temelji i tako povećava opsežnost pravila, a samim time zadaje i igračima sve veći izazov za igranje, što potencijalno smanjuje njihovu angažiranost.

Ovu igru je i implementirao Stjepan Marijanović u sklopu diplomskog rada, na

Fakultetu elektrotehnike i računarstva, 2023. godine. Njegov rad služi kao temelj i začetak web aplikacije. Kroz ovaj rad će se iskoristiti određene odluke i arhitekturna rješenja postojeće aplikacije. Odluke ponajviše po pitanju neodređenosti i nejednoznačnosti pisanih pravila originalnoga rada A. Haggmana.

2.2. Reprezentacija igre

U radu je, kao jedna instanca igre, predstavljen prostor kibernetičkog ratovanja s dvije protivničke strane. Svaka strana predstavlja povezanost na nacionalnoj razini između svojih entiteta. Po uzoru na razvijenu strategiju nacionalne sigurnosti Ujedinjenog Kraljevstva, jedna strana predstavlja upravo UK, koja je prikazana kao strana pod prijetnjom od kibernetičkih napada od strane inozemnih napadača. U zadnjoj inačici igre, svakoj su se strani pridružile nacije iz stvarnog svijeta, pa se tako i protivničkom timu pridružila Rusija kao aktor prijetnje. Pod takvim ulogama, igrači su angažiraniji. Kao period odvijanja sukoba dviju strana, odabran je period od jedne godine, 2020. godine. Jedan mjesec predstavlja period u kojemu obje strane obavljaju svoje strateške akcije.

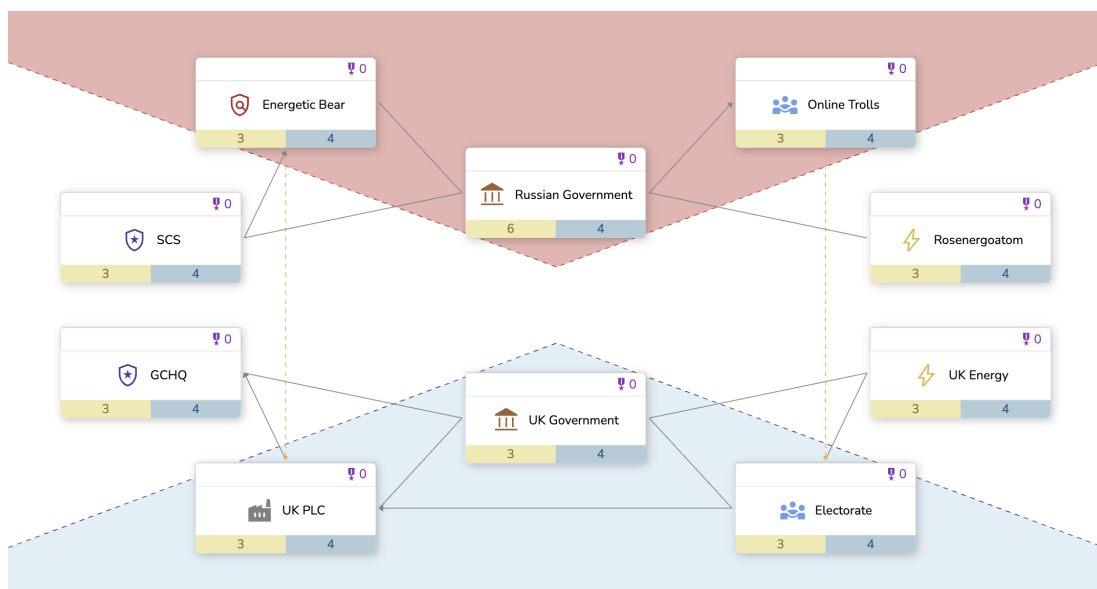
2.3. Sastavnice igre

Igra se sastoji od ploče na kojoj su raspoređene dvije strane sukoba zajedno sa svojim pripadnim entitetima. Ploča predstavlja prostor kibernetičkog ratovanja i prikazana je na slici 2.1. A raspored entiteta i njihove veze predstavljaju odnose sastavnih tijela jedne strane.

2.3.1. Entiteti

Svaki entitet je samostalno odlučivačko društvo koje posjeduje resurse, zdravlje i svrhu odnosno cilj. Tako se entitetima pridružuju brojači resursa, vitalnosti (zdravlja) i pobjedničkih bodova. Entiteti su sljedeći:

- Ujedinjeno Kraljevstvo:
 - vlada Ujedinjenog Kraljevstva
 - *Electorate*
 - *UK PLC*
 - *UK Energy*
 - *GCHQ* (eng. Government Communications Headquarters)



Slika 2.1: Ploča igre

- Rusija:

- vlada Rusije
- *Online Trolls*
- *Energetic Bear*
- *Rosenergoatom*
- SCS (eng. Special Communications Service)

Resurs entiteta predstavlja novac, radnu snagu, opremu, populaciju ovisno o tome što entitet predstavlja. Upravo su te stvari koje služe za trgovinu, kupovinu, distribuciju i sl. Pa će tako u igri resurs služiti za obavljanje različitih akcija na njegov trošak.

Vitalnost entiteta predstavlja blagostanje, zadovoljstvo, napredak nekog entiteta. Također ovisi o vrsti sustava koju entitet predstavlja. O vitalnosti treba razmišljati kad se govori o zadovoljstvu opće populacije za vrijeme glasanja, količini štete koju neki informatički ili fizički sustav može primiti i sl. Entitet će paziti na vlastitu vitalnost da ne padne na vrijednost 0 i da ovisno o ciljevima poprimi što veću vrijednost, što reprezentira prosperitetnost. Slično kao i za resurs, na igraču je da promisli što točno ova vrijednost predstavlja entitetu kojim upravlja.

Brojevnje vrijednosti resursa i vitalnosti, u implementaciji web aplikacije, moraju biti definirane kao cjelobrojne vrijednosti ili vrijednosti s pomičnom točkom. Dok ne postoji mogućnost u ikojem dijelu igre da resurs bude sveden na decimalni broj, vrijednost vitalnosti može preciznim računanjem biti svedena na decimalu. Mogućnost zaokruživanja je opisana u poglavlju 2.3.2. kada se opisuje akcija napada.

Svaki entitet ima svoju ultimativnu svrhu i cilj unutar kibernetičkog prostora za pomaganje osvajanje konflikta ili očuvanje svojeg cilja. Takav cilj se ostvaruje uz pomoć pobjedničkih bodova. Svakom entitetu posebno je pridružena lista ciljeva ili zahtjeva koje trebaju zadovoljiti kako bi maksimizirali broj svojih pobjedničkih bodova. Tako pobjednički bodovi jednog entiteta doprinose ukupnom zbroju bodova pripadne strane i pobjeda se određuje na temelju pobjedničkih bodova.

Na slici 2.2 prikazan je primjer entiteta zajedno s vrijednostima resursa, vitalnosti i broj pobjedničkih bodova. Odabir boja i njihove reprezentacije bit će raspravljene u poglavlju 3.



Slika 2.2: Količina resursa je označena žutom bojom, vitalnost plavom i pobjednički bodovi ljubičastom

Entitet vlade je centralni entitet svake strane, vlada Ujedinjenog Kraljevstva i ruska vlada. Taj entitet predstavlja pokretača cijelog sustava na nacionalnoj razini. To je također točka u kojoj se uvode novi resursi u igru i vlada je zaslužna za distribuiranje tih resursa.

Vlada UK predstavlja vladajuće tijelo i kao takvo brine se o popularnosti među svojim biračima, Electorate. Također, u sukobu s ruskom stranom promoviraju stav kojim se služe za suzbijanje ruske reputacije i tako prikupljaju istomišljenike. Entitet dobiva 1 pobjednički bod svaki mjesec ukoliko Electorate entitet završi s 4 ili više resursa. Nadalje, po završetku igre još dodatnih 5 bodova, ako ruska vlada završi s manje vitalnosti nego na početku igre.

Electorate je entitet UK strane koji predstavlja opću populaciju Ujedinjenog Kraljevstva s glasačkom moći. Resursi predstavljaju bogatstvo i vlasništvo opće populacije, a cilj je entiteta da se spriječi oduzimanje njihova vlasništva. Tako svakim slanjem resursa drugim entitetima Electorate gubi 1 pobjednički bod.

UK PLC predstavlja industriju Ujedinjenog Kraljevstva kao i poslovne interese UK strane. Tako se resursi tog entiteta odnose na koncepte poput novca, industrijske opreme, poslovne moći i sl. Kako je *Brexit* proces prisutan u periodu odvijanja ove igre

odnosno ratovanja, ovaj entitet mora očekivati moguće nepredvidive promjene. Stoga, cilj je prikupiti što više resursa i na taj način osigurati nastavak sigurnog poslovanja i industrijskog napretka. Po završetku travnja dobiva 2 boda ako posjeduje 3 ili više resursa, po završetku kolovoza 3 boda ako posjeduje 6 ili više resursa i po završetku prosinca 4 pobjednička boda ako posjeduje 9 ili više resursa.

UK Energy predstavlja proizvodnju energije, transport, telekomunikaciju i financije na nacionalnoj razini. Takav sustav je također izložen u prostoru kibernetičkog ratovanja i nastoji očuvati svoju vitalnost. Vitalnost stoga predstavlja količinu opskrbljive energije i kvalitetu infrastrukture. Cilj je povećavati kapacitet energije i unaprjeđivati infrastrukturu. Entitet dobiva 2 pobjednička boda za 6 ili više vitalnosti po završetku lipnja i 3 boda za 9 ili više po završetku prosinca.

GCHQ predstavlja obavještajnu službu koja služi u obradi povjerljivih informacija i zaštiti ključnih i povjerljivih sustava u području računalne sigurnosti. Vitalnost entiteta predstavlja mjeru količine eksperta koji su zaposleni u odjelima i koji obavljaju svoje dužnosti. Stoga, cilj je konzistentno povećavati vitalnost u svrhu poboljšanja usluga obavještajne službe. Entitet dobiva pobjedničke bodove na temelju uzastopnosti povećavanja vitalnosti po kvartalima godine. Tako će za jedan uzastopni kvartal godine dobiti 1 pobjednički bod, za dva uzastopna kvartala 3 boda, za tri uzastopna kvartala 5 boda i za sva četiri uzastopna kvartala će dobiti 7 pobjedničkih bodova. Dobitak nije kumulativan.

Ruska vlada kao opozicija UK vladi u sukobu, također je vladajuće tijelo, ali u puno centraliziranijem sustavu. Također, ruska vlada koristi od toga da raspolaže s dodatnim resursima u tako centraliziranoj vladavini. Tako entitet dobiva 1 pobjednički bod svaki mjesec kad završi s 3 ili više resursa. U cilju održavanja dovoljne kontrole nad odgovarajućim tijelima, ruska vlada gubi 1 pobjednički bod za svaki napad od strane Online Trolls entiteta u kojeg se utrošilo 3 ili 4 resursa, a gubi 2 boda za napade u kojih se utrošilo 5 ili 6 resursa.

Online Trolls entitet predstavlja manji uzorak populacije ruske strane, za razliku od Electorate entiteta. Online Trolls entitet je karakteriziran aktivnošću na internetu gdje provociraju i naslađivaju se općim korisnicima na vlastito zadovoljstvo. Na taj način čine štetu ili u psihološkom smislu ili u smislu da su izvor različitih lažnih informacija. Nadalje, provokacije manjih razmjera se smatraju dosadnim i nedovoljnim za njihove potrebe pa je njihov cilj izvršavati napade prikladnih razmjera. Stoga dobivaju 4 pobjednička boda svakim napadom u kojega se utroši 3 ili više resursa.

Energetic Bear entitet je pandem UK PLC poslovnom dijelu UK strane. Njihovo poslovanje se svodi na napade kibernetičke naravi odnosno "hakaranju". Prikupljanjem

povjerljivih informacija i nadmoći pomoću cyber napada šire svoje poslovanje i time pridobivaju nove resurse i ljude. Vitalnost dakle predstavlja napredak poslovanja i cilja je povećavati. Tako nakon svaka 4 mjeseca dobiva 1, 3 ili 5 pobjednička boda, redom, ako po završetku svakog perioda imaju više vitalnosti nego prije početka tog perioda.

Rosenergoatom je entitet koji predstavlja operatera i regulatora nuklearne energije ruske strane. Odnos Rosenergoatoma i ruske vlade je bliskiji od odnosa UK Energy entiteta s UK vladom zbog veće centralizacije vodstva ruske strane. Isto kao i UK Energy, Rosenergoatom je u cilju povećavati kapacitet energije i opskrbu infrastrukture, pa za cilj imaju povećavanje svoje vitalnosti uzastopno po kvartalima. Za jedan uzastopni kvartal godine entitet dobiva 1 pobjednički bod, za dva uzastopna kvartala 3 boda, za tri uzastopna kvartala 5 boda i za sva 4 uzastopna kvartala dobiva 7 pobjedničkih bodova. Dobitak nije kumulativan.

SCS predstavlja obavještajnu službu ruske strane. Kao pandem GCHQ, vode brigu o obrani od kibernetičkih napada, a orijentirani su tome da prikupljaju sredstva koje mogu iskoristiti za dobivanje prednosti u daljnjem sukobu. Tako se cilj ovog entiteta svodi na prikupljanje dovoljno sredstava za efektivno djelovanje protiv GCHQ entiteta. Kao cilj imaju da ruska strana svaki mjesec posjeduje više napadačkih sredstava nego što UK strana posjeduje obrambenih i time dobivaju 2 pobjednička boda.

2.3.2. Potezi i akcije

U periodu od jedne godine, svaki mjesec entiteti svake strane obavljaju svoje akcije. Kada su obje strane obavile svoje akcije, mjesec završava i kreće novi. Kao dodatni motivator igračima, i sprječavanja neodlučnosti i otezanja, određeno je vremensko ograničenje svakog tima unutar jednog mjeseca. Ograničenje je po zadanim postavkama postavljeno na tri minute kao opisano u radu po zadnjoj inačici igre, a u implementaciji aplikacije se može i mijenjati, ovisno o potrebi. Ako igrač ne obavi akciju unutar vremenskog ograničenja, taj mjesec ne čini ništa. U jednoj godini dostupno je dvanaest poteza svakog tima. Za održavanje cirkulacije resursa, na početku svakog poteza entitetu vlade je dodijeljeno 3 resursa.

Akcije predstavljaju procese koji se odvijaju unutar pripadnih tijela i svako tijelo očekuje rezultate obavljenih akcija. Svaki potez entitet obavlja jednu od pet dostupnih akcija:

- distribucija
- revitalizacija

- napad
- pristup crnom tržištu
- apstinencija

Distribuiranjem resursa entitet vlade opskrbljuje ostatak sustava, a ostali entiteti razmjenjuju resurse međusobno po potrebi. Na taj način se ostvaruje element timske suradnje u obliku trgovine ili potpore. Distribucija se odvija po vezama između entiteta. Entiteti su semantički povezani jednosmjernim, dvosmjernim vezama ili nisu povezani uopće. Maksimalan broj resursa koji se može prenijeti je pet i ne postoji ograničenje koliko entitet može posjedovati resursa.

Revitalizacija je proces obnove zdravlja entiteta, tj. povećavanja vitalnosti uz trošak resursa. Entitet nema gornjeg ograničenja na vitalnost, a trošak revitalizacije se računa po tablici 2.1.

Vitalnost	Trošak resursa
1	1
2	2
3	4
4	5
5	6
6	7

Tablica 2.1: Konverzija revitalizacije

Akcijom napadanja jedan entitet radi štetu drugom entitetu što mu se odražava smanjenjem vitalnosti. Akcija napada dostupna je onim entitetima koji imaju otvorene vektore napada koji se označavaju jednosmjernim strelicama. Narančastom bojom se označavaju vektori ruske strane, a ljubičastom bojom vektori napada UK strane. Inicijalno su dostupna dva vektora napada entitetima Online Trolls i Energetic Bear kojima napadaju Electorate i UK PLC, redom. Tijekom igre, mogu se otvarati dodatni vektori, a oni su sljedeći: vektor napada UK vlade na rusku vladu, GCHQ na Rosenergoatom i SCS na entitet UK Energy. Primjeri vektori se mogu vidjeti na slici 2.1. Na slici su prikazani inicijalno dostupni vektori entiteta Online Trolls i Energetic Bear.

Igrač prije svakog napada mora odrediti koliko resursa želi utrošiti u napad. Veće ulaganje resursa u napad povećava vjerojatnost za nanošenje veće štete. U napad se može uložiti minimalno 1 i maksimalno 6 resursa. Element nepredvidivosti u mehanici napadanja dodan je pomoću šesterostrane igraće kocke. Prije izvršavanja napada igrač

baca kocku i rezultat bačene kocke (1-6) utječe na ishod napada. Rezultat napada u ovisnosti o uložnim resursima i rezultatu kocke određuje se tablicom rezolucije napada 2.2.

Uloženo resursa	Rezultat kocke					
	1	2	3	4	5	6
1	0	1	1	1	1	2
2	0	1	1	1	2	2
3	-1	0	1	2	2	3
4	-1	0	1	2	3	4
5	-2	-1	2	3	3	4
6	-2	-1	0	3	5	6

Tablica 2.2: Tablica rezolucije napada

U slučaju negativnog ishoda napada napadač nanosi štetu vlastitoj vitalnosti. Također, negativan ishod napada nosi sa sobom negativne posljedice i određuju se u tablici atribucije 2.3.

Negativne posljedice neuspješnog napada održavaju se u gubitku vremena, resursa, dobivanje negativne reputacije i sl. U igri se time pridodaje pažljivost upućivanju svakog napada.

Nadalje, svaki napad prouzročuje i rezidualnu štetu svim entitetima koji su povezani s entitetom koji je primio štetu. Rezidualna šteta se računa u omjeru 1:2. Rezidualna šteta podsjeća na činjenicu da su entiteti međusobno ovisni do neke razine i koriste resurse, proizvode i usluge drugih tijela u sustavu, a zbog toga će i osjetiti efekte oštećenja entiteta o kojima su ovisni. Rezidualna šteta se također obračunava i za neuspješne napade.

Zbog specifičnosti omjera, postoji vjerojatnost da bude prouzročena šteta neparne vrijednosti. Rezidualna šteta će tada biti nužno decimalni broj. U svojem radu, A. Haggman resurs i vitalnost predstavlja kockicama u igri na ploči. Stoga se implicitno govori o cjelobrojnim vrijednostima. Web aplikacija u svojoj naravi dopušta decimalni zapis vrijednosti i pošto zaokruživanje nije specifično određeno i nije specificirana bitnost zaokruživanja, u prvoj se inačici računalo s cjelobrojnim vrijednostima dok se u konačnoj inačici sva šteta zaokružuje na prvu manju cjelobrojnu vrijednost.

Napad koji prouzroči smanjenje vitalnosti nekog entiteta do 0 završava igru nakon što se primijene svi efekti napada. Dodatno, ako se radi o uspješnom napadu, napadač dobiva 10 pobjedničkih bodova.

Napadački entitet	Nivo atribucije	
	-1	-2
Energetic Bear	UK dobiva "Software Update" sredstvo.	UK dobiva "Software Update" i "Recovery Management" sredstva.
Online Trolls	UK dobiva "Education" sredstvo.	UK dobiva "Education" sredstvo, Online Trolls ne može napadati 2 poteza.
SCS	UK dobiva "Software Update" sredstvo, SCS ne može stavljati ponude na crnom tržištu dva poteza.	UK može otvoriti vektor napada entiteta GCHQ na Rosenergoatom ili entiteta UK vlade na rusku vladu bez ikakvog troška.
GCHQ	GCHQ ne može napadati 2 poteza.	GCHQ ne može obavljati nikakve akcije 2 poteza, UK vlada gubi 1 vitalnost.
UK vlada	Ruska strana dobiva "Bargaining Chip" sredstvo.	Ruska strana dobiva "Bargaining Chip" sredstvo, UK vlada gubi dodatnih 2 vitalnosti i 2 resursa.

Tablica 2.3: Tablica atribucije napada

Pristupati crnom tržištu mogu samo obavještajne službe svake strane. Tržište je mjesto na kojemu igrači trebaju pažljivo ulagati svoje resurse da dobiju sredstva koja će pomoći vlastitim timovima na razne načine. Na tržištu su izlistana sva dostupna sredstva i tržište je koncipirano u obliku dražbe. Svaki potez, GCHQ i SCS stavljaju ponude na dostupna sredstva, tj. ulažu resurse. Onaj entitet koji je ponudio najviše, i ako protivnički entitet nije povećao ponudu u neposredno sljedećem potezu, osvaja sredstvo za svoj tim u svojem sljedećem potezu. Dražba je vrste "all-pay", značenja da su jednom uloženi resursi potrošeni i ne vraćaju se, čak i ako protivnički tim završi s najvećom ponudom i osvoji sredstvo. U jednom potezu se mogu dati ponude za više sredstava.

Nije određeno koliko sredstava i koliko često se sredstva dostavljaju na crno tržište. S. Marijanović je u svojem radu, također, uvidio to i dao svoje mišljenje o količini. Odlučeno je da se svaki mjesec stavlja jedno sredstvo na crno tržište. Moguće da su u originalnoj igri sva sredstva dostavljena na tržište od samog početka, međutim to ne bi predstavljalo oskudnost sredstava i njihovu nabavku. Tako će se dostavljati jedno sredstvo svaki mjesec, a u budućnosti ta količina može biti podložna i konfiguraciji u samoj igri.

Jednom osvojeno sredstvo ide u inventar tima i može se čuvati za kasniju aktivaciju efekta. Neovisno o tipu i efektu sredstva, svako sredstvo na crnom tržištu može biti osvojeno od bilo koje strane.

2.3.3. Sredstva

Sredstvo predstavlja dodatnu opremu, nadogradnju ili poboljšanje sustava koja dodaje element razvoja kibernetičkih sposobnosti jedne strane. Sredstva se osvajaju na dražbi na crnom tržištu i svako sredstvo ima jedinstven efekt. Osim pažljive akvizicije sredstava s crnog tržišta, tim također pokušava pronaći najbolje strategije upotrebe istih, ovisno o njihovim efektima. Postoje devet jedinstvenih sredstava koji se mogu osvojiti na dražbi. Nekih sredstava ima više od drugih, a ukupno ih ima šesnaest.

Popis dostupnih sredstava dan je u tablicama 2.4 i 2.5. Sredstva su podijeljena u dvije kategorije: ofenzivna i defenzivna ovisno o vrsti efekta, i propisane su minimalne ponude za crno tržište. Minimalne ponude predstavljaju vrijednost potencijalne nadogradnje ili korisnosti koje će tim imati od osvojenog sredstva.

Naziv sredstva	Efekt sredstva	Minimalna ponuda	Količina sredstava
Attack Vector	Otvara se jedan od sljedećih vektora napada: GCHQ na Rosenergoatom, SCS na UK Energy ili UK vlada na rusku vladu.	5 resursa	2
Stuxnet 2.0	Direktni napadi od GCHQ ili SCS entiteta čine dvostruko veću štetu. Jednokratna uporaba.	4 resursa	3
Ransomware	Direktni uspješni napadi entiteta Online Trolls ili Energetic Bear paraliziraju metu na 2 poteza, osim ako meta ne plati 2 resursa napadaču. Jednokratna uporaba.	3 resursa	2

Tablica 2.4: Popis ofenzivnih sredstava

2.3.4. Događaji

Događaji simuliraju nepredvidive okolnosti na koje nailaze pojedinačna tijela nekog sustava i uvode dodatnu razinu nepredvidivosti u igru. Tako igrači trebaju pokušati uračunati moguće gubitke ili dobitke. Događaji za igru su predstavljeni u obliku igračih karata. Popis karata događaja i njihovih efekata dan je u tablici 2.6. Špil karata za igru se sastoji od 16 karata, u kojem je osam od njih "Uneventful Month" karta i predstavlja nepostojanje događaja. Većinska vjerojatnost da nema nikakvih izvanrednih događaja je realna reprezentacija stvarnog svijeta jer su sustavi većinom stabilni i nova otkrića su rjeđa pojava. Špil se promiješa prije početka igre i svaki mjesec se vuče jedna karta. Efekt karte se zatim odmah primijeni.

2.3.5. Ciljevi i kraj igre

Igra može biti završena na dva načina: istekom godine ili padom vitalnost nekog entiteta na 0. Po završetku igre, svakom timu se pridružuje onoliko pobjedničkih bodova koliko zajedno imaju njegovi entiteti i pobjednikom se proglašava tim s većim brojem bodova. Stoga, pobjednički bodovi su ti koji odlučuju igru a ciljevi, s kojima se zarađuju bodovi, su od najveće važnosti. Može se isto napomenuti da iako vitalnost jednog entiteta u timu padne na 0, taj tim i dalje može pobijediti. Ta činjenica rezultira time

Naziv sredstva	Efekt sredstva	Minimalna ponuda	Količina sredstava
Education	Electorate dobiva dvostruko manju štetu sljedeća 3 poteza.	3 resursa	1
Recovery Management	Na kraju poteza, ako je UK PLC pretrpio ikakvu štetu, dobiva 1 vitalnost.	4 resursa	1
Software Update	Entitet UK PLC, UK Energy ili Rosenergoatom postaje otporan na direktni napad sljedeća 2 poteza.	2 resursa	2
Bargaining Chip	Ruska vlada prima dvostruko manje štete sljedeća 3 poteza.	3 resursa	1
Network Policy	Entitet na kojemu se primijeni efekt postaje otporan na rezidualnu štetu, ali svaki potez može slati i primati samo 2 resursa.	2 resursa	2

Tablica 2.5: Popis defenzivnih sredstava

da igrači neće napadati da što prije spuste vitalnost drugog entiteta do 0, već uvodi dodatan nivo pažljivosti pri napadanju.

Imajući na umu dosad navedena pravila, strategija igre je dakle maksimiziranje broja pobjedničkih bodova svojeg tima. Uz mehaniku akcija, efekata sredstava i događaja suigrači moraju dogovorom otkriti najbolju strategiju pritom pokušavajući maksimizirati vlastite interese odnosno ciljeve. Po načelu kibernetike gdje se akcija korigira s povratnom vezom dobivenog iskustva, igrači će uvidom u posljedice svake akcije s vremenom poboljšavati ili otkrivati nove strategije. Naravno, takva mehanika igre će omogućiti i edukacijsku komponentu za igrače.

Događaj	Opis efekta
Uneventful Month	Ništa se izvanredno nije dogodilo ovaj mjesec, nastavi igru.
Nuclear Meltdown	Hinkley Point nuklearni reaktor pretrpio je štetu od manje ali značajnije tehničke greške. UK Energy gubi 1 vitalnost.
Clumsy Civil Servant	Javni službenik je ostavio laptop s osjetljivim podacima u vlaku. Electorate gubi 1 vitalnost. UK vlada gubi 2 resursa.
Software Update	UK vlada nalaže da sve tvrtke prijeđu na najjažurnije verzije operacijskih sustava. UK PLC gubi 2 resursa.
Banking Error	Greška sa zaokruživanjem brojevnih vrijednosti u računalnom sustavu banke Bank of England je poremetila protokole transakcija. Entiteti UK-a ne mogu slati resurse ovaj mjesec.
Embargoed	Avanturizam vanjske politike Rusije je rezultirao internacionalnim embargom oružja. SCS ne može stavljati ponude na sredstva crnog tržišta ovaj mjesec.
Lax OpSec	Radnik ministarstva unutarnje politike je priključio zaraženi USB drive. Ruska vlada gubi 1 vitalnost i 1 resurs.
People's Revolt	U javnosti se organizirao se štrajk protiv cenzure interneta. Ruska vlada ne dobiva nove resurse ovaj mjesec.
Quantum Breakthrough	Google je inkorporirao kvantna računala za sve svoje servise i uređaje. Svi entiteti dobivaju 1 resurs i 1 vitalnost.

Tablica 2.6: Događaji

3. Implementacija web aplikacije

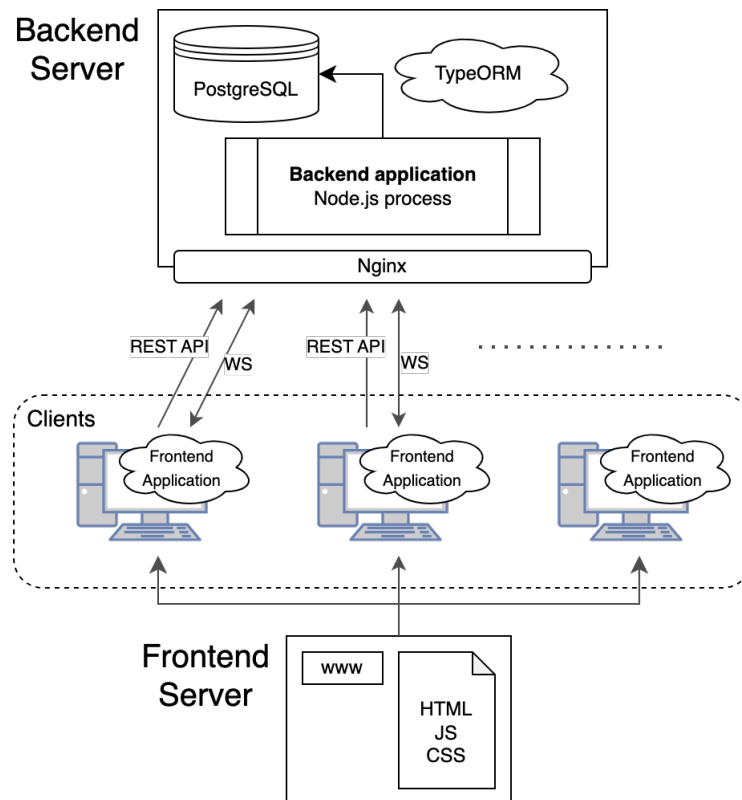
3.1. Arhitektura i korištene tehnologije

Za potrebe predložene igre, najprikladnija vrsta aplikacije je web aplikacija. Jednostavna je za izradu, pristupačna i praktična za igranje igara s više igrača. Arhitektura web aplikacije se može izvesti na razne načine. Način koji je odabran u ovom radu je jedan od modernijih vrsti arhitektura prikazana na slici 3.1. *Multiplayer* (eng. igra s više igrača) igra također zahtijeva određenu razinu komunikacije u stvarnom vremenu što će biti postignuto s protokolom WebSocket za uspostavu dvosmjerne veze klijenta s poslužiteljem. Na taj način poslužitelj neće čekati zahtjeve s klijenta, već će odašiljati poruke svim klijentima spojenih na točno određen *socket* (eng. mrežna priključnica).

Server sadržava instancu PostgreSQL baze koja će služiti za pohranu korisnika, igara, igrača, itd. Instanca baze podataka je enkapsulirana u kontejner pomoću *Docker* alata. Kontejneriziranje procesa omogućava njegovo pokretanje neovisno o okolini u kojoj se nalazi, što je pogodno svojstvo ako se ne želi konfigurirati okruženje baze podataka na poslužiteljskom operacijskom sustavu.

Zasebni proces tzv. "*backend*" aplikacija, prihvata zahtjeve klijenta, obrađuje ga, šalje upit bazi podataka i šalje nazad klijentu odgovor koji sadrži zatražene podatke iz baze podataka. Integriran je i posredni proces *Nginx*, koji služi za balansiranje zahtjeva, reverzno posredovanje, itd. Za ovu aplikaciju vrši obvezu reverznog posrednika. Posredstvo omogućuje enkapsuliranje backend procesa i izoliranje od potencijalno štetnih vanjskih utjecaja. Također, u našem slučaju služi i za primjenu *SSL* (eng. Secure Sockets Layer) enkripcije. SSL certifikat izdan od *CA* (eng. Certificate Authority) agencije služi kao vjerojodostojan indikator enkriptirane veze s klijentom. Upravo ovu sigurnosnu značajku navodi S. Marijanović kao potencijalno poboljšanje sigurnosti komunikacije klijenta i poslužitelja.

Backend aplikacija je *Node* proces i pisan je u programskom jeziku *JavaScript* (skr. JS). *Node.js* je popularno izvršno okruženje JavaScript jezika i često ga se koristi u izradi backend aplikacija. Okruženje je dovoljno dobro i po pitanju performansi za



Slika 3.1: Server - klijent arhitektura

opseg aplikacije.

Za oblikovanje sheme baze i rukovanje bazom pomoću *SQL* (eng. Structured Query Language) upita, korišten je *TypeORM*. *TypeORM* je JavaScript biblioteka koji služi za *ORM* (eng. Object-Relational Mapping). *ORM* je tehnika u kojoj se relacije tj. tablice relacijske baze pretvaraju u objekte prikladne za skladištenje na gomili u memoriji i objektno orijentirano korištenje. *TypeORM* podržava i uporabu tipiziranog JavaScript proširenja *TypeScript*.

"*Frontend*" aplikacija predstavlja vizualno interaktivno sučelje prema korisniku i preslikava korisnikove akcije u pozive prema backend serveru. Frontend aplikacija je također pisan u jeziku *TypeScript*. Za strukturiranu izradu frontend aplikacije korištena je *React JS* biblioteka verzije 18, a aplikacija je razvijena u radnom okviru *NextJS* verzije 13. Biblioteka *React* služi za izgradnju aplikacije pomoću komponenti i neke od korisnijih značajki su rukovanje stanjima komponenata i *hooks* metode. *NextJS* radni okvir pruža razne ugrađene funkcionalnost kao što su: *SSR* (eng. Server-side Rendering), *SSG* (eng. Static Site Generation), *routing*, itd.

3.2. Backend aplikacija

Backend aplikacija je zaslužna za posredovanje između baze podataka i zaprimljenih zahtjeva korisnika. Za izvršno okruženje koristi se Node.js verzije 16. Koristi se razvojno okruženje *NestJS* verzije 9 koje nudi predefiniranu strukturu aplikacije i ugrađene funkcionalnosti poput kontrolera, servisa i repozitorija. Za korištenje dodatnih JS biblioteka koristi se menadžer paketa *Yarn*.

Inicijalna struktura projekta stvorena je pomoću komandnog sučelja koje nudi Nest.

3.2.1. Moduli, kontroleri, servisi i repozitoriji

Razvoj aplikacije pratila je generalna arhitektura izvornog koda. Aplikacija je podijeljena u 6 glavnih modula. Svaki modul predstavlja nezavisnu strukturu u bazi, i sadrži vlastite klase i metode koje rukuju podacima iz baze. Moduli su:

- auth - autentikacija i organizacija korisnika
- players - operacije nad igračima
- teams - operacije nad timovima
- games - operacije nad igrama
- assets - operacije nad sredstvima igre
- event-cards - operacije nad sredstvima igre

Svaki modul kao cjelina sadrži svoj kontroler, servis i repozitorij.

U kontrolerima se definiraju klase čije metode služe za prihvaćanje zahtjeva na specifičnim rutama REST sučelja. Uz rute se definiraju vrste HTTPS poziva (GET, POST, DELETE), parametri koje ruta sadržava i *payload* (hrv. teret), dodatni podaci koji svaki zahtjev može sadržavati. U kontrolerima se također postavlja i validacija podataka. Točnije svaki dolazni zahtjev se mora analizirati i prihvatiti ako sadrži korektan format. Validacija osigurava da se ne dogode neželjene promjene u bazi podataka ili da korisnik namjerno pokuša narušiti integritet baze.

U servisima se nalaze klase čije metode obavljaju poslovnu logiku backend aplikacije, tj. u tim metodama se odvija obrada podataka i zahtjeva i dobivenih odgovora iz baze podataka. Kontroleri tako pri svakom zahtjevu pozivaju servise sa pročišćenom strukturom zahtjeva.

U repozitorijima se nalaze klase koje su instance baze, tj. točno one cjeline baze za koji je taj modul odgovoran. Tako metode tih klasa sadržavaju operacije nad bazom, generiranja SQL upita i dohvaćanje podataka. Pošto aplikacija koristi TypeORM, klase

repozitorija nasljeđuju *Repository* klasu TypeORM biblioteke, a upravo ta klasa služi za rukovanje bazom podataka. Servisi tako pozivaju metode klase repozitorija i nakon dobivanja odgovora obavljaju vlastite operacije.

3.2.2. Shema baze podataka

Osmišljena je prikladna shema baze podataka kako bi se pohranjivali podaci o korisnicima, igračima, timovima i detaljima svake instance igre. Na slici 3.2 prikazana je relacija koja služi za organizaciju korisnika. Za početne svrhe i efikasnost izgradnje *MVP* (eng. minimum viable product) rješenja, korisnicima se pridružuju najosnoviji podaci autentifikacije, korisničko ime i lozinka.

auth.User	
id	string
username	string
password	string

Slika 3.2: Relacija *User*

Na slici 3.3 prikazana je relacija *Player*. Relacija sadrži podatke specifične instanci igrača kao što su *resource*, *vitality* i *victoryPoints*. Također, relacija je asocirana relacijom *User* radi autenticiranja akcija igrača unutar igre. Bez takve protekcije, svaki klijent bi mogao slati zahtjeve, namjerno ili nenamjerno, i backend aplikacija ne bi mogla validirati zahtjev već bi ga samo prihvatila i obradila.

Svaki igrač posjeduje i svoj tim, koji se modelira relacijom prikazanoj na slici 3.4. Svaki tim ima mogućnost biti sastavljen od različitih igrača ali i od jedno te istog. U radu A. Haggmana nije navedena struktura tima, stoga je omogućeno proizvoljno popunjavanje igrača tima. Postoje pet različitih entiteta jedne strane, stoga pet dostupnih mjesta za igrače unutar jednog tima.

Zatim, relacija *Game*, prikazana na slici 3.5, modelira jednu instancu igre i pridružuju joj se identifikatori timova. Osim glavnih svojstava igre kao što su: status igre, aktivni igrači, ishod igre, itd., postoje i globalne zastave i varijable koje služe za praćenje raznosvrnih stanja ciljeva ili kondicija u igri poput: otvorenosti vektora napada, praćenje upotpunjenje GCHQ cilja "Recruitment Drive", itd. Zastavice su boolean vrijednosti koje govore o istinitosti neke tvrdnje (*isRussianGovernmentAttacked*), a

public.Player	
id	string
side	public.TeamSide
type	public.PlayerType
resource	number
vitality	number
victoryPoints	number
madeAction	public.GameAction
hasMadeBid	boolean
attackBanRemainingTurns	number
paralysisRemainingTurns	number
armor	number
armorDuration	number
hasSufferedAnyDamage	boolean
damageImmunityDuration	number
isSplashImmune	boolean
hasDoubleDamage	boolean
hasCyberInvestmentProgramme	boolean
hasRansomwareAttack	boolean
wasRansomwareAttacked	boolean
userId	string

Slika 3.3: Relacija Player

public.Team	
id	string
side	public.TeamSide
name	string
peoplePlayerId	string
industryPlayerId	string
governmentPlayerId	string
energyPlayerId	string
intelligencePlayerId	string
isEventCardRead	boolean
canTransferResource	boolean

Slika 3.4: Relacija Team

brojači su brojevne vrijednosti koji služe za praćenje preostalih broja poteza trajanja raznih efekta i kondicija.

Sredstva su modelirana zasebnom relacijom prikazanoj na slici 3.6. Sadrže podatke poput: tipa sredstva, opis efekta sredstva, minimalnu ponudu, itd. Također, zaslužna je za praćenje stanja ponuda na crnom tržištu. Jedan entitet relacije *Asset* vezan je za relaciju igre, točnije sadrži identifikator igre.

Kao i sredstvo, i karta događaja je modelirana vlastitom relacijom prikazanoj na slici 3.7. Za razliku od sredstava, karte događaja nemaju stupac opisa efekta, jer su predočene igračima u obliku grafike na kojoj je opisan efekt. Stoga, relacija sadržava jednostavne stupce: ime, status karte i identifikator relacije igre.

3.2.3. REST API

REST sučelje slijedi arhitekturni obrazac od 6 glavnih principa:

1. klijent–poslužitelj arhitektura
2. komunikacija bez stanja
3. mogućnost predmemoriranja
4. slojevitost - mogućnost posredništva između klijenta i poslužitelja bez utjecaja na sučelje

public.Game	
id	string
ownerId	string
blueTeamId	string
redTeamId	string
status	public.GameStatus
description	string
outcome	public.GameOutcome
timeLimit	number
activeSide	public.TeamSide
activePeriod	number
turnsRemainingTime	number
isRussianGovernmentAttacked	boolean
isRosenergoatomAttacked	boolean
isUkEnergyAttacked	boolean
lastAttacker	public.GameEntity
lastAttackStrength	number
didGCHQRevitaliseThisQuarter	boolean
recruitmentDriveCurrentQuartersStreak	number
recruitmentDriveMaxQuartersStreak	number
energeticBearAprilVitality	number
energeticBearAugustVitality	number
didRosenergoatomRevitaliseThisQuarter	boolean
growCapacityCurrentQuartersStreak	number
growCapacityMaxQuartersStreak	number
isRecoveryManagementActive	boolean
drawnEventCard	public.EventCardName
recordKeepingSheet	string

Slika 3.5: Relacija Game

public.Asset	
id	string
gameId	string
name	string
type	public.AssetType
effectDescription	string
minimumBid	number
blueTeamBid	number
redTeamBid	number
status	public.AssetStatus
turnsFromFirstBid	number
lastBidSide	public.TeamSide

Slika 3.6: Relacija Asset

5. uniformno sučelje - identificiranje dijelova resursa pomoću *URI* identifikatora i modularnost obrade zahtjeva

Svaki zahtjev klijenta pobuđuje odgovor poslužitelja. Svaki odgovor poslužitelja je neovisan o drugim zahtjevima i na taj način ne posjeduje stanja. Na svaki zatraženi resurs u zahtjevima poslužitelj poziva točno određeni kontroler koji obrađuje zahtjev.

Glavne funkcionalnosti aplikacije igre se osnivaju na REST API pozivima. Od autentikacije do poziva akcija u igri, svaka ruta ima definirane parametre i dodatne podatke koji se šalju u zahtjevima. U nastavku je dan popis dostupnih ruta (eng. *routes*) sa svojim parametrima i dodatnim podacima. Kontroler svakog modula je također naveden koji implementira rukovanje s tom rutom. Svaka ruta definira i vrstu HTTPS zahtjeva. Najkorištenije vrste su GET, POST, DELETE, PATCH. GET zahtjev HTTP protokola zahtjeva dohvaćanje resursa određenog URI-em. POST zahtjev označava kreiranje nove instance resursa određenog URI-em. DELETE zahtjev označava brisanje resursa identificiranog URI-em. PATCH i PUT zahtjevi su slične naravi kao i POST i koriste se za izmjenu resursa, međutim za potrebe razmatrane aplikacije koristit će se GET i POST zahtjevi, tj. stvaranje i dohvaćanje resursa. Ograničavanje na te operacije je isto u sklopu dobivanja MVP rješenja, a buduće nadogradnje bi obuhvaćale i dodavanje izmjena i brisanja instanci igara, igrača, korisnika, itd.

public.EventCard	
id	string
name	string
status	EventCardStatus
gameld	string

Slika 3.7: Relacija EventCard

Auth kontroler

POST */signup*

- Na rutu se šalju podaci *username* i *password*.
- Provjerava se postoji li već korisničko ime, ako ne, stvara se novi korisnik u bazi. Ako da, vraća prikladnu poruku klijentu u odgovoru.

POST */login*

- Na rutu se šalju podaci *username* i *password*.
- Uspoređuje unesenu lozinku s postojećom i na uspješnu autentikaciju postavlja kolačić u memoriji web preglednika za buduće zahtjeve.

POST */logout*

- Pomoću ove rute se čisti kolačić korišten za autentikaciju iz memorije preglednika i time se odjavljuje korisnika iz aplikacije.

GET */me*

- Ruta za dobivanje korisnikovih korisničkih podataka.

GET */users*

- Ruta za dobivanje liste trenutno aktivnih korisnika aplikacije.

Games kontroler

POST */games*

- Ruta za stvaranje instance igre. Podaci zahtjeva koji se šalju su:
 - *blueTeamName*
 - *electorateUserId*

- *ukPlcUserId*
- *ukGovernmentUserId*
- *ukEnergyUserId*
- *gchqUserId*
- *redTeamName*
- *onlineTrollsUserId*
- *energeticBearUserId*
- *russianGovernmentUserId*
- *rosenergoatomUserId*
- *scsUserId*
- *timeLimit*
- *description*

- Navedeni podaci se unose u obliku forme na klijentskoj aplikaciji.

GET */games*

- Ruta za dobivanje liste korisnikovih igara u kojima sudjeluje kao igrač.
- Dobivene informacije su sažetak i samo one informacije koje se prikazuju na početnom ekranu korisniku. Tek kad korisnik odabere jednu igru, šalje zahtjev za dohvaćanje svih detalja.

GET */games/ : id*

- Ruta za dobivanje kompletnih detalja igre sa zadanim ID parametrom u URL-u.
- Ovu se rutu poziva na ulazak korisnika na stranicu igre.

POST */games/action/ : type*

- Ruta za registriranje akcije igrača u aktivnoj igri.
- Parametar *type* označava vrstu akcije koju je igrač obavio.
- Na poziv ove rute osvježava se stupac *madeAction* u relaciji igre s novom korisnikovom radnjom.

POST */games/ : id/finishTurn*

- Ruta za završavanje poteza tima jedne igre određene parametrom *id*.
- Na poziv ove rute izvršavaju se sve registrirane akcije igrača i pokreće metoda započinjanja novog poteza.

GET */games/ : id/blackMarket*

- Ruta za dohvaćanje liste sredstava crnog tržišta igre određene parametrom *id*.

GET */games/ : id/ssets/ : teamSide*

- Ruta za dohvaćanje liste sredstava tima određenog parametrom *: teamSide* i igre određene parametrom *id*.

POST */games/bid/ : assetId*

- Ruta za registriranje ponude igrača na sredstvo definiranom parametrom *assetId*.

POST */games/ : gameId/activateAsset/ : assetId*

- Ruta za aktiviranje sredstva određenog parametrom *assetId* igre određene parametrom *gameId*.

POST */games/ : gameId/payRansomwareAttacker/ : answer*

- Ruta za registriranje odgovora igrača napadnutog s "Ransomware" sredstvom. *gameId* parametar određuje igru i parametar *answer* određuje odluku igrača.

POST */games/ : gameId/readEventCard/ : teamSide*

- Ruta za registriranje stanja pročitivosti karte događaja. Svaki potez kada igrači pročitaju i registriraju pročitivost klikom na klijentu, pamti se odabir da se više ne pokazuje u tekućem potezu.

3.2.4. Autentikacija

Autentikacija služi za verificiranje klijenta i njegovog zahtjeva. Tako se jedan korisnik asocira s igračem, i svaka akcija igrača mora biti autenticirana kako bi spriječili namjerne i nenamjerne zahtjeve s bilo kojih klijenata.

Za autentikaciju koristit će se *JWT* (eng. JSON Web tokens) tokeni, koji slijede tehniku *hashing* za provjeru autentičnosti korisnika. JWT tokeni će se slati pomoći kolačića web preglednika. Kolačići su pohranjeni u memoriji web preglednika u obliku ključ–vrijednost parova i na svaki zahtjev preglednika prema poslužitelju prilažu se u zaglavlju tog zahtjeva. Za svaki se kolačić definira i domena koja označava prema kojim se poslužiteljima prilažu ti kolačići. Tako se pomoću rute za prijavljivanje korisnika dobiva povratni odgovor koji sadrži novi generirani JWT token i zatim ga se sprema kao kolačić.

Za implementaciju strategije s JWT tokenima korištena je biblioteka *passport-jwt* koja nudi ugrađene funkcionalnost poput hashinga vrijednosti, definiranja konfiguracije i strategije validacije. Jedna od korištenih strategija je prilaganje korisničkih detalja svakom autenticiranom zahtjevu. To se obavlja nakon autentikacije i prije obrade zahtjeva. To je zgodan način za imati dostupne podatke o korisniku koji je poslao zahtjev. Podatak kojega se često koristi jest upravo ID korisnika.

Za povećavanje sigurnosti korisničkih podataka, na korisničku se lozinku, prije spremanja u bazu, dodatno konkatenira tzv. *salt* a potom hashira i sprema u bazu. Na taj se način sprječava potencijalna krađa podataka iz baze i njihova zlouporaba. Za tehniku *salting* i hashiranja koristi se JS biblioteka *bcrypt*.

3.2.5. WebSocket protokol

REST sučelje funkcionira po principu zahtjev–odgovor, i poslužitelj je u stanju pripravnosti u međuvremenu. Sve dok klijent ne zatraži resurs ili određenu operaciju baze podataka, poslužitelj neće odgovarati. Takva komunikacija je nepraktična za dva elementa razmatrane igre:

- brojač vremenskog ograničenja poteza
- vođenje zapisnika igre

Da bi se vremenski brojač implementirao na klijentima, trebala bi se uvoditi tehnika sinkronizacije preostalog vremena preko svih klijenata koji su priključeni u sesiju igre. To je velik izazov, jer klijent mora obavijestiti poslužitelja, koji mora autenticirati i verificirati valjanost podatka, a potom i obavijetiti sve ostale klijente, koji također imaju neko svoje vrijeme. Zato se vremenski brojač treba nalaziti na poslužitelju, koji će diktirati svim svojim klijentima vremenska ograničenja, tzv. "serversko vrijeme". Nadalje, ako bi se REST sučeljem slali zahtjevi za očitavanje trenutnog stanja brojača na poslužitelju, to bi značilo da klijent mora slati zahtjeve svaku sekundu i da poslužitelj mora zapisivati novo stanje brojača u bazu svaku sekundu. Takav slučaj bi štetio mrežnoj propusnosti klijenta, pogotovo korisnicima sporijeg mrežnog prometa, i aplikacija backenda bi trebala riješiti svakakve asinkrone i konkretne probleme. Zato se uvodi dvosmjerna komunikacija poslužitelja i klijenta. Ovo se i navodi u radu S. Marijanovića kao problem na koji je naišao. U njegovoj implementaciji istekom vremenskog ograničenja, zahtijevao se poziv s klijenta kako bi se pokrenula procedura za novi potez.

U arhitekturi u kojoj bi se riješio problem sinkronizacije klijenta i poslužitelja, klijent šalje HTTP zahtjev da želi promovirati komunikaciju u protokol WebSocket, koje

je okarakteriziran dvosmjernošću. Poslužitelj zatim pokreće *handshake* proceduru i jednom ostvoreni kanal komunikacije traje do njegovog prekida. U takvoj komunikaciji poslužitelj može *broadcastati* (hrv. odašiljati) poruke svim klijentima s takvom vrstom komunikacijskog protokola. Također, za potrebe ove igre definirane su tzv. "sobe" koje predstavljaju sesije svake pojedine igre. U te sesije se korisnici spajaju na ulazak u igru i odspajaju na izlazak.

Modul *Games* je specifičniji od ostalih modula, jer uz kontroler, servis i repozitorij inkorporira i dodatnu klasu, *Gateway* (hrv. pristupnik). Klasa koja implementira *Gateway* je ugrađena funkcionalnost radnog okvira Nest i nudi integraciju Nest okvira sa WebSocket protokolom. Temeljne funkcionalnosti odašiljanja poruka i primanja nudi dodatna JS biblioteka *socket-io*.

Gateway pristupna točka poslužitelja sadrži dvije klasne varijable: instancu *Socket IO* poslužitelja i varijablu *roomsTimers* koja sadrži popis soba i vrijednost njihovih trenutnih brojača. Brojač je implementiran pomoću *setInterval* JS funkcije koja prima tzv. *callback* funkciju. *Callback* funkcija je funkcija koja je predana drugoj funkciji i bit će izvršena nakon ispunjavanja određenih uvjeta. Tako će *setInterval* funkcija svakih 1 sekundu pozivati predanu *callback* funkciju. Predana *callback* funkcija će pozivati i odašiljanje poruka svim klijentima. U nastavku će se poruke detaljnije opisati.

Poruke mogu biti usmjerene s klijenta na poslužitelj ili s poslužitelja na klijent. *Gateway*, modula igre, sadrži dvije metode za prihvaćanje dolaznih poruka i dvije metode za odašiljanje poruka klijentima. Tako su definirane četiri vrste poruka:

- "startTimer" - dolazna poruka
- "pauseTimer" - dolazna poruka
- "tick" - broadcast poruka
- "log" - broadcast poruka

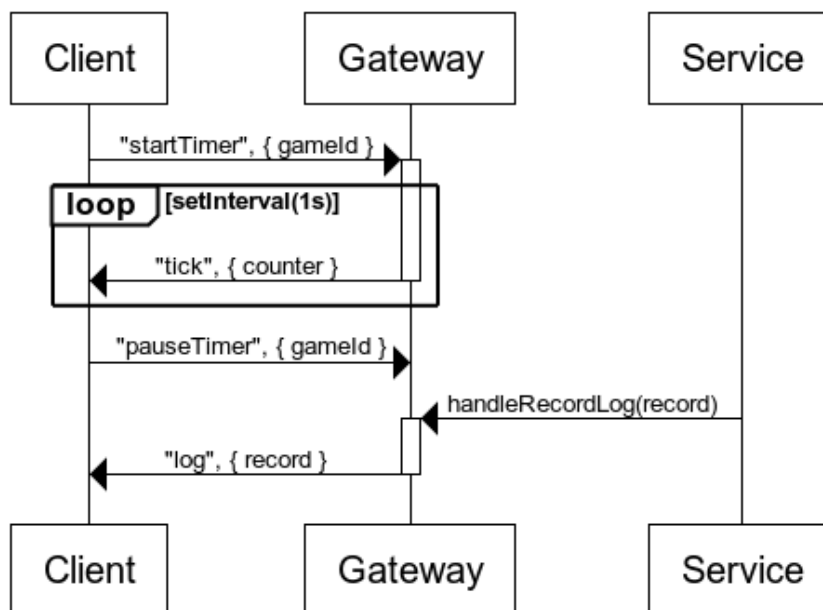
Sekvencijski dijagram WebSocket komunikacije je prikazan na slici 3.8.

Metoda *handleStartTimer* *Gateway* klase prihvaća poruku klijenta kada se želi pokrenuti brojač vremenskog ograničenja. Iz URI zahtjeva na socket određuje se parametar ID igre i određuje se identifikator sobe. Identifikator sobe se tvori kao znakovna vrijednost *room* : {*gameId*}.

Metoda *pauseTimer* zaustavlja vremenski brojač sobe određenog parametrom *gameId*.

U popularnom arhitekturnom obrascu među multiplayer igrama, poslužitelju se zadaje da u stvarnom vremenu odrađuje inkremente događaja igre. Tako će, u slučaju

WebSocket Events



Slika 3.8: WebSocket klijent–poslužitelj komunikacija jedne sesije igre

ove aplikacije, otkucaji vremenskog brojača predstavljati jedan inkrement igre koji će se morati sinkronizirati sa svim klijentima u sobi. Također postoje i metode optimiranja latencije klijent–poslužitelj komunikacije kao potencijalno daljnje unaprjeđenje [8]. Metoda *handleTimerTick* odašilje trenutnu vrijednost brojača svim klijentima u odgovarajućoj sobi kao payload porukom "tick". Upravo je to metoda koja se poziva kao callback funkcija *setInterval* JS funkcije. Nedostatak izvršavanja *setInterval* funkcije u glavnom procesu backend aplikacije je to da ako proces dođe do naglog prekida, gube se sve informacije vremenskih brojača svih soba. Postoje razne tehnike kontinuiranog zapisivanja trenutne vrijednosti u bazu, ali to je izvan opsega funkcionalnosti ove aplikacije i jedno od mogućih nadogradnji. Trenutno, vrijednost vremenskog brojača se pohranjuje u stupac *turnsRemainingTime* relacije *Game* na poruku "pauseTimer", kako bi brojač mogao nastaviti od prošlog trenutka.

Odašiljanje novog zapisa zapisnika igre nije toliko prioritarna operacija u odnosu na vremenski brojač, stoga se on odrađuje asinkrono radi povećanja performansi. Metoda *handleLogRecord* odašilje poruku "log" svim klijentima, a metodu pozivaju metode ostalih servisa kada se obave određene akcije igre.

Vođenje zapisnika igre je, također, procedura koja zahtjeva da poslužitelj obavještava sve klijente u otvorenoj sesiji igre. Zato se, na dodavanje novog događaja igre (zapisa), taj zapis odašilje svim klijentima koji ga onda prikladno prikazuju u svojim

igrama.

3.2.6. Obavljanje akcija i završavanje poteza

Nakon pokretanja igre, igrači su u mogućnosti raditi poteze. Na registraciju akcije igrača, akcija se zapisuje u relaciju *Game* i koristi klijentskoj aplikaciji u narednom potezu da pokaže što je igrač igrao zadnji potez. Naravno dok se na završi potez igrač smije mijenjati svoj odabir akcije. U prvoj inačici igre izvršavanje akcije igre bilo je instantno i nepovratno, dok se u finalnoj inačici na završetak poteza poziva ruta `/games/ : id/finishTurn` kojom se poziva metoda koja tek onda obavlja sve registrirane akcije, odražava promjene na entitetima u bazi i poziva metodu `nextTurn` servisa *Games* koji pokreće novi potez. Nakon svake akcije koja se obavi doda se zapis u zapisnik igre i odašilje svim klijentima. Izuzetak je akcija apstinencija koja se ne zapisuje radi uštede prostora i čitljivosti klijentske aplikacije. Nakon rekonstruiranja igre, svaki se mjesec može jasno zaključiti koji su igrači apstinirali.

U prvoj inačici razvoja aplikacije, obavljanje akcija je bilo nepovratno i akcije bi se dinamički obavljale u tijeku poteza tima. To bi značilo da su poslani resursi između dva entiteta odmah bili dostupni entitetu koji je dobio te resurse, čak i prije završetka poteza. Mehanika obavljanja poteza nije jednoznačno opisana u radu A. Haggmana, stoga su i neki elementi i mehanike podložne promjenama. Testiranjem igre se utvrdilo da je pogodnija mehanika u kojoj svaki igrač po završetku poteza obavlja svoje akcije. Testiranje aplikacije i njezine mehanike će se opisati u poglavlju 4.

Metoda `nextTurn` obavlja sljedeće provjere i pozive:

1. dekrementira sve kondicijske brojače igrača (zabrane, paralize, imuniteti)
2. provjerava zadovoljava li ijedan entitet neki od svojih ciljeva, ako da, dodijeli pobjedničke bodove i dodaj novi zapis igre
3. resetira sve brojače i zastavice trenutnog poteza (npr. je li entitet pretrpio štetu ovaj potez, je li se revitalizirao, itd.)
4. provjerava je li igra završena (kraj prosinca ili jedan entitet ima vitalnost 0), ako je pokreni metodu za završetak igre i kalkulaciju bodova
5. ako igra nije gotova, dostavi novo sredstvo na crno tržište i provjeri je li ikoje sredstvo osvojeno, ako je, dodijeli ga odgovarajućem timu i dodaj zapis
6. ako je kraj mjeseca, izvuci novu kartu događaja, primijeni efekt za naredni mjesec i dodaj zapis igre

3.2.7. Aktiviranje efekata sredstava i karti događaja

Aktivirati sredstvo jednog tima može bilo koji igrač u timu. Tu se također oslanja na aspekt suradnje u timu. Na POST ruti za aktiviranje sredstva efekt se primjenjuje osvjehavanjem stanja relacije igre, tima ili igrača ovisno o efektu. Igra sadržava različite oblike kondicija poput paralize, imuniteta, smanjenja dobivene štete, itd. Kondicije se prate zasebnim zastavicama u relacijama i brojačima za efekte koje traju određeni broj poteza. U metodi *nextTurn* se te zastavice i brojači na prikladan način ažuriraju. Svako sredstvo posjeduje vlastitu metodu u servisu igre koji sadrži postavljanje točno određenih zastavica i brojača. Također, pri aktivaciji sredstva se i dodaje novi zapis u zapisnik igre.

Primjer jedne aktivacije je aktivacija sredstva "Education". Efekt sredstva omogućuje entitetu *Electorate* da prima dvostruko manju štetu sljedeća 3 poteza. Aktivacijom tog sredstva, poziva se metoda *activateEducation* servisa *Games*. Ta metoda poziva metodu *activateArmor* servisa *Players* sa sljedećim argumentima: ID igrača, vrijednost štita (50) i trajanje kondicije (3). Za smanjivanje štete uveden je koncept štita koji ublažuje nastalu štetu. Štit se definira u intervalu [0, 100], gdje je štit 0 početna vrijednost, a štit 100 maksimalna vrijednost i ublažuje svu primljenu štetu.

Formula po kojoj se računa rezultatna šteta u igri je:

$$damage = ((100 - armor)/100) * attackStrength$$

Slično kao i za sredstva, i karte događaja imaju odgovarajuće metode aktivacije koje postavljaju zastavice i brojače. Međutim, ne postoji ruta za aktivaciju karte pošto backend aplikacija automatski svaki mjesec primjenjuje efekte događaja.

3.2.8. Zapisnik igre

Zapisnik igre se vodi radi evidencije akcija igrača i mogućnosti kasnije rekonstrukcije igre. Zapisnik igre može pomoći u analizi igre i uvježbavanju budućih strategija na temelju prošlih razina uspješnosti. Zapisnik igre je dostupan kao stupac *recordKeepingSheet* u relaciji *Game*.

Zapisnik se vodi u obliku HTML *rich text* formata. To je format kojega web aplikacije mogu lako čitati, kojega se može uređivati prilagođenim stilovima i čitljiv je. Klijentska aplikacija će tako moći stilizirati zapisnik u igri. Primjer zapisnika jednog mjeseca u početnom formatu prikazan je na slici 3.9.

February

[EVENT CARD] Quantum Breakthrough is drawn

[BLACK MARKET] SCS bids 2 resource on Network Policy

[ACTION] Energetic Bear attacks UK PLC. 2 resource is spent, dice rolls a "3" and attack deals 1 damage.

[ACTION] Russian Government sends 2 resource to Online Trolls

[ACTION] Rosenergoatom revitalises by 1 vitality

[ACTION] Online Trolls revitalises by 1 vitality

[BLACK MARKET] GCHQ bids 3 resource on Network Policy

[ACTION] UK PLC revitalises by 2 vitality

[ACTION] UK Government sends 2 resource to GCHQ

[ACTION] Electorate revitalises by 2 vitality

[ACTION] UK Energy revitalises by 2 vitality

[OBJECTIVE] Russian Government +1 VP "Some animals are more equal than others"

Slika 3.9: Izvadak zapisnika jednog mjeseca u igri

3.3. Aplikacija klijenta

Aplikacija klijenta treba sadržavati interaktivnu funkcionalnost igre i preslikavati korisničke akcije u pozive prema poslužitelju. Također, vizualni prikaz je bitno prezentirati na način da privuče korisnika, a funkcionalnost aplikacije na način da korisniku interakcija bude maksimalno jednostavna i intuitivna.

Za početak će se navesti stranice koje sačinjavaju aplikaciju i ponajviše će se opisati stranica same igre. Zatim će se proći kroz neke odluke dizajna i rasporeda koji služe za poboljšanje korisničkog iskustva.

3.3.1. Stranice aplikacije

Stranica prijave i registracijska stranica sadrže forme koje korisnik ispunjava da bi se registrirao, a potom prijavio u svoj račun.

Forma za prijavu je prikazana na slici 3.10. Nakon prijave, korisnika se odvodi na početnu stranicu gdje se nalazi popis svih igara u kojima korisnik sudjeluje. Početna stranica je prikazana na slici 3.11.

Svaki korisnik ima mogućnost stvaranja igre, klikom na "Create a new game", što ga vodi na stranicu za stvaranje nove igre. U popisu igara, svakoj igri su prikazane

Login



 Username



 Password

LOGIN

or Sign Up

Slika 3.10: Stranica forme za prijavu

Cyber Security Wargame Hi, Physx  

No.	Blue team	Red team	Status	Description	Link
#1	Autobots	Decepticons	Paused		

Create a new game

Slika 3.11: Početna stranica aplikacije

imena timova, status igre, opis igre i link koji vodi na stranicu igre.

Stranica za stvaranje nove igre sadrži formu, koju korisnik popunjava s informacijama o imenima timova, igračima svakog entiteta, vremenskom ograničenju i opcionalnom opisu igre. Stranica s formom je prikazana na slici 3.12.

Stranica igre prikazana je na slici 3.13. Stranica je vizualno podijeljena na 3 dijela. Vertikalnim redoslijedom to su: gornji izbornik, ploča igre i donja navigacijska traka igrača.

Prilikom ulaska na ovu stranicu obavljaju se sljedeći koraci unutar procesa frontend aplikacije:

1. dobavlja se parametar ID igre koji se nalazi u URL-u stranice
2. uspostavlja se WebSocket konekcija s poslužiteljem pomoću parametra igre i ulazi se u "sobu" igre
3. poziva se ruta za detalje igre i dohvaćaju se sve potrebne informacije za prikaz korisniku

Za determiniranje parametra ID igre iskorištena je SSR tehnika radnog okvira Next, pa je tako na serveru određen parametar iz URL a na klijentu je, pri učitavanju, taj podatak odmah dostupan. To sprječava asinkrone probleme s čitanjem parametra URL-a na klijentu, koji na kratko vrijeme može biti nedefiniran.

3.3.2. WebSocket komunikacija

Pri uspostavi komunikacijskog kanala, ulaskom na stranicu igre, definiraju se metode za rukovanjem poruka na server. Koristi se JS biblioteka *socket.io-client* za rad s WebSocket protokolom. Prilikom definiranja konekcije zadaje se URL na kojemu se odvija WebSocket komunikacija i parametar koji se prilaže na zahtjeve, a to je *gameId*. Upravo taj parametar identificira sobu kojoj se korisnik želi priključiti.

Korisnik pritiskom na gumb za nastavak ili pauziranje igre šalje poruku na poslužitelj i pokreće odnosno zaustavlja vremenski brojač. Dolazne WebSocket poruke s poslužitelja registriraju *listener* metode i prikazuju ih korisniku. Tako će za "tick" poruku aplikacija prikazivati vrijednost brojača u gornjem izborniku, a za "log" poruku će se prikazati novi zapis u zapisniku donje navigacijske trake.

Blue Team _____ VS _____ Red Team

Electorate Player*

Online Trolls Player*

UK PLC Player*

Energetic Bear Player*

UK Government Player*

Russian Government Player*

UK Energy Player*

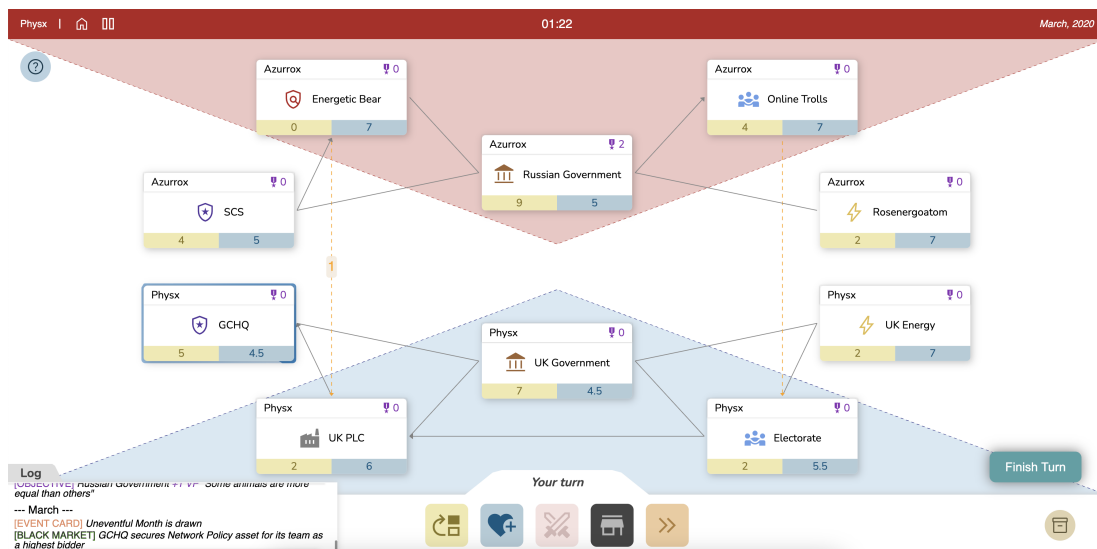
Rosenergoatom Player*

GCHQ Player*

SCS Player*

Time Limit

 min**CREATE****Slika 3.12:** Stranice za stvaranje nove igre



Slika 3.13: Stranica igre

3.3.3. Interaktivnost i korisničko iskustvo

Uz pristupačnost web aplikacije, važno je korisnicima omogućiti i najbolje korisničko iskustvo. Tijekom godina su se razlučili određeni principi korisničkih iskustava koje valja primijeniti [9]. Za početak će se razgovarati o načelu bliskosti vezano na raspored stranice igre, zatim i o asocijativnom principu kojega se ostvaruje pomoću boja. Asociranje boja s određenim elementima igre poboljšava pamćenje tih elemenata i omogućuje bojanje mjesta koji su blisko vezani s osnovnim pojmom.

Prvi dojam kojega korisnik dobiva je vizualni, tj. raspored, boje i minimalizam. Tri glavna djela na stranici igre su opisana u poglavlju 3.3.1. i prikazani na slici 3.13.

Gornja navigacijska traka sadrži redom: korisničke akcije, vremenski brojač i trenutni period igre. Interaktivnost gornjeg izbornika se odvija samo u gornjem lijevom kutu i korisnik odlazi tamo samo ako želi ili izaći iz igre ili ju zaustaviti. Ostali statusni dio, vremenski brojač i trenutni period igre, igrač samo povremeno pogleda kada je to potrebno. Na taj način se sva interaktivnost vezano uz igru premjestila na ploču i u donju navigacijsku traku. "Igrač" igre dolazi u gornji lijevi kut kada želi biti "korisnik" aplikacije, a u donji dio stranice kada želi biti "igrač" aplikacije.

Srednji dio, predstavlja ploču igre i sadrži sve relevantne informacije o stanju igre. Tu se igraču predočava prostor na kojemu se odvija sukob kibernetičkog ratovanja. Tu je također i mjesto na kojemu može razgledavati i ciljeve klikom na pobjedničke bodove bilo kojeg entiteta. Zbog jednostavnosti i intuitivnosti igrač odabire točno onaj svoj entitet s kojim želi obaviti akciju, i tada mu se omogućuju akcije u donjoj navigacijskoj traci.

Ploča igre nailazi na manu web aplikacija po pitanju prostora. Dimenzije ekrana mogu varirati i predstavljati problem korisnicima s manjima zaslonima. Tako je ova aplikacija ograničena na rezolucije od 720 piksela u širini naviše. U originalnom radu je ploča predstavljena sa horizontalnim rasporedom sukobljenih strana. To je, međutim, zahtjevno za prikazati na računalnim zaslonima, jer bi onda i entitete trebalo prikazivati rotirane za 90 stupnjeva. Tu se postavlja i pitanje, zašto bi to htjeli za korisnike web aplikacije. Web aplikacija se razlikuje od ploče u tome što igrači nisu okupljeni oko ploče, već jedan korisnik računala gleda u svoj zaslon i raspored na njemu. Imajući to na umu, intuitivnije je za korisnika ako gleda svoju stranu sukoba na donjoj strani zaslona orijentiranu prema gore, gdje se nalazi protivnički tim. Time se i čuva princip bliskosti, jer će se akcije u donjoj navigacijskog traci odnositi na entitet korisničkog tima, koji je isto na donjoj strani. Ista je odluka donešena i u radu S. Marijanovića.

Donja navigacijska traka je mjesto gdje igrač može odabrati akciju koju želi napraviti s pojedinim entitetom. Na taj način su se akcije u potpunosti premjestile u navigacijsku traku i igrač će sve vezano za akcije tražiti dolje. Također, u donjem desnom kutu navigacijske trake igrač može otvoriti i inventar svog tima gdje se nalaze skupljena sredstva. Ta sredstva može onda aktivirati.

U donjem lijevom kutu navigacijske trake igraču je prikazan zapisnik igre, gdje igrač može pregledavati dosadašnje obavljene akcije, izvučene karte događaja, ponude na crnom tržištu i aktivacije sredstava.

A. Haggman u svom radu također pridodaje važnost bojama i pridružuje specifične boje određenim elementima igre. Tako su te boje zadržane i u implementaciji aplikacije, kao i dodane nove boje na mjestima gdje je to prikladno. Korištene boje prikazane su u tablici 3.1.

3.4. Daljnja poboljšanja

Za početak, menadžment korisnika aplikacije je stvoren s minimalnom autentikacijom koja sadrži dovoljno sigurne procese. Daljnji koraci nadogradnje bi bili uvođenje registracije s e-mail adresom, omogućavanje Google *OAuth* autentikacije, pripisivanje korisnicima različite uloge i razine autorizacije.

Vezanje e-mail adresa s korisničkim računima omogućuje napredne verifikacijske postupke kao što je potvrda novostvorenog računa. Također, siguran je način obnove zaboravljenih lozinka i ostalih korisničkih detalja.

Google OAuth autentikacija nudi jednostavnu registraciju i prijavu korisnika. Gmail računi danas čine veliku većinu osobnih elektroničkih adresa i pojednostavio bi

Pojam	Boja
primarna boja aplikacije	"firebrick", #b22222
UK strana	svijetlo plava, rgba(48, 141, 193)
ruska strana	svijetlo crvena, rgba(190, 64, 55)
resurs, akcija distribucije	zlatna, #877725
vitalnost, akcija revitaliziranja	svijetlo plava, #105881
pobjednički bodovi, zapis o postignutom cilju	tamno ljubičasta, #8729b3
crno tržište, zapis o crnom tržištu	crna
sredstva, zapis o sredstvima	smeđa, #e0d4af
ruski vektor napada	narančasta
vektor napada UK-a	ljubičasta

Tablica 3.1: Tablice korištenih boja u aplikaciji

proces registracije i prijave tako da korisnici ne pamte nove detalje stvorenog računa. Ovdje se naravno može raspraviti o sigurnosti takvog pristupa.

Jedna od mogućih primjena uloga i različitih nivoa autorizacije korisnika je ako se igrama dodaju mogućnost daljnje konfiguracije. Tako bi korisnik s većom razinom autorizacije imao pristup većim razinama postavki određenih igara. Naravno, na svakve se načine može ograničiti uporaba aplikacije.

Nadalje, ima mjesta za poboljšanje i po pitanju sinkronizacije stanja poslužitelja i klijenata. Ako jedan igrač napravi promjenu u igri koja se ostalim igračima treba očitovati odmah, ne samo nakon završenog poteza, tada bi se mogla uvesti dodatna poruka za WebSocket komunikaciju s kojom bi poslužitelj obavijestio sve klijente da osvježe svoje stanje igre. Na ključne bi se događaje odašiljala ta poruka koja bi prouzročila da svi klijenti unutar jedne sesije pošalju zahtjev za najnoviju verziju igre. Međutim, tu bi rješenje trebalo sagledati i po pitanju performansi. Za jedan odaslani signal, potencijalno 10 igrača bi moglo zatražiti isti resurs. Osim opterećenja poslužitelja, i klijent bi možda opterećivao svoju mrežnu propusnost. Poruka "tick" se u ovoj implementaciji nije željela koristiti kao takva vrsta signala, jer se time svaku sekundu nepotrebno osvježavaju podaci igre na svim klijentima.

Implementacija REST API sučelja dozvoljava ne samo korisnicima, već i skriptama da šalju svoje zahtjeve na poslužitelj. Pisanje skripti i automatizacija igranja igre može sakupiti veći broj igara u manje vremena od uobičajenih korisnika i otvoriti mogućnost analize svake pojedine igre. U potrazi za najboljom strategijom, na određenom setu podataka mogu se trenirati i strojni modeli čiji bi rezultati mogli dati uvid u neke nove

ili neočekivane strategije. Ili jednostavno u one najbolje.

Jedna od mogućih nadogradnji je i dodavanje novih ruta za omogućivanje brisanja i uređivanja igara i korisnika. Na taj način se korisniku može omogućiti kontroliranje natrpanosti igara na početnoj stranici, a time se može i štediti na memorijskom zauzeću poslužitelja. Osim toga, na preglednosti početne stranice se može dobiti i pomoću filtera i tražilice koje korisnik može koristiti da sakrije, pokaže ili pronađe točno specifičnu igru.

4. Testiranje aplikacije

4.1. Korisničko iskustvo

Od početka su bili primijenjeni principi korisničkog iskustva vizualnog dijela, ali se mehanika igre mijenjala. Prikupljenim mišljenjima, iskustvima i implicitnih zaključaka originalne igre u radu A. Haggmana igra se razvila kroz dvije inačice.

Prva inačica sadržavala je mehaniku igre u kojoj su se akcije vršile nepovratno i rezultati akcija su bili dostupni odmah po izvršenju, prije završetka poteza. U 5 odigranih igri, 1 je završila agresivno, tj. napadom se reducirao UK Electorate entitet na 0. Dok se u 5 odigranih igara druge inačice 3 od 5 igri završilo agresivnim načinom. Jedan od mogućih zaključaka je da prvotna mehanika omogućuje veći protok resursa kroz cijeli sustav, a time i puno uspješnije revitaliziranje. Općenito, i mehanika ponuda na crnom tržištu je lakša s većim protokom resursa. Otežavanje mehanike igre svakako stavlja veći naglasak na razmišljanje i pronalaženje prave strategije. Zbog toga se, u drugoj inačici promijenila mehanika obavljanja akcija po završetku poteza tima.

Drugo, vitalnost je predstavljena u prvoj inačici kao brojeva vrijednost s pomičnom točkom. Pa su entiteti poprimali i decimalne vrijednosti vitalnost. U inačici dva sva, zbog predstavljanja vitalnosti pomoću fizičkih kockica u originalnoj igri na ploči, šteta se zaokružuje na prvu manju cjelobrojnu vrijednost.

Akcija distribucije, u originalnoj igri na ploči, isto nije jednoznačno opisana po pitanju broja entiteta kojima se mogu slati resursi u jednome potezu. Ograničavanje na jedan entitet povećava kompleksnost igre, kao i obavljanje akcija po završetku poteza. S. Marijanović u svojem radu navodi da se resursi mogu slati bilo kojem broju povezanih entiteta, ali s obzirom na princip otežane mehanike obavljanje akcija po završetku poteza, odlučeno je da se i s akcijom distribucije ide na težu varijantu radi otežavanja strategije i potencijalno boljeg učenja.

Vizualni dijelovi su većinom odgovarali igračima i kontrole igre su bile intuitivne. Potpuni uspjeh je doživio pak princip asocijacija boja, jer izostanak natpisa resursa i vitalnosti na karticama entiteta nije dovelo do zbunjivanja igrača. Mogući razlog

tomu je i dodani pomoćni izbornik u gornjem lijevom kutu igre ploče u finalnoj inačici igre. Boje su također i pridonijele vizualnoj estetici što pridonosi početnoj privlačnosti aplikacije. Više igrača je reklo da je dizajn čist i lijep.

4.2. Iskustva odigranih igara

U prvoj inačici igre, nakon 5 odigranih igara igrači su uočili da se ne isplati biti agresivan, što nije više bio slučaj u drugoj inačici zbog navedenih promjena. Druga inačica je uvela ravnomjerniju korištenost različitih strategija navedenom promjenom. Time se, u drugoj inačici igre, uvela i strategija veće akvizicije sredstava koja donosi bonuse za vlastiti tim. Od 5 odigranih igara druge inačice, 3 su pobjede Rusije i 2 UK strane. Pobjednička strategija Rusije je bila agresivnost napadima potpomognuto "Ransomware" i "Stuxnet 2.0" sredstvima, 1 pobjednička strategija UK strane je bila akvizicija obrambenih sredstava i 1 pobjednička strategija UK strane je bila povećana rata revitalizacije uz sredstvo "Cyber Investment Programme" i ganjanje ciljeva svih entiteta.

Kroz sakupljena iskustva igri, postojali su različiti trenuci u kojima bi se, na temelju novih predočenih informacija, preispitivale dosadašnje odluke i pokušavalo utvrditi najsigurniju strategiju prema naprijed. Upravo je to, rekao bi, svrha uvježbavanja kibernetičkog ratovanja i na taj način se prikupljaju tzv. trenuci učenja kroz igru.

4.3. Nedostatci online igara

Postoje mane online igri s više igrača. Pogotovo, po pitanju bliske suradnje sa suigračima. Jedna od tih mana je i preglednost situacije. Kao što je opisano već u poglavlju 3.3.3., dimenzije računalnih ekrana ograničavaju raspored elemenata igre i tjeraju dizajnere i programere da osmisle najprikladniji način za prikazivanje svih elemenata. Međutim, donesene odluke dizajna aplikacije, u ovom radu, su dovoljno pažljivo donesene i ublažavaju tu manu. Nadalje, prednost pristupačnosti i automatizacije procesa igre su ključni faktori za koje se misli da mogu prevagnuti u korist web aplikacije.

5. Zaključak

Završna inačica implementacije igre sadržava sve bitne elemente igre. Pokriva sva pravila i nadograđivana je na temelju iskustava igrača, savjeta i implicitnih zaključaka. Valja napomenuti da, iako sadržava svu ciljanu funkcionalnost, je i dalje razvijena kao MVP rješenje, kojega se, naravno, može nastaviti poboljšavati.

Tijekom izrade web aplikacije, izabrale su se tehnologije koje su moderne, nude skalabilan razvoj i popularne su, što znači da postoje bogate zajednice i forumi na kojima se mogu tražiti savjeti ili razriješiti problemi. Uz tehnologije, odabrani su i prikladni arhitekturni obrasci za izradu igre s potporom za više igrača. Ostvarena je klijent–poslužitelj arhitektura u kojoj je riješena i komunikacija u stvarnom vremenu koristeći protokol WebSocket. REST sučelje je implementirano modularno, a time i skalabilno otvoreno za buduće nadogradnje. Tu se nudi mogućnost automatizacije igranja igri i dubinske analize statistika odigranih igri.

U razvoju aplikacije klijentske strane, dotaknuti su i principi korisničkih iskustava i primijenjen je dizajn i raspored aplikacije za koji se smatra da je dovoljno jednostavan i intuitivan kako bi se korisnici mogli posvetiti i udubiti u strategije igre. Također, razriješene su i neke nedoumice oko originalnog seta pravila. Osim vlastitih zaključaka uzeti su i savjeti testnih igrača, mentora i asistenta koji su mi pomagali tijekom semestra i zaključaka u diplomskom radu Stjepana Marijanovića.

Na kraju poglavlja 3. dani su prijedlozi za budući nastavak rada i poboljšanja aplikacije. Od potencijalnih poboljšanja, najistaknutije bi bila bolja organizacija korisnika i dodavanja postavki i parametara igri i mogućnost njihovog podešavanja.

Sve u svemu, postignuta implementacija igre dobro odražava igru u njezinim pravilima i njezinoj reprezentaciji kako je opisana u radu A. Haggmana. Igrači mogu tijekom igranja web aplikacije naučiti i uvježbati mogućnosti različitih strategija. Unatoč nedostacima web aplikacije, i prednosti igara s fizičkom pločom, navedene su prednosti web aplikacije i njezine uporabe, kako za korisnike tako i za druge uporabe.

LITERATURA

- [1] W. Ross Ashby. *An introduction to cybernetics*. Chapman & Hall Ltd., 1957.
- [2] Katrin Becker. Teaching with games: the minesweeper and asteroids experience. *Journal of Computing Sciences in Colleges - JCSC*, 17, 01 2001.
- [3] Holiseum GamePartners. Cyber Wargame. <https://www.cyber-wargame.com/en/home> [Accessed: 20. Jun 2023.].
- [4] Andreas Haggman. *Cyber wargaming: Finding, designing, and playing wargames for cyber security education*. PhD thesis, Royal Holloway, University of London, 2019.
- [5] Group of Naval Postgraduate School students. CyberCIEGE Wargame. <https://nps.edu/web/c3o/cyberciege> [Accessed: 20. Jun 2023.].
- [6] Group of volunteering contributors. OverTheWire Wargame. <https://overthewire.org/> [Accessed: 20. Jun 2023.].
- [7] Lee Sheldon. Game-Based learning, collateral learning, and beyond. In *Game-Based Learning, Collateral Learning, and Beyond*, Washington, D.C., August 2015. USENIX Association.
- [8] Eirik Rodvang Tveterås. Evaluating loss and latency mitigation techniques in a tick-based game server. Master's thesis, University of Oslo, 2018.
- [9] Jon Yablonski. *Laws of UX: Using psychology to design better products & services*. O'Reilly Media, 2020.

Implementacija igre za uvježbavanje kibernetičkog ratovanja na strateškoj razini

Sažetak

Kibernetičke vježbe služe za osposobljavanje posebno opremljenog tima za protekciju informatičkih sustava i ublažavanje štete koji razni napadači mogu prouzročiti. Ovaj rad proučava igru opisanu u disertaciji Haggman, Andreas. Cyber wargaming: finding, designing, and playing wargames for cyber security education. Diss. Royal Holloway, University of London, 2019. Opisana igra za uvježbavanje je izrađena za potrebe uvježbavanja kibernetičkog ratovanja na strateškoj razini. Rad opisuje metodologiju i razvoj igre u obliku web aplikacije. U radu se navode tehnologije web aplikacije, pregled implementacije, izazovi web aplikacije u odnosu na igru s pločom i korisničko iskustvo stvorene aplikacije. Također se, uz stečeno znanje već jedne postojeće implementacije igre, pokušavaju nadograditi nedorečeni elementi igre.

Ključne riječi: sigurnost, kibernetika, ratovanje, igra, strategija, simulacija, aplikacija, uvježbavanje

Implementation of a game for practicing cyber war on a strategic level

Abstract

The purpose of cyber exercises is to train a specialized team to protect an information system and mitigate any damage that any of the possible attackers might inflict. This thesis studies an exercise game described in a Ph.D. dissertation Haggman, Andreas. Cyber wargaming: finding, designing, and playing wargames for cyber security education. Diss. Royal Holloway, University of London, 2019. The described exercise game is made for practicing cyber war on a strategic level. The thesis describes the methodology and development of a game in the form of a web application. The thesis describes the technologies used for the application development, describes the implementation itself, lists the challenges of the web application game versus the board game, and analyses the user experience of the implementation result. Also, with the help of an already existing game implementation, ambiguities of some game elements are tackled.

Keywords: cyber-security, wargaming, strategy, simulation, application, exercise