

**Laboratorij za tehnologije znanja (KTLab)**

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Fakultet elektrotehnike i računarstva

Sveučilište u Zagrebu

## **Interni dokument**

**© 2010 KTLab**

Niti jedan dio ovog dokumenta ne smije se fotokopirati,  
umnožavati niti prevoditi na drugi jezik  
bez prethodnog pismenog odobrenja.

SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

# Programska izvedba simulatora i igračih algoritama za društvenu igru dame

---

**Nikola Novak**

DIPLOMSKI RAD

Zagreb, listopad 2010.

Zadatok

INTERNI DOKUMENT

*To think is an act of choice. [...] Reason does not work automatically; thinking is not a mechanical process; the connections of logic are not made by instinct.*

--Ayn Rand, "Galt's Speech" in For the New Intellectual

Zahvaljujem svom mentoru prof. dr. sc. Bojani Dalbelo-Bašić  
na svoj pruženoj pomoći,  
dr. sc. Janu Šnajderu  
na korisnim savjetima i smjernicama danim u bilo koje vrijeme,  
dipl. ing. Matiji Lackoviću  
na produktivnoj suradnji  
i svim sudionicima Izazova 2010  
bez čijih igračih algoritama veliki dio ove studije ne bi bio moguć.

## Popis tablica

Tablica 1. Kratki opis igrača koji su se koristili u turnirima dame. ....	28
Tablica 2. Prikaz svih varijanti igrača Lockdown izvedenih za ablacijsku studiju. ....	29
Tablica 3. Rezultati turnira s izvornim igračem Lockdown .....	30
Tablica 4. Rezultati turnira s verzijom igrača Lockdown bez evaluacije figura .....	30
Tablica 5. Rezultati turnira s verzijom igrača Lockdown bez evaluacije dama .....	31
Tablica 6. Rezultati turnira s verzijom igrača Lockdown bez evaluacije poteza .....	32
Tablica 7. Rezultati turnira s verzijom igrača Lockdown bez dijela heuristike za kretanje prema protivniku .....	34
Tablica 8. Igre odigrane s igračem Lockdown protiv rangiranih ljudi. ....	37
Tablica 9. Rezultati turnira s igračem Junior .....	38
Tablica 10. Popis direktorija na priloženom DVD-u koji sadrže turnire i pojedinačne igre.....	46

INTERNI DOKUMENT

## Popis slika

Slika 1. Grafički prikaz rada algoritma minimax. ....	6
Slika 2. Korisničko sučelje igre dame. ....	20
Slika 3. Korisničko sučelje igre halma. ....	25
Slika 4. Vrijednosti polja ploče za igru halme koja se koriste pri proračunu heurističke vrijednosti pozicije. ....	27
Slika 5. Pozicija nastala u igri TheMarvin vs. Lockdown-Kings, potez 65. ....	32
Slika 6. Pozicija nastala u igri VOV vs. Lockdown-Taking. ....	33
Slika 7. Dvije pozicije nastale tokom partije TheMarvin vs. Lockdown-Taking: a) potez 27, b) potez 45. ....	34
Slika 8. Dvije pozicije nastale u igri Lockdown-Travel vs. VOV: a) potez 28, b) potez 36. ....	35
Slika 9. Jedna u nizu pozicija koja bi se trebala dogoditi da bi igrač Lockdown spasio svoje figure u igri protiv igrača VOV. ....	36
Slika 10. Pozicija u partiji Junior vs. TheMarvin, potez 34. ....	38
Slika 11. Grafički prikaz uz opis pokretanja testnog primjera. ....	43
Slika 12. Grafički prikaz uz opis učitavanja igre dame. ....	44
Slika 13. Grafički prikaz uz opis učitavanja igre halma. ....	45

# SADRŽAJ

<b>1. UVOD</b> .....	<b>1</b>
1.1. O UMJETNOJ INTELIGENCIJI .....	1
1.2. MOTIVACIJA.....	2
1.3. CILJ RADA .....	2
1.4. STRUKTURA RADA .....	3
<b>2. ALGORITMI KORIŠTENI ZA IZRADU IGRAČA</b> .....	<b>4</b>
2.1. ALGORITAM MINIMAX.....	5
2.2. PODREZIVANJE ALFA-BETA .....	8
2.3. NEURONSKE MREŽE .....	9
2.4. GENETIČKI ALGORITAM.....	11
2.5. OPIS ALGORITMA IGRAČA ZA IGRU HALMA .....	13
<b>3. OKRUŽENJE MGMTM</b> .....	<b>15</b>
3.1. SUČELJE PREMA KORISNIKU .....	15
3.2. PROGRAMERSKA SUČELJA .....	16
3.2.1. <i>Sučelja prema igri</i> .....	17
3.2.2. <i>Sučelje prema igračima</i> .....	17
<b>4. IGRE I IGRAČI</b> .....	<b>19</b>
4.1. IGRA DAME .....	19
4.1.1. <i>Pravila igre</i> .....	20
4.1.2. <i>Igrač Randomizer</i> .....	21
4.1.3. <i>Igrač Lockdown</i> .....	22
4.1.4. <i>Igrač Junior</i> .....	23
4.2. HALMA .....	24
4.2.1. <i>Pravila igre</i> .....	25
4.2.2. <i>Igrač Randomizer Jr.</i> .....	26
4.2.3. <i>Igrač Dumbo</i> .....	26
<b>5. VREDNOVANJE IGRAČA</b> .....	<b>28</b>
5.1. TURNIR I PROTIVNICI.....	28
5.2. IGRAČ LOCKDOWN .....	29
5.2.1. <i>Ablacijska studija</i> .....	29
5.2.2. <i>U igri protiv čovjeka i samoga sebe</i> .....	37
5.3. IGRAČ JUNIOR .....	38
5.3.1. <i>U turniru protiv drugih igračih algoritama</i> .....	38
5.3.2. <i>U igri protiv čovjeka i samoga sebe</i> .....	39

5.4.	IGRE LOCKDOWN VS. JUNIOR I JUNIOR VS. LOCKDOWN.....	39
5.5.	IGRAČ DUMBO .....	39
<b>6.</b>	<b>ZAKLJUČAK .....</b>	<b>40</b>
<b>7.</b>	<b>LITERATURA .....</b>	<b>42</b>
A)	Dodatak: Učitavanje i izvršavanje testnog primjera putem sučelja (dame).....	43
B)	Dodatak: Učitavanje i pregled spremljene igre putem sučelja (dame) .....	44
C)	Dodatak: Učitavanje i pregled spremljene igre putem sučelja (halma) .....	45
D)	Dodatak: Lokacije spremljenih turnira i pojedinih igara na priloženom DVD-u.....	46

INTERNI DOKUMENT

# 1. UVOD

## 1.1. O umjetnoj inteligenciji

Kada govorimo o umjetnoj inteligenciji, zapravo govorimo o nečemu što je osmislio ljudski intelekt i izradila ljudska ruka, a istovremeno pokazuje neke karakteristike koje se generalno pripisuju čovjeku. Postoji više definicija inteligencije, no većina se slaže da se inteligencija tiče sposobnosti rješavanja problema u promjenjivim okolnostima. Pokazuje li neki entitet (živo biće ili neživi objekt) sposobnost prilagodbe ponašanja u odnosu na novonastalu situaciju, možemo s velikom sigurnošću reći da je inteligentan.

U svrhu boljeg razumijevanja rada dao bih nešto konkretniju definiciju: Inteligencija je sposobnost entiteta usmjerenog k cilju da uslijed promjenjivih okolnosti pronađe put do istog.

U kontekstu ovog rada izrađeno je nekoliko igrača za igre dame (engl. *checkers*) i halmu (izvorno *Stern-halma* prema njem. *Stern* – zvijezda i grč. *halma* ili *ἄλμα* – skok, a također i naziv slične igre za 4 igrača koja potječe iz SAD-a; na engleskom jeziku također je poznata pod nazivom *Chinese checkers*). Cilj svakog od tih igrača jest dobiti partiju igre, dakle radi se o entitetima usmjerenim k cilju. Promjenjivost okolnosti osigurana je postojanjem protivničkih igrača – oni neovisno o igraču mijenjaju poziciju na ploči na način koje on mora anticipirati. K tome, moguće je da u sljedećoj partiji na mjesto protivnika "sjedne" novi igrač koji koristi drukčiju taktiku.

Ovdje predstavljeni igrači, međutim, nisu inteligentni. Pojavi li se igrač koji je objektivno jači od njih, nijedan od igrača neće biti sposoban prilagoditi svoju igru ne bi li izvojevao pobjedu, ili barem izjednačenje u nekoj od narednih partija. Ono što bi možda moglo utjecati na ishod je povećanje vremena koje igrač može iskoristiti za proračun poteza, ili ugradnja bržeg procesora. Oboje može značiti da će time i protivnik postati jači, no to nema veze s inteligencijom.

Pogledamo li ljude i njihov način igre, možemo uočiti da neki ljudi, kao i ovdje opisani igrači, neće uspjeti prilagoditi svoj način igre – ili postizanja bilo kakvih ciljeva – novonastalim okolnostima, na način da postignu pobjedu. Pratimo li gore opisanu logiku, moglo bi se zaključiti da stoga ni ljudi nisu inteligentni. To je, naravno, pogrešno.

Podsjetimo se da je inteligencija *sposobnost* da se pronađe put do cilja. U računalni algoritam može se ugraditi sposobnost prilagodbe. Što je algoritam općenitiji, to će se program bolje prilagođavati na širem spektru okolnosti. Međutim, algoritam ne može birati hoće li, ili neće, u danom trenutku iskoristiti neku sposobnost. U tome je velika razlika između računala i čovjeka, odnosno između umjetne inteligencije i one "prirodne" kakvu posjeduje čovjek. Dok svaki zdrav čovjek zasigurno posjeduje sposobnost prilagodbe

novonastalim situacijama, on također posjeduje i mogućnost odabira – hoće li u danoj situaciji iskoristiti tu sposobnost, sve svoje znanje, sve svoje resurse, do krajnjih granica izdržljivosti, ne bi li rješio kakav problem i time postigao svoj cilj. Računalni algoritam uvijek, i u svakoj situaciji, koristi sve svoje sposobnosti, sve svoje znanje i sve svoje resurse, i to na upravo onakav način kakav je programer prevideo, ne bi li proizveo traženi rezultat.

"Inteligencija" algoritma tako je uvijek ograničena općenitošću rješenja, programerovim znanjem i generalno na rješavanje samo jednog problema. Čovjek nema takvih ograničenja. Na kraju, "Inteligencija nije isključivi monopol genijalca; ona je atribut svakog čovjeka i razlike su samo po pitanju stupnja" [7].

## 1.2. Motivacija

Od samih početaka razvoja igre dame, pa do današnjeg okruženja MGMTM gdje je igra dame samo jedna komponenta, motivacija je uvijek bila omogućiti studentima i ostalim interesentima područja umjetne inteligencije da isprobaju vlastite sposobnosti programiranja i eventualno inovacija na tom području. Igra dame razvijala se prva iz jednostavnog razloga što je na kolegiju strojnog učenja bila prikazivana kao jedan od primjera na kojem je moguće iskoristiti tehnike strojnog učenja pri izradi igrača.

Zanimacija studenata za sudjelovanje u takvom projektu izradom igrača – čija je snaga ispitana u turniru protiv igrača ostalih studenata – služila je kao daljnja motivacija, najprije za poboljšanje sučelja konkretno za dame, a zatim i izrade općenitog sučelja putem kojeg je moguće organizirati turnir u bilo kakvoj igri.

## 1.3. Cilj rada

Kao rezultat ovog rada očekuje se, osim manjih prilagodbi već postojeće igre dame, izrada nove igre sa vlastitim sučeljem. Odabrana je igra halma kao logičan sljedbenik igre dame s kojom – uza sve razlike – ima nekih sličnosti. Zahtijeva se i da se obje igre mogu pokrenuti unutar okruženja MGMTM.

U izradu nove igre planiralo se krenuti još prije nego je MGMTM bio razvijen, s razlogom da se održanim turnirima da malo različitosti. Konkretno, halma je odabrana jer se činila kao logičan izbor nakon dama. Pravila igre su dosta jednostavna, a igrači algoritmi mogu biti itekako kreativno izvedeni. K tomu, obzirom da je igra predviđena za do 6 igrača, ne može se upotrijebiti klasični algoritam minimax.

Uz igre, izrađuju se dva igrača za igru dame i jedan za halmu. Igrači za dame, osim kao pokušaj da se naprave bolji igrači od već postojećih iz prethodnih turnira, izrađuju se ne bi li budući studenti mogli prije samog turnira odmjeravati snage u igri protiv njih i time na neki način znati je li njihov igrač imalo dobar.

Za jednog od igrača dame izrađena je ablacijska studija opisana u poglavlju 5.2.1 te je obavljeno vrednovanje svih igrača kroz igranje turnira, igara samih protiv sebe i igara protiv čovjeka.

Svakako se nadam da će provedena studija biti od pomoći budućim sudionicima izazova (u izradi igrača) pri odabiru vlastite heuristike te da će rezultati traženja neuronske mreže genetičkim algoritmom, kao i par navoda o prethodnim neuspjelim pokušajima, pomoći pri odabiru topologije neuronske mreže, kao i načina njenog treniranja.

Naravno, tu je i moje zanimanje za područje umjetne inteligencije, na kojem sam svoje prve samostalne korake napravio upravo u ovom diplomskom radu. Svakako je jedan od bitnih ciljeva ovog rada – bez kojeg ga ne bi bilo – bio zadovoljenje svoje želje za znanjem o principima i praksama na ovom području.

#### **1.4. Struktura rada**

Nakon uvoda i kratke teorijske pozadine (opisa korištenih algoritama), u poglavlju 3 daje se kratak opis okruženja MGMTM, i to njegovog sučelja prema korisniku, ali će se reći i nekoliko riječi o programerskim sučeljima, njihovim nedostacima i prednostima.

U poglavlju 4 detaljnije su opisane izrađene igre te su navedena pravila kojih se algoritmi moraju pridržavati. Uz svaku izrađenu igru, nabrojani su i detaljnije opisani igrači koji su izrađeni za tu igru u okviru ovog diplomskog rada.

U poglavlju 5 opisan je način evaluacije igrača te su prikazani rezultati studija, i to prvenstveno za igrača Lockdown, s čijom je heuristikom provedena ablacijska studija. Također je prikazano i analizirano nekoliko zanimljivijih situacija koje su nastale na ploči tokom igre.

## 2. ALGORITMI KORIŠTENI ZA IZRADU IGRAČA

Kod igrača dama, za pretragu prostora stanja korišten je isključivo algoritam minimax i podrezivanje alfa-beta. Igrači se međusobno razlikuju po načinu izračuna heurističkih vrijednosti pozicija. Algoritam minimax je izveden s manjim modifikacijama kako bi se prilagodio okružju u kojem igrač na raspolaganju ima određeno vrijeme, odnosno tako da igrač vrati potez unutar njemu dodijeljenog vremena.

Prva modifikacija ticala se direktno vraćanja poteza u roku, a sastoji se u tome da igrač prestaje izvršavati algoritam kada je vrijeme za vraćanje poteza. Obzirom da se može dogoditi da je relativno brzo potreban bilo kakav potez, algoritam je izveden s iterativnim produbljivanjem pretrage, odnosno najprije se pretražuje prostor stanja do dubine 2, zatim 4, pa 6, itd. Koraci promjene dubine uvijek su parni ne bi li se na igračev potez analizirao i protivnikov odgovor – inače se potez može činiti mnogo boljim nego što zaista jest [1].

Ne bi li se do najboljeg poteza došlo što je prije moguće, učinjene su još dvije modifikacije. Prva se sastoji u tome da igrač iz poteza u potez pamti koju je figuru posljednju pomaknuo. Pretraživanje prostora stanja uvijek počinje upravo od nje (osim ukoliko u sljedećem krugu nije više moguće učiniti potez s tom figurom, ili ju je protivnik prikladnim potezom maknuo s ploče). To dovodi do bržeg pronalaženja dobrog poteza u situacijama u kojima je potrebno napraviti više uzastopnih poteza istom figurom, a pogotovo ako se već u drugom potezu bitno povećava vrijednost pozicije. Ta modifikacija koristi se isključivo za biranje početka pretrage algoritmom minimax u prvoj iteraciji (do dubine 2) i nije nužno da će se u drugoj iteraciji pretrage (do dubine 4) također početi pretraživati potezi iste figure.

Tome je zaslužna druga modifikacija koja nakon svake iteracije, tj. nakon završene pretrage stabla do trenutne maksimalne dubine sortira pronađene poteze prema vrijednostima heuristika pozicija kojima oni rezultiraju. Tako se potezi s najboljom heuristikom iz prethodne iteracije prvi pretražuju u sljedećoj.

Radi uštede vremena, a pogotovo u igrama u kojima se vrijeme mjeri po cijeloj partiji (umjesto po potezu), ukoliko igrač ima samo jedan potez, tada on odmah vraća taj potez, bez ulaska u pretragu minimax. Također, ukoliko minimax u nekom trenutku pronađe siguran put do pobjede, odmah se vraća potez koji do nje vodi, bez daljnje pretrage. Međutim, pronađeni put se ne pamti nego će algoritam u sljedećem svom potezu ponovo morati pretražiti stablo ne bi li pronašao isti put. Obzirom da će taj put sada biti kraći za minimalno jedan potez svakog igrača, to će brže algoritam ponovo pronaći taj put, pa nije bilo potrebno raditi dodatnu optimizaciju pamćenjem (to više što protivnik može napraviti neki drugi, tj. lošiji potez, koji će rezultirati bržom pobjedom igrača).

Slijede opisi svih algoritama koji su korišteni pri izradi igrača, u kojima su objašnjene eventualne varijacije algoritma koje su ovdje korištene.

## 2.1. Algoritam minimax

Algoritam minimax je algoritam za pretragu prostora stanja prilagođen igrama za dva igrača. Algoritam započinje pretragu s jednim od svih mogućih poteza igrača za kojeg se traži potez te potom analizira sve moguće protivnikove odgovore. Na svaki protivnikov odgovor ponovo se pretražuje stablo prostora stanja za najbolji sljedeći potez igrača. Zaustavljanje takve pretrage u dubinu događa se kada je ispunjen jedan od dva uvjeta:

- Pronađena pozicija na ploči predstavlja kraj igre (pobjedu za jednog od igrača) ili
- Dosegnuta je maksimalna dubina pretrage.

U oba ta slučaja pozicija se ocjenjuje heurističkom funkcijom. Očito, pobjeda igrača mora biti ocijenjena najvećom mogućom ocjenom, a poraz najmanjom mogućom, tako da igrač uvijek izbjegava poraz, a kreće se prema pobjedi.

Slijedi pseudokod algoritma minimax:

```
Minimax(PozicijaIgre poz) {
    return Max(poz);
}

Max(PozicijaIgre poz) {
    if (KrajIgre(poz)) return OcijeniPoziciju(poz);
    else {
        najbolji_potez <- {};
        svi_potezi <- GenerirajPoteze(poz);
        za svaki potez u svi_potezi {
            t_potez <- Min(UciniPotez(poz, potez));
            if (Vrijednost(t_potez) > Vrijednost(najbolji_potez)) {
                najbolji_potez <- t_potez;
            }
        }
        return najbolji_potez;
    }
}

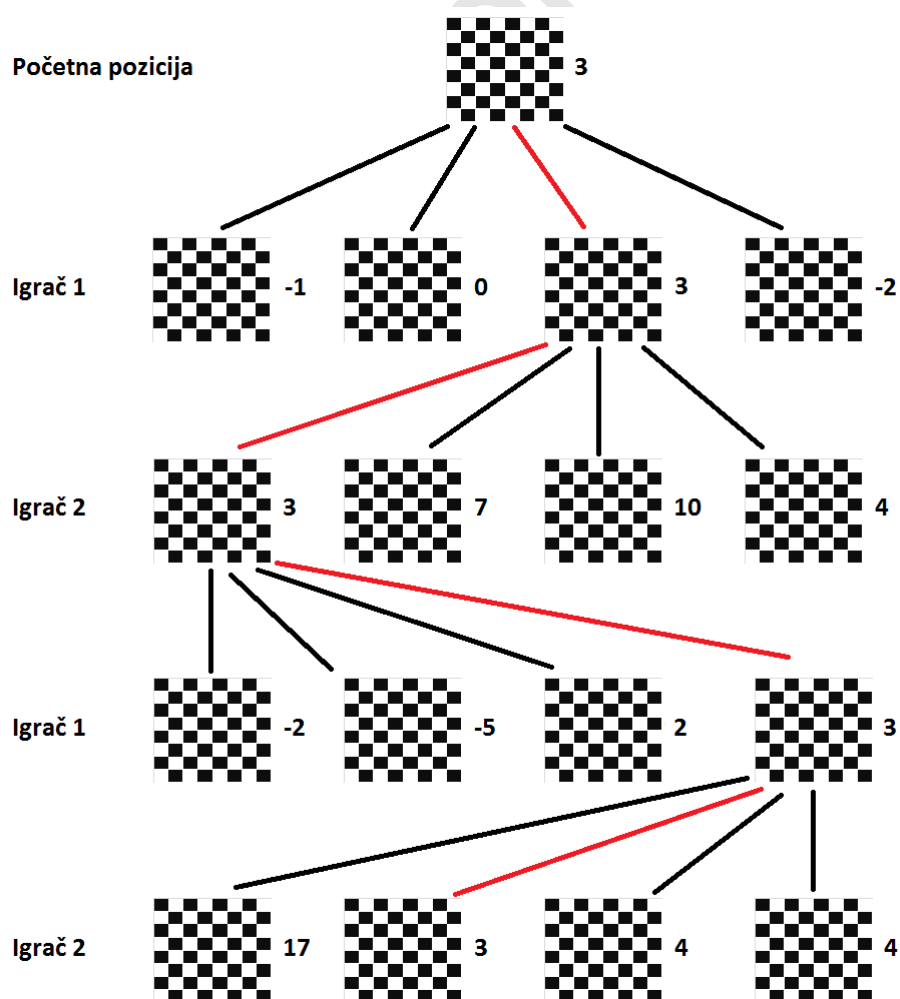
Min(PozicijaIgre poz) {
    if (KrajIgre(poz)) return OcijeniPoziciju(poz);
    else {
        najbolji_potez <- {};
        svi_potezi <- GenerirajPoteze(poz);
        za svaki potez u svi_potezi {
            t_potez <- Min(UciniPotez(poz, potez));
            if (Vrijednost(t_potez) < Vrijednost(najbolji_potez)) {
                najbolji_potez <- t_potez;
            }
        }
        return najbolji_potez;
    }
}
```

Naravno, nerealno je očekivati da će algoritam pronaći pobjedu već s početne pozicije, što na dobru heuristiku postavlja uvjet koji ona mora ispunjavati da bi igrač bio uspješan, a to je da stanja ocjenjuje

ispravno, odnosno ona stanja koja sigurno vode do pobjede mora ocjenjivati visokom ocjenom, a ona stanja koja vode do poraza mora ocjenjivati malom, čak negativnom ocjenom.

Kod jednostavnih igara, poput 3x3 križić-kružić, svako se stanje može ručno ocijeniti prikladnom ocjenom, ili se već iz početnog stanja mogu predvidjeti svi mogući ishodi. Problem nastaje pri velikim igrama s velikim brojem stanja, poput šaha, ali i dama. Za izradu heuristike u tim igrama obično se vrijednosti računaju prema određenim značajkama stanja, kao što su broj figura igrača i njegovog protivnika, i slično (ovisno o igri). Osim toga, za heurističku procjenu ocjene stanja može se koristiti i neuronska mreža (vidi 2.3). Tako je Tesauro razvio igrača za trik-trak (engl. *backgammon*) evaluirajući pozicije prethodno treniranom neuronskom mrežom [9], a Chellapilla i Fogel genetičkim su algoritmom (vidi 2.4) pronašli neuronsku mrežu koja uspješno igra dame [1].

Zanimljivo pitanje koje se postavlja kod izrade algoritma minimax jest kako on koristi takve heurističke procjene da bi pronašao najbolji potez na danoj poziciji (uz pretpostavku da heuristika zadovoljava gore navedeni uvjet). Grafički prikaz rada algoritma minimax prikazuje Slika 1.



Slika 1. Grafički prikaz rada algoritma minimax.

Kod proučavanja slike vrijedi držati na umu da minimax pretražuje poteze u dubinu. Kada je zadovoljen jedan od uvjeta zaustavljanja (navedeni gore), pravi se heuristička procjena pozicije i prenosi se u prethodnu dubinu. Radi pojednostavljenja, pretpostavimo da algoritam pretražuje do dubine 4, da se radi o igri dame te da je igrač bijele boje (prema sučelju razvijenom u okviru ovog rada). Također, neka dubina 0 predstavlja pretragu prvog poteza, dubina 1 protivnikov odgovor, dubina 2 igračev drugi potez, a dubina 3 protivnikov odgovor na igračev drugi potez. Dubina 4 (odnosno igračev treći potez) se ne analizira, već se jednostavno računa heuristika za rezultirajuću poziciju i vraća se na prethodnu dubinu.

Cilj bijelog igrača je pronaći onaj potez koji će rezultirati što boljom heuristikom, bez obzira na poteze protivnika, dok je cilj protivnika pronaći onaj potez koji će rezultirati što lošijom heuristikom za bijelog igrača (odnosno boljom za sebe). Zato se minimax obično radi s dvjema funkcijama – jedna pokušava maksimizirati heuristiku, dok ju druga pokušava minimizirati – te se prvo poziva maksimizirajuća (igrač za sebe pokušava pronaći najbolji potez). Pretraga pretpostavlja simetričnu heuristiku (bolja ocjena za bijelog igrača znači isto toliko goru ocjenu za crnog), iako to ne mora biti slučaj.

Kada se s dubine 4 (bijeli) u dubinu 3 (crni) vrata heurističke vrijednosti pozicija koje rezultiraju za svaki mogući potez crnog igrača, algoritam se nalazi u minimizirajućoj funkciji, što znači da će funkcija vratiti onu heuristiku (i ekvivalentan potez) koja ima najmanju vrijednost. Slično na dubini 2 (bijeli) u maksimizirajućoj metodi, pozivom minimizirajuće funkcije očekuju se za bijelog igrača najbolje dosegljive heuristike za svaki od mogućih poteza na danoj poziciji. Kada ih se pronađe, algoritam traži najbolju među njima i vraća je pozivajućoj metodi (minimizirajućoj funkciji na dubini 1), gdje se ponovo od svih mogućih bira najmanja heuristika te se vraća u maksimizirajuću funkciju na dubini 0. Kada završi maksimizirajuća funkcija na dubini 0 sa svojim radom analizirani su svi mogući potezi (do zadane dubine) te se na temelju vraćenih heuristika može zaključiti o najboljem mogućem potezu.

Uzmimo za konkretan primjer izvođenje algoritma minimax kojeg prikazuje Slika 1. Slika prikazuje pretragu do dubine 4, pa pretpostavimo za sve pozicije koje nisu ekspanzirane da su bile prethodno ekspanzirane i da su pronađene heuristike kakve su prikazane. Igrač na danoj poziciji pretražuje sve poteze u svojoj maksimizirajućoj funkciji. Za tri pozicije već ima najbolju garantiranu dosegljivu heuristiku redom -1, 0 i -2. Sada analizira još jedan potez te poziva minimizirajuću funkciju s rezultirajućom pozicijom kao parametrom. Minimizirajuća funkcija pronađe 4 moguća poteza te za prva tri procijeni da su heuristike 7, 10 i 4. Sada analizira posljednji potez te poziva maksimizirajuću funkciju. Maksimizirajuća funkcija za tri moguća poteza dobiva heuristike -2, -5, 2 te za posljednji opet poziva minimizirajuću funkciju. Dolazi se do posljednje dubine. Za svaki mogući potez generira se rezultirajuća pozicija te se (obično još jednim pozivom maksimizirajuće funkcije) samo traži heuristička vrijednost pozicije (bez daljnje pretrage). Kao što slika prikazuje dobile su se heuristike 17, 3, 4 i 4. Obzirom da se nalazimo u minimizirajućoj funkciji, odabire se najmanja moguća heuristika (tj. najbolja moguća za igrača 2) te se

vraća u maksimizirajuću funkciju na prethodnom nivou. U maksimizirajućoj funkciji, prema prethodno pronađenim i novopristiglom rezultatu, određuje se da je najbolja moguća heuristika 3 (jer ova funkcija nastoji pronaći najbolji mogući ishod za igrača 1) te se rezultat vraća u minimizirajuću funkciju na nivou 2. Minimizirajuća funkcija određuje da je 3 najgori mogući ishod za igrača 1 te vraća taj rezultat na prvi nivo u maksimizirajuću funkciju. Tu se napokon određuje najbolji mogući potez, slanjem najbolje moguće heuristike u funkciju koja je započela cijelu pretragu minimax algoritmom.

Termin "najbolji mogući potez" ovdje je potrebno uzeti sa zadržkom. Dubina 4 relativno je "plitko" pretraživanje stabla i vrlo lako se može dogoditi tzv. efekt horizonta [8]. Efekt horizonta naziva se događaj tokom pretrage prostora stanja kada je neki loš, ali neminovan ishod za igrača dosegljiv u pretraženom prostoru, no određenim potezom (obično žrtvom slabije figure) on se može odgoditi do dubine do koje se pretraga nije uspjela izvršiti. To rezultira gubitkom obje figure, tj. još lošijom pozicijom nego da je igrač prvotno izgubio samo jaču figuru.

Na primjer, pretpostavimo da igramo šah te naš igrač na dubini 4 pronađe potez u kojem gubi kraljicu, a pretražuje se prostor stanja do dubine 6. Međutim, daljnjom pretragom pronađe da ukoliko na dubini 2 izgubi kulu, neće izgubiti kraljicu u pretraženom prostoru stanja. Ovdje je moguć efekt horizonta ukoliko ponudivši kulu da spasi kraljicu, igrač izgubi i kraljicu na dubini 8 – potez koji u tom trenutku nije vidio.

Efekt horizonta nije uvijek moguće riješiti, a obično se umanjuje vjerojatnost njegove pojave tako da se u ovakvim i sličnim opasnim situacijama, dozvoli pretrazi minimax da prekorači zadanu maksimalnu dubinu u određenim situacijama [8]. Problem se time generalno ne rješava, jer (osim ukoliko se pronađe kraj igre) pretraga mora stati usred igre [4].

Algoritam minimax može se izreći teoremom kojeg je postavio matematičar John von Neumann [6] još 1928. godine, a koji glasi:

*Za svaku zero-sum igru za dva igrača s konačnim strategijama, postoji vrijednost  $V$  i miješana strategija za svakog igrača, takva da vrijedi: a) Za danu strategiju igrača 2, najbolja moguća vrijednost za igrača 1 je  $V$  te b) Za danu strategiju igrača 1, najbolja moguća vrijednost za igrača 2 je  $-V$ .*

Drugim riječima, određena strategija igrača 1 garantira mu dobivenu vrijednost  $V$ , bez obzira na strategiju igrača 2, a igraču 2 za njegovu strategiju garantirana mu je vrijednost  $-V$ , bez obzira na strategiju igrača 1.

## 2.2. Podrezivanje alfa-beta

Podrezivanje alfa-beta (engl. *alpha-beta pruning*) naziv je za cijelu klasu tehnika optimizacije pretrage prostora stanja, a može se iskoristiti i kod pretrage minimax [4] [8]. U ovom radu u svim izvedbama algoritma minimax, koristila se optimizacija podrezivanjem alfa-beta.

Podrezivanje alfa-beta kao optimizacija algoritma minimax izvodi se tako da se u standardnu izvedbu algoritma minimax kao parametri šalju se dvije nove varijable, alfa i beta. Konceptualno, vrijednost alfa predstavlja najbolju dosegljivu heuristiku za maksimizirajuću funkciju (tj. uz pretpostavke iz prethodnog dijela, za bijelog igrača) u potpunom stablu prostora stanja. Vrijednost beta pak predstavlja najbolju dosegljivu heuristiku u pojedinom podstablu generiranom odabirom početnog poteza za minimizirajuću funkciju (crnog igrača).

Ukoliko neki poziv minimizirajuće ili maksimizirajuće funkcije vrati bolju pronađenu heuristiku, osvježava se beta, odnosno alfa vrijednost. Ukoliko se u nekom trenutku dogodi da je vrijednost alfa postala veća ili jednaka vrijednosti beta, zaustavlja se pretraga na toj razini.

Radi jednostavnijeg objašnjenja zašto je to tako, pretpostavimo da je bijeli igrač pronašao potez koji ga može dovesti do pozicije čija je heuristička vrijednost 10 (vrijednost alfa). Pretraživajući svoj sljedeći mogući potez, minimizirajuća funkcija otkrila je odgovor koji može bijelog igrača dovesti do pozicije čija je heuristička vrijednost 5 (vrijednost beta). Dakle, ukoliko bijeli učini taj potez, protivnik ga može dovesti u situaciju koja je gora od najgore situacije u kojoj bi se našao da je igrao potez koji je analizirao ranije. Za bijelog igrača to znači da nema smisla dalje analizirati ovu granu, jer je očito da može završiti u lošijoj situaciji nego u nekoj od prethodno analiziranih grana, što nije poželjno.

U tome se sastoji cijela logika podrezivanja alfa-beta – zaustaviti pretraživanje grana koje dokazano vode do lošijih situacija nego neka druga, već analizirana grana. Ovakva optimizacija bitno ubrzava algoritam minimax, a najgori slučaj, u kojem će se izvršiti cijeli minimax, je onaj u kojem se potezi slučajno poredaju baš na način da se najprije analiziraju najlošiji potezi. U ovom kontekstu trebali bi biti jasni razlozi zbog kojih se dodatni trud u ovom radu uložio u sortiranje poteza prema dobivenim heurističkim vrijednostima.

### **2.3. Neuronske mreže**

Da bismo razumjeli neuronske mreže, potrebno je najprije razmotriti funkcioniranje pojedinačnog neurona kao njenog sastavnog dijela. Ideja modeliranja neurona i korištenja takvog modela u računalnoj znanosti potekla je iz psiholoških istraživanja učenja u životinja [4]. Relativno jednostavni model neurona zasniva se na stvarnom biološkom neuronu koji se sastoji od određenog broja dendrita, svaki sa svojim skupom sinaptičkih ulaza, soma ili tijela neurona s jezgrom te aksona sa sinaptičkim završecima. Sinaptički ulazi na dendritima su "ulazi" u neuron. Određeni podražaj ulaza "okida" neuron na način da se pojavi impuls na sinaptičkim završecima aksona. Drugim riječima, neuron preko svojih ulaza prima određeni podražaj, nad svim primljenim podražajima obavlja nekakvu funkciju u tijelu te daje rezultat te funkcije na izlazu, odnosno završecima aksona [3].

Model neurona vrlo je sličan opisanome, iako mnogo jednostavniji. Svakom ulazu u model pridijeljena je određena težina kojom se množi podražaj na tom ulazu. Svi tako množeni ulazi zbrajaju se i rezultat je ulazni parametar za nekakvu funkciju koju nazivamo aktivacijskom funkcijom. Aktivacijska funkcija može biti bilo kakva, a neke od češćih su:

- Funkcija skoka,  $f(x) = \begin{cases} 1, v \geq 0 \\ 0, v < 0 \end{cases}$
- Identitet,  $f(x) = x$
- Aktivacijska funkcija linearna po odsječcima,  $f(x) = \begin{cases} 1, v \geq 0.5 \\ v + 0.5, -0.5 < v < 0.5 \\ 0, v \leq -0.5 \end{cases}$
- Sigmoida,  $f(x) = \frac{1}{1+e^{-ax}}$
- Simetrična sigmoida,  $f(x) = \frac{2}{1+e^{-x}} - 1$
- Radijalna bazna funkcija (RBF),  $f(x) = e^{-\alpha x^2}$ .

Ovdje je  $x = \sum_i w_i x_i$ , gdje su  $w_i$  težine, a  $x_i$  ulazi u neuron. Rezultat aktivacijske funkcije pojavljuje se na izlazu neurona.

Neuronske mreže dobivaju se spajanjem izlaza neurona na ulaz nekog drugog neurona. Tako se neuronske mreže mogu prikazati pomoću orijentiranih grafova. Obično se skupovi neurona slažu u slojeve neuronske mreže, pa su izlazi neurona s jednog sloja ulazi u sljedeći sloj. Radi jednostavnijeg označavanja, obično se ulazi u neuronsku mrežu promatraju kao dodatni (ulazni) sloj neurona koji na izlazu daju istu vrijednost kao što je primaju na (jedinom) ulazu. Posljednji sloj naziva se izlaznim, a svi slojevi između ulaznog i izlaznog nazivaju se skrivenim slojevima.

S obzirom na način spajanja neurona u mreži, neuronske mreže mogu se podijeliti na četiri vrste:

- Jednoslojne mreže bez povratnih veza (engl. *single-layer feedforward networks*) u kojima je prvi sloj izlazni, a ulazni se ne broji jer nema procesiranja;
- Višeslojne mreže bez povratnih veza (engl. *multi-layer feedforward networks*) koje osim ulaznog i izlaznog imaju jedan ili više skrivenih slojeva;
- Mreže s povratnim vezama (engl. *recurrent networks*) imaju barem jednu povratnu vezu, tj. izlaz barem jednog neurona iz  $n$ -tog sloja spojen je na ulaz jednog od neurona u  $m$ -tom sloju, gdje je  $m \leq n$ ;
- Ljestvičaste mreže (engl. *lattice structures*) u kojima je svaki ulaz spojen na sve neurone koji su raspoređeni u  $n$ -dimenzionalnom polju, bez povratnih veza, sa skupom izvornih čvorova koji daju ulazne signale.

U ovom radu korištena je potpuno povezana višeslojna neuronska mreža bez povratnih veza (svi izlazi sloja  $n$  spojeni su na jedan od ulaza svakog neurona u sloju  $n + 1$ ).

Neuronska mreža vrlo dobro može aproksimirati neku nepoznatu nelinearnu funkciju. Kod igre dame, nepoznata funkcija je točna vrijednost heuristike za danu poziciju na ploči. Neuronska mreža može postepeno učiti određenim algoritmima (odnosno izlaz neuronske mreže može se prilagođavati postepenim mijenjanjem težina na ulazima neurona tako da za dani primjer dobijemo željeni izlaz). Jedan od njih je algoritam *backpropagation*, koji se inicijalno namjeravao koristiti za učenje neuronske mreže. Zbog velikog broja izjednačenih partija u toku učenja, koje je vrlo teško ispravno ocijeniti, učenje je zapelo u lokalnom minimumu i u daljnjim partijama igrač je igrao jednu te istu izjednačenu partiju, istovremeno igrajući veoma loše naspram ostalih dostupnih igrača.

## 2.4. Genetički algoritam

Umjesto učenja neuronske mreže algoritmom *backpropagation*, odabran je genetički algoritam, koji se u kombinaciji s neuronskom mrežom već ranije pokazao kao dosta uspješnim rješenjem [1]. Genetički algoritam može se promatrati kao algoritam koji se koristi za pretragu velikog prostora stanja kakvog klasični algoritmi pretraživanja ne bi mogli pretražiti u doglednom roku.

U korištenoj varijaciji algoritma pretraživale su se težine neuronske mreže uz koje neuronska mreža na svome izlazu daje najbolju aproksimaciju heurističke funkcije za danu poziciju. Više o ovoj konkretnoj izvedbi biti će spomenuto u poglavlju 4.1.4, a ovdje će se navesti samo općeniti načini izvedbe genetičkog algoritma.

Poput neuronskih mreža, genetički algoritam je zasnovan na istraživanjima iz biološkog svijeta [4] [8]. Konkretno, ovdje se radi o teoriji evolucije, gdje se jedinke u velikoj populaciji bore za preživljavanje uz reprodukciju kojom se potomcima prenosi dio genetičkog koda te oni postaju bolje prilagođeni svojoj okolini i imaju više šanse za preživljavanje.

Da bi se u implementaciji genetičkog algoritma ostvarilo takvo napredovanje jedinki, potrebno je implementirati nekoliko ključnih dijelova algoritma:

- Geni – potrebno je dobro razmisliti u pojedinoj implementaciji na koji način je najbolje implementirati genetički kod jedinke ne bi li pronalaženje rješenja uopće bilo moguće. Vrlo česta implementacija je predstavljanje gena nizom bitova.
- Uvjet zaustavljanja – izvođenje genetičkog algoritma može potrajati više dana, pa čak i tjedana. Prije početka određuje se kada će se izvođenje zaustaviti. Neki od čestih uvjeta zaustavljanja su jednostavno zaustavljanje nakon određenog broja generacija, ili nakon što se u više generacija nije bitno promijenio genetički materijal (npr. najbolja jedinka ostala je ista veći broj generacija).

- Funkcija dobrote (engl. *fitness*) – način ocjene jedinke prema kojem se bira hoće li ona ući u sljedeću generaciju, odnosno hoće li biti korištena kod križanja.
- Križanje (engl. *crossover*) – način na koji će se nove jedinke generirati iz starih jedinki (svojevrсна reprodukcija). Križanje je genetički operator kojim se iz odabranih jedinki prethodne generacije generira nova jedinka u sljedećoj generaciji. Obzirom da genetički algoritam pretražuje veliki prostor stanja, a pokreće ga osoba koja traži jedinku koja rješava dani problem (najbolje aproksimira neku nepoznatu funkciju), križanje služi da bi se iz dvije jedinke koje se nalaze u blizini lokalnog minimuma pogreške procjene te funkcije stvorila nova jedinka, također u blizini ili u samom minimumu, uz nadu da će mu se nalaziti bliže (odnosno, ciljnu funkciju će bolje aproksimirati).
- Mutacija – ovo je drugi genetički operator, a predstavlja nasumičnu promjenu genetičkog zapisa ili jednog njegovog dijela. U kontekstu genetičkog algoritma služi ne bi li novu jedinku izbacio iz lokalnog minimuma funkcije pogreške u kojem se sakupio veći broj jedinki te je eventualno preselio u neki drugi minimum ili njegovu blizinu. Time se povećava vjerojatnost da genetički algoritam na kraju svog izvršavanja pronađe globalni minimum.
- Prelazak u novu generaciju – potrebno je također odrediti način odabira jedinki iz jedne generacije koje će se prenositi u sljedeću generaciju, kao i jedinke koje će se koristiti za križanje. Tu važnu ulogu može igrati elitizam, kojim se najbolja jedinka (ona s najboljom ocjenom funkcije dobrote) uvijek prenosi iz jedne generacije u drugu. Način odabira ostalih može se vršiti na različite načine. Primjerice, moguće je jednostavno prenijeti najbolje jedinke iz stare generacije u novu. Još jedna efikasna metoda je tzv. rulet, u kojoj se na brojevnom pravcu (kotaču ruleta) dodijeljuju rasponi vrijednosti jedinkama. Što je veća dobrota jedinke, veći je i njoj pridijeljeni raspon na pravcu. Zatim se nasumično odabire jedna vrijednost s pravca i jedinka kojoj je ta vrijednost unutar raspona vrijednosti prelazi u novu generaciju.
- Odabir parametara – nakon što je algoritam implementiran, potrebno je pažljivo odabrati parametre, i to broj jedinki u pojedinoj generaciji, broj jedinki koje se nepromijenjene prenose iz jedne generacije u sljedeću, vjerojatnost mutacije kompletnog kromosoma, ili vjerojatnost mutacije jednog gena, i ostalo. Ti parametri mogu bitno utjecati na brzinu pronalaska optimalne jedinke.

Genetički algoritam u srži je dosta jednostavan, odnosno kako slijedi u pseudokodu:

```

generacija = 0;
inicijaliziraj populaciju P(generacija);
dok nije zadovoljen uvjet završetka algoritma ponavlja
    funkcijom dobrote procijeni svakog člana populacije P(generacija);
    odaberi članove P(generacija) za prelazak u novu generaciju;
    iz tako odabranih jedinki genetičkim operatorima stvori nove jedinke;
    odbaci stare jedinke na osnovu funkcije dobrote i zamijeni ih novima;
++generacija;

```

## 2.5. Opis algoritma igrača za igru halma

Obzirom je igra halma u općenitom slučaju za do šest igrača, igrač za tu igru nije mogao biti izveden klasičnim algoritmom minimax. Umjesto njega, izveden je novi algoritam koji je morao zadovoljavati sljedeće uvjete:

- Algoritam mora u kratkom roku biti u stanju vratiti potez koji se čini najboljim barem na dubini jedan;
- Mora se ispravno izvršavati bez obzira na broj igrača i njihov razmještaj;
- Pri dužim vremenima izvršavanja mora moći pronaći bolje poteze.

Izveden je algoritam koji funkcionira na sljedeći način:

1. Najprije ispitaaj sve poteze koje mogu napraviti te za svaki izračunaj heuristiku na rezultirajućoj poziciji;
2. Ako je dosegnuta maksimalna dubina ili kraj igre, vrati najbolji potez, inače ako je dubina 0 postavi najbolji do sada pronađeni potez na najbolji potez;
3. Ako ima poteza koji povećavaju heurističku vrijednost pozicije, tada izbacij iz analize sve poteze koji je smanjuju;
4. Za svaki preostali potez rekurzivno zovi ovu funkciju za sljedećeg igrača na ploči, uz povećanje trenutne dubine;
5. Vрати najbolji pronađeni potez.

Algoritam odgovara na prvi zahtjev, odnosno u relativno kratkom roku može vratiti dobar potez, uz dobru heuristiku. Također odgovara i na drugi zahtjev, jer je metoda "sljedeći igrač" implementirana u klasi koji predstavlja poziciju na ploči. Svaka instanca ploče "zna" na temelju parametra prosljeđenog u tu metodu, koji igrač je sljedeći na potezu, obzirom da je po ispoštivanim pravilima redosljed igrača u smjeru kazaljke na satu.

Da bi ispunio treći uvjet, algoritam je, poput minimaxa u igri dame, kombiniran iterativnim produbljanjem pretrage stabla.

Treći korak algoritma potrebno je malo objasniti. Igrači halme u svakom potezu imaju opciju napraviti pomak neke od figura prema natrag, odnosno dalje od svog cilja. Budući da su situacije u kojima je takav potez dobar vrlo rijetke, ovime se omogućava (ako je heuristika prikladno izvedena) da se takvi potezi uopće ne analiziraju, čime se barem malo smanjuje faktor grananja.

Iako će dosta brzo imati nekakav potez, algoritam je spor. Problem je što već na početnoj poziciji svaki igrač može napraviti 14 poteza, a na gotovo svaki od njih svaki igrač ima 14 mogućih odgovora. Igra li igru šest igrača, to znači da je broj pozicija koje je potrebno ocijeniti za samo jedan (prvi) potez svakog igrača tek za nekoliko manji od  $14^6 = 7\,529\,536$  pozicija. Očito je da je u ovoj igri klasično pretraživanje

prostora stanja potpuno neučinkovito i sporo, pa će se u budućim radovima biti potrebno okrenuti drukčijim metodama.

INTERNI DOKUMENT

### 3. OKRUŽENJE MGMTM

MGMTM (Multi-Game Multi-Tournament Manager) je programsko okruženje koje omogućava izvođenje turnira igračih algoritama [2]. Iako su ovdje opisane igre izvedene tako da se mogu pokrenuti i u tom okruženju, obje imaju vlastito sučelje te se mogu pokrenuti neovisno o MGMTM-u.

Komunikacija između igre i MGMTM-a odvija se putem sučelja definiranih unutar imenika Interfaces koji je definiran u datoteci Interfaces.dll. Određeni problemi vezani uz povijest razvoja aplikacije onemogućavaju da se igre pokrenu, a da na sustavu ne postoji ta datoteka. U procesu razvoja pronađeno je rješenje za taj problem, no nije implementirano.

MGMTM je izveden dovoljno općenito da je moguće izraditi sučelje koje će spajati MGMTM i bilo koju igru, iako ona nije prvotno bila zamišljena kao dio tog okruženja. Takva rješenja, međutim izlaze iz okvira ovog rada.

#### 3.1. Sučelje prema korisniku

Sučelje MGMTM-a prema korisniku je naspram mogućnosti kompletnog sustava dosta šturo i prati samo nužne funkcionalnosti koje omogućavaju organizaciju, odvijanje praćenja turnira i pregled rezultata. Trenutno je vrlo lako moguće izraditi ekstenzivne promjene na sučelju, bez opasnosti da se turniri koji su već odigrani ranije ne bi mogli učitati u okruženje i pregledati partije igara koje su se igrale.

Potrebno je, međutim, precizno razgraničiti u cijelom sustavu koji dio skupa elemenata korisničkog sučelja zaista potječu od strane MGMTM-a, a koji ne. Naime, sustav dinamički učitava određene elemente te ih prikazuje onakve kakve ih pojedine igre generiraju.

Primjerice, ne bi li se igrama omogućila potpuna sloboda kada nude korisniku odabir postavki igre, kontrola koja služi za odabir postavki mora se dizajnirati unutar projekta same igre. MGMTM nudi samo apstraktnu klasu koju ta kontrola mora nasljeđivati da bi se ispravno prikazala.

Sljedeći element korisničkog sučelja u koje MGMTM ne dira jest prozor za odabir igrača. Ovdje sustav daje igri potpunu slobodu, a od nje očekuje događaj (engl. *event*) kada se igrač želi prijaviti u sustav. Hoće li igra prijaviti igrača koji je prisutan lokalno na istom računalu, ili će primati zahtjeve putem Interneta, u potpunosti je njena briga, kao što je i njena briga da se sva potrebna sučelja za bilo koji model prijave igrača ispravno prikažu na ekranu.

Na kraju, naravno, kada korisnik želi vidjeti igru koja se trenutno odvija na turniru, MGMTM poziva prikladnu metodu putem sučelja, a na igri je da prikaže sučelje.

Iako takva izvedba pomalo otežava izvedbu igre koja će ispravno funkcionirati u ovom okruženju, upravo je zbog nje omogućen pristup razvoju igre koji je u potpunosti neovisan o sustavu MGMTM i

njegovim zahtjevima do trenutka kada kompletna igra i sva korisnička sučelja potrebna za njeno neovisno pokretanje nisu u potpunosti izrađena. U tom trenutku može se, opet potpuno neovisno i uz minimalan trud, razviti sučelje prema MGMTMu koje će omogućiti da se putem tog sustava organiziraju turniri (uz, naravno, uvjet da za igru postoje igrači. Oni se, opet, mogu razvijati u potpunosti nezavisno od sučelja prema sustavu MGMTM.

Dakle, od kontrola koje se prikazuju korisniku, igra mora implementirati sljedeće:

- Sučelje za definiciju postavki igre
- Sučelje za prijavu igrača
- Sučelje za prikaz igre

Opcionalno, svaka igra može po volji otvarati druga sučelja.

### 3.2. Programerska sučelja

Sva komunikacija između igara i MGMTM-a odvija se putem sučelja (engl. *interfaces*) koja klase koje služe za komunikaciju sa sustavom moraju implementirati. Bilo kakve promjene na tim sučeljima značit će da već postojeće igre neće funkcionirati unutar sustava i spomenute klase trebat će se promijeniti u skladu s promjenama u sustavu.

Trenutna sučelja, međutim, morat će se mijenjati, planira li se sustav ekstenzivno koristiti, ili je u planu izrada novih igara, a prvenstveno zato što u nekim aspektima na igru postavlja nelogične zahtjeve. Primjerice, MGMTM očekuje da se klasa koja implementira sučelje IGameProvider sama brine o tome da se njene kontrole prikazuju u istoj dretvi (engl. *thread*) kao i kontrole MGMTM-a (u protivnom sustav blokira), jer nije definirano hoće li se pozivi metoda i svojstava uvijek pozivati u istoj dretvi (poznato je samo da se ne pozivaju uvijek iz dretve u kojoj su pozvani konstruktori kontrola prikazanih na sučelju). Rješenje tog problema je prilično nespretno jer zahtijeva prikaz nevidljive korisničke kontrole u nevidljivom prozoru pri pozivu konstruktora (ukoliko se prikaže korisnik ga može zatvoriti, pa je funkcija tog prozora narušena) te naknadnim pozivom metoda za prikaz dodatnih prozora metodama implementiranim unutar tog nevidljivog prozora, odnosno kontrole. Istovremeno, sam GameProvider mora pamtit referencu na kontrole koje je stvorio, ne bi li ih ispravno zatvorio, uništio i zaustavio dretve koje se eventualno izvode kao rezultat prikaza tih kontrola.

Nadalje, osim putem zapisa u *log* (koji korisnik možda hoće, a vjerojatno neće pogledati, a sam sustav se ne brine o sadržaju zapisa), igra nema načina da sustavu dojava da se u njoj dogodila neka greška (pa se, na primjer, trenutna igra neće završiti). Igra se može prisilno završiti (bez obzira što se ona možda još uspješno izvršava, ukoliko se greška dogodila u nekom drugom dijelu) i označiti kao prekinuta. Takvu poruku, međutim, MGMTM tumači kao "prekinuta od strane korisnika", što nije nužno istina, bez obzira na to što se upiše u opis greške.

Dakle, bez obzira na izvrsnost izvedbe ovog okruženja, u nekim od osnovnih stvari još uvijek se javljaju problemi koje je potrebno ispraviti.

Pobliže su sva sučelja i što se od pojedinih svojstava i metoda očekuje, opisana u [2] te nema potrebe ovdje ulaziti u detalje.

### 3.2.1. Sučelja prema igri

Da bi se igra mogla učitati u sustav MGMTM, potrebno je izraditi komponente koje implementiraju određena sučelja. Za olakšavanje toga posla, ponuđena je klasa `AGeneric<TSettings>` koja na dosta smislen način već implementira neke od traženih metoda, a koje se mogu preopteretiti (engl. *overload*) u deriviranoj klasi ukoliko se za time pokaže potreba.

Također je dana i klasa `GenericUserControl` koja se u igri prvenstveno koristi za prikaz kontrole na kojoj korisnik odabire postavke igre. Kontrola koja implementira tu funkcionalnost mora nasljeđivati klasu `GenericUserControl`.

Postojeća sučelja prema igri su:

- `IGameProvider` – ovo je takoreći "srce" veze okruženja MGMTM i igre. Klasa koja implementira to sučelje odgovorna je za dojavljivanje događaja u igri sustavu (kraj igre, igra pregledana, prijava igrača te dojava o napretku igre ukoliko se ona izvršava), otvaranje i korektno zatvaranje elemenata sučelja, pokretanje i pregled igre te dojavu sustavu o tome je li klasa spremna za novu igru.
- `IGameDescription` – komponenta koja služi za opis igre, a sadrži razne detalje poput naziva igre, imena autora, datuma nastanka, verzije i slično.
- `IGame` – komponenta koja daje podatke o konkretnoj igri. Radi se o informacijama o stanju igre te eventualnom rezultatu i igračima. Izvedbe opisane u ovom radu implementiraju to sučelje u istoj klasi koja implementira i sam mehanizam igre.
- `ISettings` – kao što joj ime govori, od klase koja implementira ovo sučelje tražit će se da bude sposobna predati sustavu svoje postavke. Predaju se isključivo standardne postavke, a one se mijenjaju putem kontrole koja mora nasljeđivati klasu `GenericUserControl`.

### 3.2.2. Sučelje prema igračima

Postoji samo jedno sučelje prema igračima, i to sučelje `IPlayer`. Svaki od implementiranih igrača implementira to sučelje (bez posredničke klase). Jedna od važnih funkcionalnosti koju traži ovo sučelje jest da igrač nedvosmisleno identificira lokaciju svoje DLL-datoteke na disku. Pri učitavanju igre ta se informacija koristi da bi se ispravno učitali i igrači koji su u njoj sudjelovali. Igre koje omogućavaju mrežno igranje morat će se pobrinuti da informiraju sustav (i korisnika) o tome da igrač možda nije dostupan. Jedno rješenje za taj problem (implementirano pri spajanju igre dame sa sustavom MGMTM) jest da se

umjesto stvarnih igrača učitaju nadomjesni igrači koji nemaju implementiranu igraču logiku, ali su sposobni vratiti sve ostale podatke o igraču koje zahtijeva sustav MGMTM.

Od ostalih funkcionalnosti, komponenta koja implementira sučelje IPlayer također mora ispravno opisati igrača. Sustav će od igrača tražiti slične podatke o igraču kao i kod igre (naziv, autore, itd.), uz još neke podatke koje se tiču isključivo igrača.

INTERNI DOKUMENT

## 4. IGRE I IGRAČI

U okviru ovog rada razvijene su dvije društvene igre i to dame i halma. Razlozi za to prvenstveno se tiču povijesti razvoja cjelokupnog okruženja MGMTM. Dok su dame postojale od ranije i postupno su prilagođavane okruženju MGMTM te time služile kao svojevrsna testna igra za cijelo okruženje, zanimalo nas je je li moguće izraditi općenito sučelje za organizaciju turnira u bilo kojoj igri za dva ili više igrača.

MGMTM je pripremljen s time na umu, no nije u potpunosti prilagođen igrama za više od dva igrača, što se očituje prvenstveno na sučelju prema čovjeku, ali i u organizaciji turnira. Primjerice, organizira li se turnir u halmi za šest igrača, MGMTM će u grupi sa šest igrača generirati šest igara na način da igrači jednostavno rotiraju mjesta (svaki puta igraju drugom bojom). Međutim, to se baš u slučaju halme pokazalo potpuno nepotrebnim, obzirom da prema pravilima igru započinje nasumično izabrani igrač.

Prije nego se krene na opise pojedinih igara valja spomenuti da nije bilo ozbiljnijih pokušaja optimizacije koda – ni u simulatorima, ali ni u igračima (osim standardnog podrezivanja alfa-beta i manjih modifikacija vezanih uz njega). Obzirom na podužu povijest razvoja – koliko kompletnog okruženja MGMTM, toliko i samih igara i igrača – to može značiti da se ponegdje mogu pronaći neiskorištene (ili zastarjele) metode i svojstva, ili linije koda koje ne pridonose rezultatu izvršavanja tog koda, kao i neka rješenja koja se u kontekstu aplikacije kakva sada jest čine neobičnima jer je naknadno razvijena nova funkcionalnost koja bi mogla ubrzati ili olakšati izvedbu rečenog rješenja.

To se prvenstveno odnosi na igru dame, obzirom da je halma razvijana podosta vremena nakon što su sva sučelja prema okruženju MGMTM već bila definirana i nisu se naknadno mijenjala.

### 4.1. Igra dame

Igra dame, kao i MGMTM ima podosta dugu povijest razvoja. Njeno sučelje, kako se vidi pokretanjem klijenta, prikazuje Slika 2. Prvotno, izvedba je bila zamišljena kao gotova aplikacija koja u jednoj dretvi simulira cjelokupni turnir, a igračima za poteze daje procesorsko vrijeme u drugoj dretvi te im šalje prekide kod isteka vremena za potez. S tom izvedbom, gledanje partija u stvarnom vremenu kako igrači zaista vuku poteze bilo je nemoguće te se cijeli turnir morao simulirati prije nego što je zaista prikazan. To je dovelo do izvedbe sustava za sakrivanje rezultata igre prije nego što je igra prikazana na zaslonu.

Dodatno je osigurano da se tokom vremena predviđenog za igračev proračun poteza u aplikaciji ne događa baš ništa, ne bi li mu se ostavilo što više procesorskog vremena.

Takva izvedba, međutim, nije bila povoljna za ostvarivanje željenog cilja, a to je korištenje okruženja u edukacijske svrhe. Naime, od igrača se zahtjevalo da u danom vremenskom roku vrata potez koji smatraju najboljim za poziciju na ploči koja im se slala kao argument pri pozivu funkcije za proračun poteza.

Edukacijski cilj bio je potaknuti studente da osmisle što bolje rješenje za igrača i što bolju heuristiku, a da bi se na kraju rješenja svela na jednostavnu heuristiku i bila optimirana za što bržu pretragu stabla.



Slika 2. Korisničko sučelje igre dame.

Poučeni time, relaksirali smo pravilo o "nesmetanju" igrača, na način da smo odlučili "smetati" svakom igraču jednako generirajući im sve moguće legalne poteze za svaku poziciju. Efektivno su budući programeri time bili prisiljeni koristiti već izvedeni generator poteza, jer izvesti svoj generator, koliko god optimiran za brzinu, je dodatni trošak procesorskog vremena koji si u većini slučajeva ne mogu priuštiti.

Time se optimizacija uvelike preselila na algoritam pretrage stabla (ukoliko je isti postojao) i više je pozornosti posvećeno izradi dobre heuristike. Obzirom da je u ovom slučaju cilj edukacije bio upravo algoritam minimax i podrezivanje alfa-beta, time je naš cilj postignut.

Osim samog generatora poteza, igrača će još "ometati" otkucavanje vremena na sučelju te eventualno korisnikovo pomicanje prozora po ekranu. Iscrtavanje figura pri izvedbi poteza neće smetati igrača, obzirom da se to obavlja između dva poteza.

Pokazalo se da, osim što nam je pomogao pri ostvarenju edukacijskih ciljeva, generator poteza ugrađen u igru umjesto igrača omogućava općenitiju izradu igrača koji ne mora brinuti o pravilima igre. Obzirom da pravila uvelike govore o tome koje je poteze moguće napraviti na danoj poziciji, briga o pravilima preseljena je u generator poteza, a pretragu prostora stanja moguće je tada raditi bez ikakvog znanja o pravilima. Jasno je, određene značajke igre doći će do izražaja u nekom dijelu igračevog koda, npr. u heuristici, no vjerojatnost da neki igrač bude neispravan jer vraća nelegalne poteze uvelike se smanjila.

#### 4.1.1. Pravila igre

Pravila igre koja su poštivana su:

- Igru igraju dva igrača, i to igrač svjetle boje koji počinje igru prvi te igrač tamne boje;

- Svaki igrač na početku ima dvanaest figura (pijuna), raspoređena u donja za svijetlog, odnosno gornja tri reda šahovske ploče za tamnog, i to isključivo na bijela polja;
- Kada je na redu, igrač vuče jednu svoju figuru za jedno mjesto dijagonalno, s time da se pijun može kretati samo prema suprotnoj strani ploče od strane na kojoj je počeo, a dama se može kretati i prema natrag (također jedno mjesto dijagonalno);
- Kada pijun dođe u zadnji red na suprotnoj strani ploče, on biva promoviran u damu. Promocija u damu označava kraj poteza za tog igrača (tj. nije moguće nastaviti činiti ulančani skok u istom potezu nakon promocije);
- Ako se na polju na koje se figura inače može pomaknuti nalazi protivnička figura, a polje iza te figure je prazno, igrač može svojom figurom preskočiti protivnikovu figuru i maknuti je iz igre te staviti svoju figuru na prazno polje iza (takvi preskoci su također dijagonalni u jednom smjeru). Ako je nadalje na polju na kojem se nakon skoka figura nalazi moguć još jedan ili više takvih skokova, igrač ih mora nastaviti raditi do onog trenutka kada više nema mogućih skokova. To je ulančani skok. Pojedini skokovi za pijuna uvijek su u smjeru dijagonalno prema naprijed, a za damu su u bilo kojem dijagonalnom smjeru. Svaki pojedini skok kod ulančanog skoka može biti u bilo kojem – za tu figuru – legalnom smjeru;
- U jednom skoku moguće je preskočiti i uzeti samo jednu protivničku figuru. Više od jedne protivničke figure moguće je uzeti samo kod ulančanog skoka;
- Ako je na danoj poziciji moguće uzeti jednu ili više protivničkih figura, tada igrač mora igrati potez kojim uzima protivničku figuru. Ako igrač ima više poteza kojima može uzeti protivničke figure, može igrati samo jedan od tih poteza;
- Igrač je dobio igru ukoliko je njegov suparnik ostao bez figura, ili nema legalan potez;
- Igra završava izjednačenjem ukoliko se ista pozicija ponovi tri puta na ploči, ili je dosegnut maksimalan broj poteza ranije postavljen u postavkama igre.

#### 4.1.2. Igrač Randomizer

*Randomizer* je naziv najstarijeg od svih igrača dama razvijenih pod okruženjem MGMTM. Iz skupa legalnih poteza on nasumično bira jedan i vraća ga u igru. Osim što je služio za testiranje sustava, pokazao se svrsishodnim i kod testiranja ozbiljnijih igrača dama. Svaki takav (ozbiljniji) igrač trebao bi biti u stanju pobijediti igrača Randomizera u gotovo svakoj partiji (vjerojatnost da igrač Randomizer odabere optimalne poteze na svakoj poziciji u jednoj partiji je mala, ali postoji).

Osim igrača Randomizera postoji još jedan igrač koji vraća nasumične poteze, no on uz legalne poteze može vratiti i nelegalne, ili pustiti da vrijeme istekne. Vjerojatnost da se bilo koja od tih grešaka dogodi je 0.5%, a svrha tog igrača – nazvanog *Mr. Error* – bila je testiranje otpornosti sustava na greške u igračima.

### 4.1.3. Igrač Lockdown

Igrač Lockdown je dobio ime po heuristici koja je prvotno bila osmišljena za njega – a kasnije se ispostavila lošom – koja je trebala tom igraču omogućiti da blokira protivnikove poteze i time "zaključava" protivničke figure na mjestu (tako da se ne mogu pomaknuti). Problem koji se javio bio je taj što je takvim zaključavanjem, igrač Lockdown "zaključao" i vlastite figure. Nadalje, dubina do koje je algoritam minimax (koji ovaj igrač koristi) u mogućnosti stići u danom vremenu nije dovoljno velika da bi igrač "vidio" da neki potezi "zaključavanja" baš i nisu u konačnici potezi kojima se protivniku onemogućava potez, već da su ti potezi zapravo potezi koji će rezultirati gubitkom jedne ili više figura. K tome je vrlo lako moguće u igri dame da igrač s manje figura ima više legalnih poteza od igrača s više figura. Ono što je pokazalo pravu slabost takve heuristike je činjenica da prema pravilima igre, ako igrač može napraviti skok (uzimanje protivničke figure), on ga mora napraviti. To znači da ukoliko igrač "ponudi" figuru protivniku, protivnik (u većini slučajeva) ima samo jedan mogući potez, a ovakva heuristika takvoj poziciji daje veliku ocjenu.

Očito je bilo da je u takvu heuristiku bilo potrebno dodati i vrednovanje figura. Dodano je vrednovanje razlike broja vlastitih i protivničkih figura, i to tako da je bez obzira na broj mogućih poteza, razlika od jedne figure uvijek prevagnula nad razlikom u broju poteza. Jedan potez tada se ocjenjivao ocjenom 1, a faktor koji množi razlika broja figura postavljen je na 20. Također je uvedeno vrednovanje razlike u broju kraljeva, s istim faktorom.

Nakon više odigranih partija, primjećeno je da u određenim završnicama u kojima igrač Lockdown ima ogromnu nadmoć nad svojim suparnikom, igra završava izjednačeno ponavljanjem pozicija. Da bi se riješio taj problem, bila je razmatrana mogućnost uvođenja nove heuristike koja bi se koristila u takvim situacijama. Ista je implementirana, no ubrzo se pokazalo da je uvjet u kojem se ta heuristika aktivira bio loše postavljen. Naime, vrlo je teško odrediti trenutak u kojem igra ulazi u svoju završnicu i uvjet koji bi trebalo postaviti je previše kompleksan i ovisi o mnogo faktora (npr. broj dama na ploči, broj figura na ploči, trenutni potez, razlika u ukupnom broju figura i dama te pijuna i dama, i sve to za svakog igrača, tek su neki od faktora). Iz tog razloga na kraju je odlučeno da se heuristika za rješavanje završnice ubaci u već postojeću heuristiku kao dio nje, i to tako da ne uništava već učinjenu evaluaciju figura i poteza.

Najprije su svi otprije odabrani faktori bili množeni s 5 (što je bilo pretpostavljeno kao dovoljno za ubacivanje novih evaluacija). Novi dio heuristike broji figure po četvrtinama ploče. Nakon toga sortiraju se četvrtine prema broju bijelih, odnosno crnih figura. Gleda se na kojoj je četvrtini ploče najveća koncentracija protivničkih figura. Ako se vlastite figure ne nalaze na toj četvrtini, onda se za svaku takvu figuru igrač kažnjava s određenim negativnim brojem, čija je veličina proporcionalna udaljenosti od te četvrtine. Ideja te heuristike je natjerati igrača da se primakne protivniku, a zatim pustiti da algoritam minimax pronađe put do pobjede. Micanjem prema protivniku efektivno se smanjuje dubina do koje algoritam minimax treba pretraživati stablo, ne bi li pronašao pobjedu.

Eventualna mana ovakvog pristupa je u tome što je moguće da završnica igre bude nepovoljna za igrača Lockdown, odnosno igrač Lockdown ima manje figura od protivnika. Obzirom da trenutno nema dovoljno snažnih igrača dama razvijenih u našem okruženju koji bi mogli dovesti do takve situacije, izrađeno je nekoliko testnih pozicija koje demonstriraju to ponašanje (u igri protiv samoga sebe), a raspleti se mogu učitati putem sučelja za testiranje, kako je to opisano u dodatku.

Osim toga, postavlja se pitanje što se događa ukoliko je protivnik jednoliko raspršen po više četvrtina ploče. To ovisi samo o tome koju će četvrtinu funkcija za sortiranje staviti prvu, jer igrač Lockdown će se u tom slučaju primicati upravo toj četvrtini. Naravno, obzirom da je taj dio heuristike koji prisiljava igrača Lockdown da se kreće prema protivniku tako ubačen u već postojeću heuristiku da se ne poništava utjecaj evaluacije razlike u broju figura, kraljeva i poteza, igrač će se primicati oprezno, pazivši da pritom ne izgubi neku od figura bez razloga.

#### 4.1.4. Igrač Junior

Ovaj igrač nema standardno izvedenu heuristiku, već heurističku procjenu pozicije daje neuronska mreža. Mreža na svoj ulaz prima iste značajke pozicije kakve se ocjenjuju i kod igrača Lockdown (dakle, razlika u broju figura, kraljeva, poteza te razlike u broju figura prema četvrtinama ploče. To čini 8 ulaza. U dva skrivena sloja mreža ima po 12 neurona, a na izlazu se nalazi jedan. Mreža je potpuno povezana.

Za ovu konkretnu mrežu uzeta su dva skrivena sloja prema modelu opisanom u [1]. Obzirom da se neuronska mreža nije trenirala na fiksnom skupu partija, već su prikladne težine "pretraživane" genetičkim algoritmom (čime se teoretski dobio neograničen broj partija), prenaučenost se nije smatrala problemom. Zbog toga je u skrivene slojeve stavljen veći broj neurona nego je zaista potrebno.

Genetički algoritam bio je postavljen na 200 generacija, iako je konačna mreža pronađena mnogo prije. Svaka generacija imala je po 10 jedinki, a u novu generaciju prelazilo je 5 nepromijenjenih jedinki. Križanje između dvije jedinke radilo se računanjem prosjeka između ekvivalentnih parova težina. Mutacija se događala samo nad novom jedinkom, uz vjerojatnost od 10% da se jedinka uopće mutira te vjerojatnost od 2% da se mutira pojedini gen jedinke (svaka jedinka imala je po  $8 * 12 + 12 * 12 + 12 = 252$  gena, odnosno težina). Mutacija gena (težine) radila se tako da se težini pridjelila nova, nasumična vrijednost u rasponu  $[-1,1]$ .

Inicijalno je zamišljeno da se generira više neuronskih mreža s različitim brojem skrivenih slojeva i neurona u svakom sloju, te se za igrača Junior iskoristi najbolja. Genetički algoritam je, na žalost, izvršen samo za ovu mrežu jer su se sve partije simulirale putem sučelja izvedenog u sklopu klijenta za igru dame. Obzirom da to sučelje nije optimirano za izvođenje velikog broja partija (već u samoj svojoj izvedbi uvodi vremenske stanke, najprije za prikaz partije, a zatim fiksnu stanku između poteza od 100 ms), treniranje je teklo iznimno sporo. Na samo 9000 partija odigranih u konačnoj (uspješnoj) pretrazi utrošena su puna

četiri dana. Zbog toga, a i zbog relativno dobrog rezultata dobivenog ovim treniranjem, nije se izvodio genetički algoritam za ostale mreže.

U svakoj generaciji igralo se po 45 partija dama. Prvi igrač igrao je kao bijeli protiv svih ostalih (9 partija), zatim je drugi igrač igrao protiv svih osim prvog (8 partija), itd. Razlog zbog kojeg igrači nisu igrali i kao bijeli i kao crni bio je smanjenje broja partija. Za svaku pobjedu, igrač je dobivao 3 boda, za izjednačenu partiju 1 bod, a za gubitak 0 bodova. Igrač s najvećim brojem bodova nakon što je odigrano svih 45 partija automatski je prelazio u novi krug (elitizam). Preostalih četiri igrača koji su nepromijenjeni prelazili u novi krug birali su se po principu ruleta. Prema osvojenom broju bodova pridjeljivala im se određena vjerojatnost da budu odabrani – i to što je veći broj bodova to je veća i vjerojatnost – te su se nasumično birali prema toj vjerojatnosti. Kada je odabran jedan od njih, on je prebačen u novu generaciju, a ostali su ulazili u slijedeći krug ruleta.

Novе јединке izvodile su se isključivo od najboljih јединки u upravo ocijenjenoj generaciji, i to križanjem najbolje i druge najbolje јединке, zatim najbolje i treće najbolje, pa druge i treće najbolje, pa najbolje i četvrte najbolje i napokon druge najbolje i četvrte najbolje јединке.

Konačna mreža pronađena je neobično brzo, i to već u jedanaestoj generaciji, no uvjet zaustavljanja bio je postavljen na izvedenih 200 generacija (9000 igara).

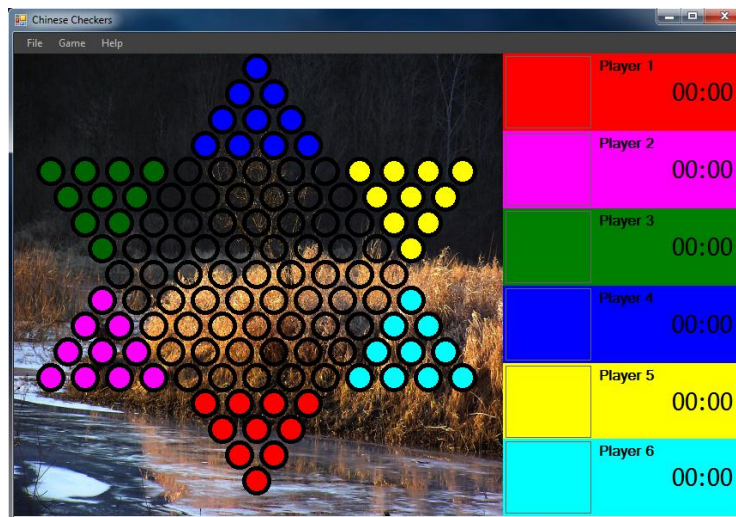
Osim što poziciju ocijenjuje neuronska mreža, prije njene evaluacije pozicija se analizira. Ukoliko je pobjednička za bilo kojeg igrača, funkcija vraća vrijednost +1 ili -1, ovisno o igraču koji je pobijedio (+1 ako je pozicija pobjednička za tog igrača, a -1 ako je pobjednička za protivnika)

## 4.2. Halma

Igra halma izrađena je kao prva igra za više od dva igrača u okružju MGMTM. Slika 3 prikazuje njeno korisničko sučelje. Iako je programirana tako da bude dostupna i kao samostalna aplikacija i kao jedna od igara u okružju MGMTM, neće se moći pokrenuti na sustavu bez okružja MGMTM, zbog problema navedenih ranije.

Za razliku od dama, halma nema dugu povijest razvoja u kojoj bi sudjelovalo više od jednog programera, pa stoga ima konzistentnije projektirane klase koje su, ukoliko se njihov dizajn razumije, mnogo lakše nadogradive nego je to slučaj u dama.

Osim u dizajnu, izvedba donosi neke novosti i za igrače. Za razliku od dama, gdje se svi potezi generiraju za svaku poziciju koja se konstruira (pa tako i za pozicije koje su listovi u stablu generiranom pretragom), ovdje se potezi generiraju isključivo na zahtjev igrača, i to samo na odabranom polju. Funkcija za generiranje svih poteza ne postoji, nego je igrači moraju implementirati ukoliko je to potrebno. To igračima daje opciju da koriste svoj generator poteza, ili ugrađeni.



Slika 3. Korisničko sučelje igre halma.

#### 4.2.1. Pravila igre

Pravila koja se poštuju su sljedeća:

- Igru može igrati dvoje, troje, četvero ili šestero igrača. Ako se igra u dvoje, figure su smještene u donjem trokutu (crveni igrač) i gornjem (plavi igrač). Ako se igra u četvero, figure su smještene u donji i gornji trokut te još u bilo koja dva nasuprotna trokuta na ploči. Ako se igra u troje, figure su smještene u donji, gornji lijevi i gornji desni trokut. Ako se igra u šestero, svi trokuti su popunjeni;
- Igrač koji počinje igru bira se nasumično. Dalje igrači dolaze na red kako su razmješteni u smjeru kazaljke na satu;
- Igrači u jednom potezu pomiču jednu figuru za jedno polje u bilo kojem smjeru, ili preskaču drugu figuru (uključujući svoju). Skokovi se mogu ulančavati, tj. moguće je odigrati više skokova u istom potezu. Jedan skok može biti u bilo kojem, ali jednom smjeru te se može preskočiti samo jedna figura. Ulančani skok je niz skokova te ih je moguće ulančavati u različitim smjerovima;
- Igrač ne mora odigrati najdulji mogući lanac skokova i može stati s ulančavanjem u bilo kojem trenutku, bez obzira na to može li napraviti još ulančanih skokova;
- Figure se nikada ne miču s ploče, a smiju se pomaknuti na bilo koje polje na ploči;
- Ciljni trokut je trokut na suprotnoj strani ploče od trokuta u kojem se početno nalaze igračeve figure;
- Cilj igre je smjestiti svih 10 figura u ciljni trokut;
- Ukoliko se u ciljnom trokutu nalazi protivnička figura i sva polja ciljnog trokuta su popunjena, igrač smije napraviti zamjenu, tj. staviti svoju figuru na zauzeto mjesto, a protivnikovu figuru maknuti na mjesto s kojeg igrač pomiče svoju figuru. Protivnik koji se nalazi u tuđem ciljnom trokutu ne može raditi zamjenu figura u tom trokutu.

#### 4.2.2. Igrač Randomizer Jr.

Kao i kod igre dame i igrača Randomizer, ovdje je igrač Randomizer Jr. služio za testiranje aplikacije. Međutim, obzirom da su pravila igre ovdje manje rigorozna nego u igri dame i moguće je figuru pomaknuti u bilo kojem od 6 smjerova (ukoliko su polja slobodna), nije se pokazao kao veoma dobar igrač za testiranje. Igrač Randomizer Jr. nije u stanju doći do ciljnog trokuta sa svim svojim figurama u doglednom vremenu, već besciljno luta po ploči. Zato je, umjesto njega, korišteno sučelje prema čovjeku (tzv. ljudski igrač) i ispravno okidanje događaja o kraju igre, pobjedi i slično, testirano je na taj način.

#### 4.2.3. Igrač Dumbo

Dumbo je dobio svoje ime po tome što je razmjerno neinteligentan. U osnovi, sposoban je kretati se prema cilju i završiti igru u dogledno vrijeme, no faktor grananja prostora stanja je ogroman i s obzirom da Dumbo nema gotovo nikakvu optimizaciju izvedenu nad svojim algoritmom, on pretražuje gotovo sve moguće poteze (u principu ne pretražuje poteze prema natrag) i sve moguće pozicije koje se mogu doseći iz danog stanja na ploči.

Termin "u principu" ovdje znači da igrač Dumbo, iako izbacuje poteze prema natrag iz liste za pretraživanje, on može prilagoditi igru i napraviti takav potez ukoliko nijedan drugi potez nije moguć (što je zapravo iznimno rijetka situacija). Pojedinačni skokovi u ulančanom skoku ipak mogu biti prema natrag, jer je jedini uvjet nad svim potezima da završavaju na lokaciji koja je bliža cilju (prema heuristici) nego ona s koje počinju.

Dumbo ocjenjuje pozicije prema položaju svojih i protivničkih figura na ploči. Svakom polju na ploči, za svakog igrača, pridijeljena je određena vrijednost, i to tako da su pozitivnije vrijednosti u samom ciljnom trokutu, a negativnije na početnom trokutu. Ukoliko se igračeva figura nalazi na određenom polju, vrijednost tog polja za igračevu boju pribraja se ukupnoj ocjeni pozicije. Ukoliko se protivnikova figura nalazi na određenom polju, vrijednost tog polja za protivnikovu boju oduzima se od ukupne ocjene pozicije.

Za crvenog igrača, vrijednosti polja prikazuje Slika 4. Za ostale igrače vrijednosti polja su ekvivalentne.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0					4												
1				3	3												
2			2	2	2												
3			1	1	1	1											
4	-10	-8	-6	-4	-2	-2	-2	-2	-2	-4	-6	-8	-10				
5	-10	-8	-6	-4	-4	-4	-4	-4	-4	-6	-8	-10					
6	-10	-8	-6	-6	-6	-6	-6	-6	-6	-8	-10						
7	-10	-8	-8	-8	-8	-8	-8	-8	-8	-8	-10						
8	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10	-10						
9	-12	-12	-12	-12	-12	-12	-12	-12	-12	-12	-12						
10	-14	-14	-14	-14	-14	-14	-14	-14	-14	-14	-14						
11	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16	-16						
12	-18	-18	-18	-18	-18	-18	-18	-18	-18	-18	-18						
13			-20	-20	-20	-20											
14				-22	-22	-22											
15					-24	-24											
16						-26											

Slika 4. Vrijednosti polja ploče za igru halme koja se koriste pri proračunu heurističke vrijednosti pozicije.

Na toj slici, prvi redak i prvi stupac označavaju koordinate polja kako se one koriste u samoj igri, a sjenčanje polja služi lakšoj orijentaciji.

Ovakvo ocjenjivanje pokazalo se pomalo nespretno u završnici igre, pogotovo ako je neka od protivničkih figura ostala u igračevom ciljnom trokutu. Dumbo tada popuni ciljni trokut, i kada posljednje figure stignu do same granice s ciljnim trokutom, on protivničke figure ne izbacuje van iz trokuta, nego ih ugura u sam vrh trokuta. To je slučaj ukoliko je vrijeme za potez dosta kratko (isprobano na vremenima od jedne do 5 sekundi po potezu za 6 igrača), i Dumbo ne stigne pronaći pobjedu koja je toliko blizu (a toliko daleko). Takva igra na kraju završava izjednačenjem zbog prekoračenja ograničenja na broj poteza (nema provjere, kao u igri dame, je li se ista pozicija ponovila tri puta).

Drugi slučaj u kojem se ovo ocjenjivanje pokazalo nepraktičnim je kada je protivnik maknuo svoje figure iz igračevog ciljnog trokuta, a igrač je doveo gotovo sve (osim npr. jedne) figure u ciljni trokut, a jedna, dvije ili više figura su na poljima na unutarnjim kutevima ciljnog trokuta (kao što prikazuje Slika 4, za crvenog igrača to su polja (4, 4) i (8, 4), ali i sva polja dijagonalno lijevo-dolje, odnosno dijagonalno desno-dolje). Uz pretpostavku da su jedno ili oba polja, od (5, 3) i (6, 3) prazna, Dumbo neće svojim figurama izvan ciljnog trokuta činiti horizontalni pomak, jer je njemu dobitak na heuristici za sljedeću poziciju za taj potez nula, tj. taj potez ekvivalentan mu je bilo kojem horizontalnom pomaku unutar ciljnog trokuta.

Ova igra će možda završiti, a sve ovisi o protivničkim igračima, tj. o tome hoće li i oni doći u sličnu situaciju ili neće. Igre s deset i više sekundi po potezu u principu su završavale na računalo koje je korišteno za testiranje.

## 5. VREDNOVANJE IGRAČA

Da bi se provjerilo koliko su izrađeni igrači zaista dobri, za svakog igrača proveden je neki oblik njegovog vrednovanja. Dodatno je za igrača Lockdown učinjena ablacijska studija u kojoj je izrađeno nekoliko verzija igrača, pa su zatim originalan Lockdown i sve njegove varijacije stavljene u turnir protiv drugih igrača. Za igrača Junior također je organiziran zasebni turnir.

Svi turniri i sve igre igrane su na istom računalu, s procesorom Intel(R) Core(TM) i7 930 na 2.80GHz, sa 6.00 GB RAM, na 64-bitnom sustavu Windows 7 Ultimate. Rezultati mogu biti bitno drukčiji na različitim računalima, ovisno o brzini procesora, količini memorije te ostalim aplikacijama koje se izvršavaju u pozadini i njihovom načinu korištenja procesora.

Opis turnira i ostalih igrača slijedi.

### 5.1. Turnir i protivnici

Turniri su organizirani u igri dame. Budući da trenutno nema dovoljno igrača za turnir u igri halme, igrač Dumbo testirao se na drukčiji način. Igrači koji vuku nasumične poteze nisu se testirali i nisu sudjelovali u turnirima.

U svakom turniru sudjelovao je igrač koji se testira i po sedam drugih igrača. Te igrače izradili su studenti koji su prethodnih godina u sklopu predmeta Umjetna Inteligencija sudjelovali u tzv. izazovu, i to isključivo studenti koji su u izazovu sudjelovali 2010. godine. U turnir nisu ulazili igrači koji su se pokazali lošima, odnosno oni kojima je vrijeme predviđeno za potez isticalo. Također, u turnir nije ušao jedan igrač koji je konzistentno utrošio cijelo predviđeno vrijeme, a onda povukao očito loš potez. Ti igrači mogli su biti uključeni u turnir, no to nije učinjeno u interesu skraćivanja vremena potrebnog za simulaciju turnira.

Svi igrači implementiraju algoritam minimax i optimizaciju podrezivanjem alfa-beta, a razlikuju se jedino prema svojoj heurističkoj funkciji, te prema načinu na koji raspoređuju vrijeme koje im je dodijeljeno. Tablica 1 daje kratak opis tih igrača.

Naziv igrača	Heuristika
Chuckers	Koristi se više različitih, ovisno o stanju ploče
TETKICA	Pojedine karakteristike ocjenjuju se različito, ovisno o fazi igre koja se određuje prema broju dama na ploči
Darko	Jedinstvena kroz sve faze igre, sa mogućnošću odabira nasumičnog poteza
TheMarvin	Jedinstvena, ocjenjuje se velik broj značajki
Vrijeme je za dddvboj	Jedinstvena, ocjenjuju se samo bijele figure
Nameero	Razdijeljena u tri faze, ocjenjuje se velik broj značajki
VOV	Jedinstvena, pijuni se ocjenjuju ovisno o udaljenosti od suprotne strane, a dame ovisno o udaljenosti od figure protivničkog igrača

Tablica 1. Kratak opis igrača koji su se koristili u turnirima dame.

Postavke turnira uvijek su bile iste, odnosno igračima se davala puna minuta po potezu, dok je ograničenje na broj poteza bilo 150 po igraču. Igrači su svrstani u jednu grupu na jednom nivou turnira te je svaki igrač igrao sa svakim i kao crni i kao bijeli, dakle odigrano je po 56 partija u jednom turniru, odnosno po 14 partija po igraču. Ishodi su se ocjenjivali standardno, odnosno 3 boda za pobjedu, jedan bod za izjednačenje te 0 bodova za gubitak partije.

Primjećeno je da igrači koje su izradili studenti uzimaju otprilike konstantno vrijeme po potezu, bez obzira na stvarno raspoloživo vrijeme. To je razumljivo, obzirom da tadašnja verzija igre dame nije davala potpunu informaciju o raspoloživom vremenu. Igrač Lockdown je, za razliku od njih, koristio puno vrijeme po potezu. No htio bih naglasiti da minuta po potezu nije igračima dana da bi se igračima Lockdown i Junior dala prednost, već stoga što je cilj studije bio testirati heuristiku Lockdowna, odnosno neuronsku mrežu Juniora. Kod igrača Lockdown dodatno su time naglašene razlike između originalnog igrača i njegovih verzija korištenih za ablacijsku studiju, i stoga je bilo lakše analizirati razlike u heuristikama.

Na kraju je između svih devet igrača – Lockdown i Junior te sedam igrača studenata – organiziran još jedan turnir u stvarnim uvjetima, odnosno s postavkama vremena kakve su korištene na jednom stvarnom turniru (2 minute i 30 sekundi za svakog igrača i sve njegove poteze)<sup>1</sup>.

## 5.2. Igrač Lockdown

### 5.2.1. Ablacijska studija

Za potrebe ablacijske studije izrađene su četiri dodatne verzije igrača Lockdown. U svakoj je eliminiran po jedan od faktora koji se računavaju u konačnu heurističku procjenu pozicije. Podsjetimo se, faktori koji određuju ocjenu heuristike su:

- Razlika u broju figura igrača i protivnika;
- Razlika u broju kraljeva igrača i protivnika;
- Razlika u broju poteza igrača i protivnika, s računatim brojem poteza kojima se uzimaju figure;
- Četvrtina (kvadrant) ploče na kojoj se nalazi najveći broj protivničkih figura.

Naziv igrača	Heuristika
Lockdown	Kompletna, izvorni igrač
Lockdown-Pieces	Bez evaluacije pijuna
Lockdown-Kings	Bez evaluacije dama
Lockdown-Taking	Bez evaluacije poteza
Lockdown-Travel	Bez heuristike za micanje prema protivniku

Tablica 2. Prikaz svih varijanti igrača Lockdown izvedenih za ablacijsku studiju.

<sup>1</sup> Rezultati tog turnira mogu se vidjeti učitavanjem turnira putem sučelja MGMTM. Sve potrebne datoteke za to nalaze se na DVD-u uz

Time su se dobila četiri dodatna igrača koje kratko opisuje Tablica 2. Oni različito ocjenjuju pozicije te se kroz igru protiv raznih drugih heuristika može analizirati utjecaj svakog od tih faktora te eventualno pronaći neki od faktora koji nema nikakav utjecaj, pa se može izbaciti iz konačne heuristike.

Prema rezultatima studije, pokazalo se da je svaki od faktora bitan, a da bi saznali zašto, potrebno je napraviti usporednu analizu odigranih turnira svake verzije igrača, s rezultatom turnira s izvornim igračem. U analizi se neće činiti analiza pojedinih partija osim u onoj mjeri koliko je to potrebno da bi se razumjeli utjecaji pojedinih čimbenika heuristike izvornog igrača. U protivnom za to bi bio potreban stručnjak u igranju igre, pa će se, gdje je moguće, se samo gledati razlike u dobivenim, izjednačenim i izgubljenim partijama. Također će se pokušati odrediti zbog čega je određena verzija igrača igrala bolje ili lošije u odnosu na izvornog igrača, uzimajući u obzir dio heuristike koji je eliminiran.

### Rezultati s izvornim igračem

Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	Lockdown	14	12	2	0	38
2	TheMarvin	14	9	3	2	30
3	TETKICA	14	9	3	2	30
4	VOV	14	7	3	4	24
5	Chuckers	14	5	4	5	19
6	Vrijeme je za dddvoboj	14	4	1	9	13
7	Nameero	14	2	0	12	6
8	Darko	14	0	0	14	0

Tablica 3. Rezultati turnira s izvornim igračem Lockdown

Kao što prikazuje Tablica 3, izvorni igrač je apsolutni pobjednik turnira, s tri pobjede više od drugoplasiranog igrača TheMarvin, dva izjednačenja i nijednom izgubljenom partijom<sup>2</sup>. Kao što je iz istih vidljivo, igrač je izjednačio po jednu partiju protiv igrača TETKICA i VOV, i to ponavljanjem pozicija. Oba protivnika imala su po figuru više, s time da je igrač VOV bio u mnogo težoj poziciji od igrača TETKICA, odnosno krivi potez VOV-a je mogao koštati partije.

### Rezultati s igračem bez evaluacije figura

Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	TheMarvin	14	11	3	0	36
2	TETKICA	14	10	3	1	33
3	VOV	14	9	2	3	29
4	Chuckers	14	8	3	3	27
5	Vrijeme je za dddvoboj	14	5	2	7	17
6	Lockdown-Pieces	14	4	1	9	13
7	Nameero	14	2	0	12	6
8	Darko	14	0	0	14	0

Tablica 4. Rezultati turnira s verzijom igrača Lockdown bez evaluacije figura

<sup>2</sup> Rezultati turnira dostupni su na DVD-u uz ovaj rad, a nalaze se u direktoriju \Tournaments\Lockdown\_Ablation\_Original\.

Tablica 4 prikazuje rezultate turnira igranog s ovom verzijom igrača<sup>3</sup>, i već tu je vidljivo koliko je vrednovanje figura bitno u procjeni pozicije. Naspram izvornog igrača ova varijacija igra vrlo loše, a partije koje su bile dobivene, dobivene su zato što je igrač Nameero igrao iznimno loše, dok je igraču Darku isteklo vrijeme u obje partije. Jedina izjednačena partija dogodila se protiv igrača Vrijeme je za dddvoboj ponavljanjem pozicija, iako je objektivno protivnik imao daleko bolju poziciju na kraju partije, ali očito nije uspio pronaći put do pobjede.

Iz partija koje je igrač izgubio jasno je vidljivo kako igrač ne mari za svoje figure, već ih ostavlja protivniku da ih uzme. Očito tu prevaguje heuristika koja računa razliku između broja poteza, obzirom da takve pozicije ostavljaju protivniku samo jedan potez, dok ova varijacija Lockdowna ima velik broj poteza. Možda bi igrač bio i dobio još koju partiju, da mu je u srazu na sredini ploče ostalo više figura koje bi zatim promovirao u dame, obzirom da njih ipak cijeni, no do trenutka kada su dame na ploči igraču više nije preostalo dovoljno figura da bi išta mogao učiniti.

#### *Rezultati s igračem bez evaluacije dama*

Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	TheMarvin	14	10	4	0	34
2	Lockdown-Kings	14	10	4	0	34
3	TETKICA	14	7	5	2	26
4	Chuckers	14	6	4	4	22
5	VOV	14	5	5	4	20
6	Vrijeme je za dddvoboj	14	4	1	9	13
7	Nameero	14	2	1	11	7
8	Darko	14	0	0	14	0

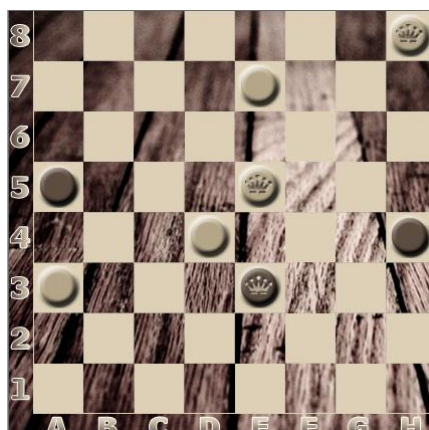
Tablica 5. Rezultati turnira s verzijom igrača Lockdown bez evaluacije dama

Iako je iz rezultata kojeg prikazuje Tablica 5 jasno da bez evaluacije dama igrač igra znatno lošije, a i intuitivno je jasno da ipak dama vrijedi više od običnog pijuna, vrijedi pogledati zašto je to tako. Obzirom da ova varijacija igrača nije izgubila nijednu partiju tokom turnira, pogledat ćemo jednu od izjednačenih partija protiv igrača TheMarvin, kojeg je izvorni Lockdown pobijedio u obje partije u prvom turniru. U izvornoj partiji, TheMarvin vs. Lockdown, vidi se kako Lockdown dolazi s velikim brojem svojih figura do drugog kraja ploče, a zatim kreće prema protivniku kojem su ostale još samo tri figure.

U izjednačenoj partiji TheMarvin vs. Lockdown-Kings, Lockdown-Kings ne mari mnogo o tome što njegov protivnik uspješno stiže do drugog kraja ploče sa svojim figurama, a sam se ne trudi pomaknuti svoje figure prema suprotnom kraju ploče (npr. figuru na polju H4 koju prikazuje Slika 5, a koja samo stoji na tom polju od poteza 32 pa do kraja igre, a čija je promocija na dubini 6 i Lockdown-Kings zasigurno vidi tu promociju, obzirom da s jednom minutom po potezu redovito stiže do dubine 8). Umjesto toga, figuru

<sup>3</sup> Rezultati ovog turnira također su dostupni na DVD-u i nalaze se u direktoriju \Tournaments\Lockdown\_Ablation\_NoPieceEval\.

koju je uspio promovirati koristi za uzimanje protivničkih pijuna (na poziciji sa slike i nekoliko poteza nakon te pozicije).



Slika 5. Pozicija nastala u igri TheMarvin vs. Lockdown-Kings, potez 65.

Nesumnjivo, Lockdown-Kings ocjenjuje svoju konačnu poziciju visokom ocjenom, obzirom da mu je ostao jedan pijun kojeg naprosto nije htio promovirati, jer mu prema ovako zakinitoj heuristici to ne povećava vrijednost pozicije.

#### Rezultati s igračem bez evaluacije poteza

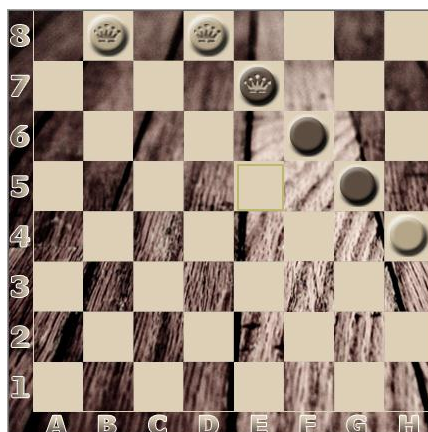
Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	TheMarvin	14	10	4	0	34
2	Lockdown-Taking	14	8	6	0	30
3	TETKICA	14	8	5	1	29
4	VOV	14	7	5	2	26
5	Vrijeme je za dddvboj	14	4	2	8	14
6	Chuckers	14	3	5	6	14
7	Nameero	14	2	1	11	7
8	Darko	14	0	0	14	0

Tablica 6. Rezultati turnira s verzijom igrača Lockdown bez evaluacije poteza

Ovo je jedan od faktora heuristike za koji nije jednostavno shvatiti zašto je uopće bitan, no Tablica 6 ipak pokazuje gori rezultat od varijacije igrača koji ne vrednuje dame više od običnih pijuna. Pogledamo li igru dame na sasvim apstraktnom nivou, možemo reći da cilj igrača nije oduzeti protivniku sve figure, već mu onemogućiti potez u sljedećem krugu (npr. ako protivniku ostane jedna figura za koju nema nijedan legalan potez, onda gubi partiju). To bi pojasnilo stvar da ranije nije rečeno (vidi poglavlje 4.1.3) da bi algoritam s takvom heuristikom trebao pretraživati veoma duboko u prostor stanja, ne bi li našao potez koji zaista i dugoročno smanjuje broj protivnikovih poteza, što je zbog broja pozicija koje bi bilo potrebno analizirati nemoguće napraviti u zadanom roku.

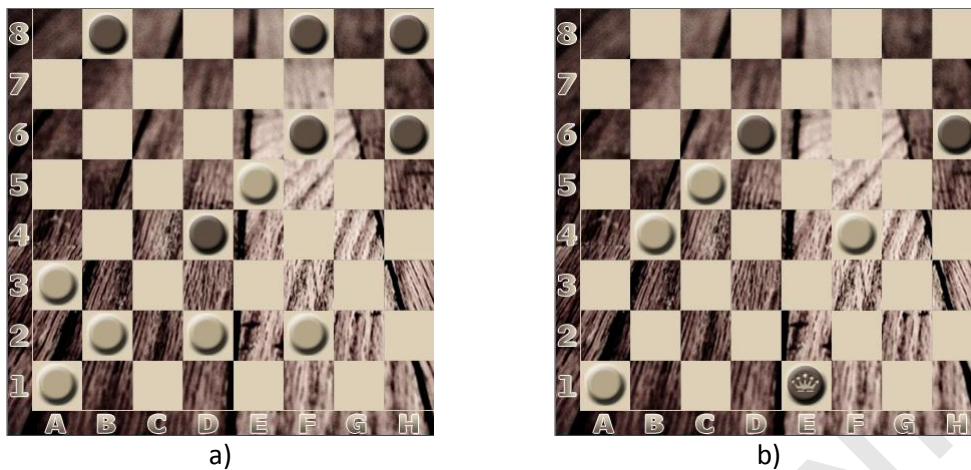
Ne bismo li objasnili ovaj fenomen, pogledajmo sporne partije, primjerice partiju VOV vs. Lockdown-Taking. Na potezu 65, Lockdown-Taking doveo se u situaciju s ograničenim opcijama, odnosno ima svega 4 legalna poteza, od kojih nijedan nije zanimljiv. Pomakne li u svom sljedećem potezu, odnosno na poziciji

koju prikazuje Slika 6, pijuna u sredini na polje E5, gubi damu i pijuna. Pomakne li desnog pijuna na F4, protivnik može pomaknuti svoju figuru na G5. Lockdown-Taking tada ima samo jedan legalan potez, a to je uzimanje figure, nakon čega opet gubi damu. Protivnik na ploči ostaje s dvije dame, a Lockdown-Taking s dva pijuna i vrlo vjerojatno će u najboljem slučaju izgubiti barem jednog od njih na putu do promocije. Pomak dame je jedina opcija, no pokuša li doći na polja G7 ili E5, opet će se dovesti do situacije u kojoj gubi barem pijuna (pa stoga izbjegava taj potez).



Slika 6. Pozicija nastala u igri VOV vs. Lockdown-Taking.

Pogledajmo kako sada izgledaju partije protiv igrača TheMarvin. Usporedi li se partija Lockdown-Taking vs. TheMarvin s partijom Lockdown vs. TheMarvin, vidljivo je da Lockdown-Taking igra vrlo zbijeno i nespretno, odnosno u usporedbi s izvornim igračem, ostavlja si vrlo malo opcija pomičući svoje figure prema rubu ploče i zbijajući ih na hrpu. S druge strane, kao crni u partiji TheMarvin vs. Lockdown-Taking, nije sposoban preuzeti inicijativu i vrlo kasno u igri počinje pomicati figure sa zadnjeg reda, što ga na kraju košta dvije figure razlike (situacija u kojoj bi izvorni Lockdown na mjestu bijelog igrača dobio partiju). Na potezu 45 (Slika 7 b) ne vidi jedini donekle spasonosni potez pijuna s D6 na E5, koji bi mu otvorio put pijuna na H6 do promocije. Na potezu 27 (Slika 7 a) koji označava prekretnicu u cijeloj igri, umjesto da onemogući protivniku uzimanje figure pomakom na G7 figurom iz kuta (čime efektivno otvara desnu stranu ploče i može jednostavno čekati da protivnik bude prisiljen napraviti loš potez), on odlučuje dopustiti protivniku uzimanje, što dovodi najprije do figure razlike, a zatim i druge.



Slika 7. Dvije pozicije nastale tokom partije TheMarvin vs. Lockdown-Taking: a) potez 27, b) potez 45.

Kao zaključak može se navesti da ovaj faktor heuristike omogućava igraču Lockdown da postigne određenu "harmoniju" igre, odnosno spretnu igru u kojoj ostavlja svoje opcije otvorenima, a protivniku zatvara sve izlaze. Dakle, ipak se pokazalo da je naziv igrača sasvim opravdan na jednom vrlo suptilnom nivou, i to upravo uključivanjem one heuristike koja je sama za sebe promašena.

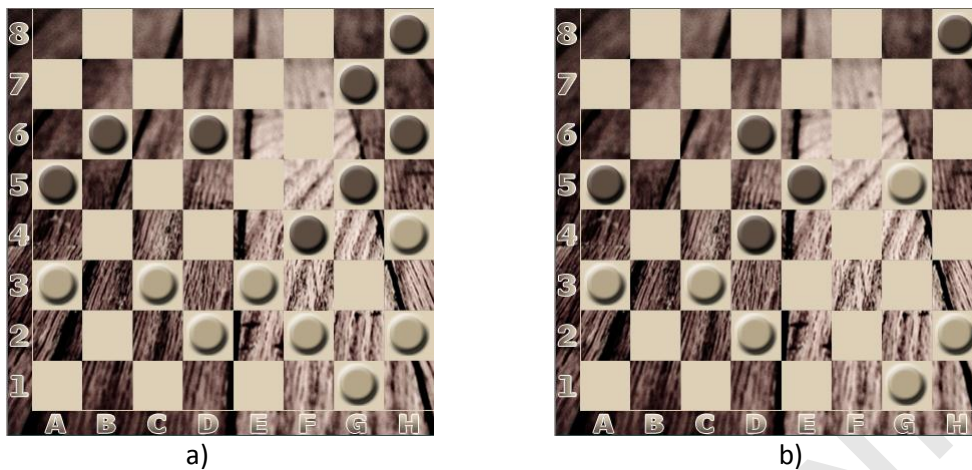
#### Rezultati s igračem bez dijela heuristike za kretanje prema protivniku

Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	TheMarvin	14	10	4	0	34
2	Lockdown-Travel	14	10	3	1	33
3	VOV	14	8	3	3	27
4	TETKICA	14	6	6	2	24
5	Chuckers	14	6	3	5	21
6	Vrijeme je za dddvoboj	14	3	1	10	10
7	Nameero	14	2	2	10	8
8	Darko	14	0	0	14	0

Tablica 7. Rezultati turnira s verzijom igrača Lockdown bez dijela heuristike za kretanje prema protivniku

Isključivanjem heuristike za pomicanje figura prema protivnikovim bilo je očekivano da će turnir završiti tako da ova varijanta igrača ima više izjednačenih partija nego izvorni Lockdown. Ono što nije bilo očekivano obzirom na rezultat izvornog igrača, a Tablica 7 to prikazuje, je jedna izgubljena partija. Krenimo najprije pogledati što se tu dogodilo.

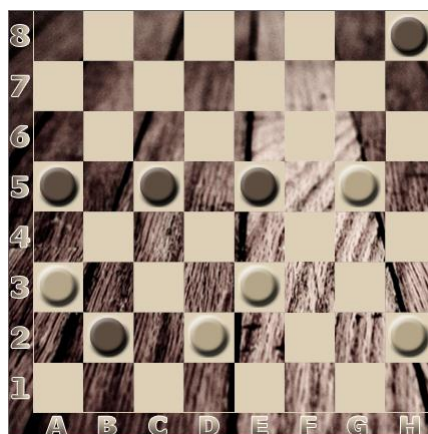
Igrač Lockdown-Travel je izgubio kao bijeli u partiji protiv igrača VOV. Gledajući partiju, u oko upada jedna velika razmjena koja počinje na potezu 28, nakon koje potezom 34 Lockdown-Travel ostaje s figurom prednosti. Međutim, igrač VOV nadoknađuje tu figuru već u potezu 37, nakon čega slijedi potpuna eliminacija bijelog, od koje Lockdown-Taking nema obrane. Vratimo li se na potez 36, odnosno poziciju koju prikazuje Slika 8, vidimo da igrač ovdje započinje svoju jurnjavu za damom figurom na lokaciji G5. Pitanje je postoji li potez kojim on može spriječiti poraz, i kakve veze taj potez ima s heuristikom gibanja figura prema protivniku.



Slika 8. Dvije pozicije nastale u igri Lockdown-Travel vs. VOV: a) potez 28, b) potez 36.

Očito je da nijedan potez s figurama na lijevoj strani nema nikakvog smisla, jer svaki vodi do gubitka svih figura na toj strani, pa pokušajmo s figurom na G1. Pomakne li se na polje F2, u sljedećem potezu VOV ima samo jedan odgovor, a to je uzimanje figure koju je prethodno napao. Naknadni pomak bijelog na polje E3 efektivno zatvara cijeli prostor na donjoj-lijevoj strani ploče. Ukoliko igrač promovira svog pijuna u damu, ne može izaći. Razmotrimo koji je VOV-ov najbolji odgovor u toj situaciji. Pomicanje s H8 nema efekta jer bijeli može napasti figuru na G7 pomakom na H6. No na tu grešku crnog bijeli ne mora tako govoriti, budući da je ta figura tada kontrolirana (ne može se pomaknuti bez da bude oduzeta pijunom na G5), pa može nastaviti sa spašavanjem situacije na donjem dijelu ploče. Pomak figure s E5 može rezultirati gubitkom te figure – ako je pomakne na polje D4 to je osobito loše za njega jer bijeli tada ima ulančani potez kojim uzima tu figuru i figuru na D6 – dakle, odlična situacija za našeg igrača. Ako pak je pomakne na F4, bijeli dobiva tu figuru pomakom s G5 na F6.

Pokušajmo pomakom figure na D6, jednim legalnim potezom za tu figuru na C5. Dobiva se situacija koju prikazuje Slika 9. Najbolji ishod za bijelog dobiva se s D2 na C3, jer se time zatvaraju gornje tri figure crnog igrača, iako mu se otvara promocija i eventualni bijeg dame. Nakon toga crni radi B2 na C1 gdje promovira pijuna, bijeli radi H2 na G3, crni C1 na D2 i bijeli E3 na F4. Crni tada mora uzeti figuru na C3, a bijeli uzima figuru na E5 i broj figura je uravnotežen i bijeli ima velike šanse doći do dame bez nepotrebnih gubitaka kakvi su se dogodili.



Slika 9. Jedna u nizu pozicija koja bi se trebala dogoditi da bi igrač Lockdown spasio svoje figure u igri protiv igrača VOV.

Pogledajmo koliko je to poteza. Na potezu 36 (Slika 8), najprije bijeli vuče G1 na F2. Zatim crni uzima s D4 na B2. Zatim bijeli vuče s F2 na E3. Pa crni D6 na C5 (Slika 9). Zatim bijeli D2 na C3. Crni B2 na C1. Bijeli H2 na G3, crni C1 na D2, bijeli E3 na F4, crni D2 na B4 i napokon bijeli F4 na D6. To je ukupno 11 poteza da bi se bijeli izvukao iz nezgodne situacije u kojoj se našao nakon razmjene. Obzirom da pretraga prostora stanja prelazi 10 poteza samo u rijetkim slučajevima kada u prvom potezu igrač ima dva moguća poteza (ponekad uspije pretražiti i 12 poteza unaprijed), možemo reći da se ovdje dogodio svojevrsan efekt horizonta. Lockdown-Taking vidio je promociju u damu na dubini 6 uz gubitak dodatne figure, ali nije vidio da na dubini 12 može ostati u poziciji s jednakim brojem figura kao i protivnik te nije ispravno mogao ocijeniti rezultirajuću poziciju, s koje je tek na dubini 22 neizbježno počeo gubiti ostatak svojih figura.

Kao što je ranije rečeno, efekt horizonta događa se kada igrač primjerice žrtvom figure odgađa neki veći, neminovni gubitak na dubinu koju algoritam nije uspio pretražiti, pa se upravo zbog te žrtve nađe u lošijoj poziciji. Ovdje se, međutim, dogodilo da igrač nije vidio mogući izlaz iz situacije u kojoj gubi, pa je promociju u damu smatrao spasonosnim potezom. Međutim, kako se pokazalo, upravo je niz poteza potreban za promociju doveo do gubitka partije.

Situacija baš nema direktne veze s ovim djelom heuristike, jer ona služi isključivo za primicanje dama protivničkim figurama, a u analiziranoj situaciji na ploči su se nalazili isključivo pijuni. Obzirom da heuristika za kretanje prema protivniku malo – ali postojano – kažnjava postojanje pijuna igračeve boje na ploči (što je bitno u završnici kada može prevagnuti broj dama nad ukupnim brojem figura), nepostojanje te kazne očito je već ranije utjecalo na odabir poteza Lockdown-Takinga (već su početni potezi igrača Lockdown u partiji protiv VOV-a drukčiji od njegove varijacije Lockdown-Taking), što je dovelo do eventualnog gubitka partije.

Na žalost, u ovom turniru nije se dogodila situacija iz koje je direktno vidljivo kako ova varijacija igrača ne pomiče svoje figure prema protivniku u završnici (naspram izvornom igraču). Lockdown-Travel je izjednačio partije protiv igrača TheMarvin, što se nije dogodilo u turniru s izvornim igračem. Međutim, te situacije sličnije su situacijama sa tom heuristikom, nego bez nje, a najvjerojatnije zato što je Lockdown-

Travel zbog nekih drugih faktora procijenio da je bolje nalaziti se bliže neprijatelju (npr. zato što to ograničava njegove opcije). Također, te pozicije vjerojatno su bile dosegljive pretragom prosotra stanja od trenutka promocije, ili igrač jednostavno nije imao ili nije vidio bolje poteze. Primjerice, u igri Lockdown-Travel vs. TheMarvin, igraču preostaje pijun na drugoj liniji, no ne vidi njegovu promociju jer se nalazi tek na dubini 12.

Zbog takve situacije, izrađen je testni primjer u koji je moguće ubaciti jednog od igrača Lockdown, ili Lockdown-Travel, da igra sam protiv sebe te vidjeti u čemu se razlikuju njihovi potezi. Pokretanje testnog primjera opisano je u dodatku. Potrebno je uzeti u obzir da je u testu moguća samo jedna vremenska postavka, i to 3 sekunde po potezu. Na sporijim računalima u tom vremenu možda ni izvorni Lockdown neće uspjeti pronaći pobjedu.

### 5.2.2. U igri protiv čovjeka i samoga sebe

Zanimljivo je testirati igrača Lockdown u igri protiv čovjeka. Dosad nije bilo moguće učiniti nešto slično jer sučelje to nije podržavalo, no u okviru ovog rada to je učinjeno i igrač se mogao ovako testirati<sup>4</sup>. Na taj način dobiva se realna procjena igračevih sposobnosti.

Kao protivnici birani su ljudi čiji je rang 1200 i više na Yahoo! Games Internetskoj stranici<sup>5</sup>. Putem nje moguće je igrati velik broj igara za jednog ili više igrača, pri čemu igrači mogu međusobno komunicirati, te povremeno organiziraju turnire. Za rangiranje igrača koristi se sustav za rangiranje Elo, prema kojem protivnici spadaju u klase C i D<sup>6</sup>. Lockdownu je dato po 10 sekundi po potezu te je uvijek igrao kao bijeli na našem sučelju (odnosno crveni na sučelju stranice) iz tehničkih razloga (sučelje Yahoo! Games zakreće ploču tako da su igračeve figure uvijek na dnu). Zbog tog problema odigrane su samo tri partije. Protivnicima nije rečeno da sudjeluju u igri protiv računala. Tablica 8 daje opise odigranih igara. Neki od protivnika predali su igru u nekom trenutku. To je funkcija koju trenutni klijent dame ne podržava i nemoguće je spremiti igru koja je u toku. Ako je nadmoć igrača Lockdown bila očita u tom trenutku, igra je odigrana do kraja na način da se do pobjede igrača Lockdown dođe što prije, tako da se igra mogla spremiti na tvrdi disk.

Igra	Rang protivnika	Ishod
1	1348	Protivnik predaje partiju na potezu 49
2	1409	Lockdown gubi
3	1270	Protivnik predaje partiju na potezu 87

Tablica 8. Igre odigrane s igračem Lockdown protiv rangiranih ljudi.

Igrajući protiv samoga sebe, Lockdown na istim postavkama na gore opisanom računalu uvijek igra istu, izjednačenu partiju<sup>7</sup>. Ovakva igra dobro može poslužiti za testiranje igrača i izjednačenje je očekivani

<sup>4</sup> Sve igre mogu se pronaći na DVD-u u direktoriju \Saved\_Games\Checkers\Lockdown\.

<sup>5</sup> <http://games.yahoo.com/board-games>

<sup>6</sup> Za više detalja o sustavu rangiranja Elo, čitatelja se upućuje na [http://en.wikipedia.org/wiki/Elo\\_rating\\_system](http://en.wikipedia.org/wiki/Elo_rating_system)

<sup>7</sup> Partija je dostupna u već navedenom direktoriju, pod nazivom Lockdown\_vs\_Lockdown.game

rezultat. Dogodi li se suprotno, to može značiti da heuristika ili sam algoritam pretrage ne funkcioniraju pravilno za obje strane, ili da igrač jednostavno igra loše.

### 5.3. Igrač Junior

Obzirom da procjenu heuristike kod igrača Junior daje neuronska mreža, nije učinjena ablacijska studija, već je konačna mreža stavljena u jedan turnir protiv svih dostupnih igrača (osim igrača Lockdown). Sudionici turnira birani su po istom ključu kao i oni za igrača Lockdown.

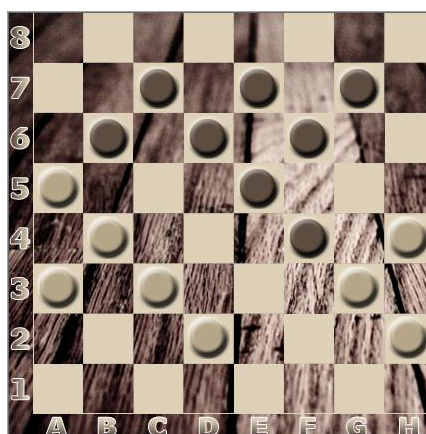
#### 5.3.1. U turniru protiv drugih igračih algoritama

Plasman	Naziv igrača	Broj igara	Pobjeda	Izjednačenja	Gubitaka	Bodova
1	TheMarvin	14	10	4	0	34
2	TETKICA	14	7	6	1	27
3	VOV	14	7	4	3	25
4	<i>Junior</i>	14	6	7	1	25
5	Nameero	14	3	6	5	15
6	Chuckers	14	3	5	6	14
7	Vrijeme je za dddvoboj	14	2	2	10	8
8	Darko	14	1	0	13	3

Tablica 9. Rezultati turnira s igračem Junior

Prema rezultatu kojeg prikazuje Tablica 9, Junior je izjednačen s igračem VOV. Zanimljivo je, međutim, što je Junior izjednačio obje partije protiv igrača TheMarvin, što je igraču VOV pošlo za rukom samo kada je igrao kao crni (drugu partiju je izgubio). Jedina izgubljena partija dogodila se protiv igrača TETKICA, dok igrač VOV ima čak tri izgubljene partije. Međutim, sortiranje – nakon bodova – u obzir uzima broj pobjeda.

Svaki od igrača, TheMarvin, TETKICA i VOV su solidni igrači. TheMarvin je u svakom turniru završavao prvi, osim u turniru protiv izvornog Lockdowna, gdje je bio drugi, dok su TETKICA i VOV izjednačili po jednu partiju čak i s igračem Lockdown.



Slika 10. Pozicija u partiji Junior vs. TheMarvin, potez 34.

Kratkim pregledom partija može se zaključiti da Junior ne igra onako "harmonično" kao Lockdown, odnosno ne zatvara protivnika toliko efektivno te se dovodi u situacije kada ne bira najbolji potez, već najmanje loš potez. Kao primjer za to može poslužiti partija Junior vs. TheMarvin, gdje na potezu 34 (Slika

10), Junior više nema dobrog poteza, nego je prisiljen protivniku ponuditi figuru bez mogućnosti nadoknade. Također se može vidjeti da mreža nije ispravno naučila kretanje prema protivniku.

### **5.3.2. U igri protiv čovjeka i samoga sebe**

Jačina Juniora nije opširnije testirana u igrama protiv čovjeka. Igrale su se igre isključivo između autora igrača i Juniora, ili između autoru bliskih prijatelja i Juniora<sup>8</sup>.

U igri Junior vs. Junior, kao i u slučaju Lockdown vs. Lockdown, vidi se ponovo jedna dosta ujednačena situacija, no s podosta više zanimljivih serija razmjena. Također se vidi igračeva nemogućnost da se kreće prema protivniku. Igru slučaja bijeli je završio u četvrtini ploče s najviše crnih figura, no dama crnog ima opciju kretanja prema toj četvrtini bijelog, što ne čini.

### **5.4. Igre Lockdown vs. Junior i Junior vs. Lockdown**

Odigrane igre<sup>9</sup> su izjednačene, što je pomalo iznenađujuće, obzirom na njihove rezultate u turniru. Ovdje se neće detaljno ulaziti u analizu tih partija.

### **5.5. Igrač Dumbo**

Jačina igrača Dumbo nije testirana, već je testirano samo njegovo ponašanje u igrama s različitim postavkama i različitim brojem igrača<sup>10</sup>. Problemi s ovim igračem koji su navedeni ranije vidljivi su iz tih igara i bilo bi ih dobro imati na umu pri izrađivanju boljih igrača.

---

<sup>8</sup> Odigrane igre nalaze se u direktoriju \Saved\_Games\Checkers\Junior\.

<sup>9</sup> Igre Lockdowna i Juniora nalaze se u direktoriju \Saved\_Games\Checkers\.

<sup>10</sup> Igre se nalaze u direktoriju \Saved\_Games\Chinese\_Checkers\Dumbo\.

## 6. ZAKLJUČAK

Težište ovog rada uvelike je stavljeno na izradu i vrednovanje igrača igara za igre dame i halma (u manjoj mjeri). Igrači su bili izvedeni tehnikama umjetne inteligencije, točnije algoritmom minimax i podrezivanjem alfa-beta.

Da bi se igrači mogli testirati, izrađena su i sučelja za spomenute igre. Svaka od igara može se pokrenuti samostalno, ili kao dio okruženja MGMTM. Razlozi za to najprije su bili edukacijske prirode, tj. da bi se omogućilo studentima da kroz učenje odmjere snage vlastitih rješenja za igrače algoritme u turnirima (do sada samo u igri dame) koji su se odvijali u sklopu prikladnih kolegija. Popularnost projekta motivirala je njegov daljnji razvoj.

Izrađeni su simulatori za obje igre, koji omogućavaju prisustvovanje različitim igračih algoritama u igri putem njenog sučelja, ili u turniru putem okružja MGMTM. Također su izrađena ukupno tri igrača, dva za igru dame i jedan za halmu. Za igrača igre dame nazvanog Lockdown izrađena je ablacijska studija i prezentirani su njeni rezultati. Za igrača Junior, drugog igrača dame, učinjeno je vrednovanje na temelju odigranog turnira protiv igračih algoritama koji su prethodnih godina sudjelovali u turniru i igara protiv igrača Lockdown, samoga sebe i nekoliko igara protiv ljudi. Igrač Dumbo, prvi igrač igre halma (ne računamo li igrača koji vuče nasumične poteze), nije ekstenzivno vrednovan, već je njegov algoritam testiran samo u različitim postavkama igre.

Svrishodnost izvedenih algoritama prema cilju davanja smjernica budućim generacijama u izradi svojih igračih algoritama, tek se mora utvrditi. Iz većeg broja pokušaja izrade igrača jasno je sljedeće:

- Kod treniranja neuronske mreže algoritmom *backpropagation* uz igranje sam protiv sebe, u slučaju igre dame problem će predstavljati izjednačene igre za koje je teško odrediti koji izlaz se traži za pojedinu poziciju.
- Genetički algoritam kojim se traži optimalna neuronska mreža pokazao se u [1] veoma uspješnim pristupom, no 100000 partija dama ne bi li se došlo do tog rezultata nije mali broj.
- Sučelja izvedena u ovom radu nisu prikladna za bilo kakva treniranja – algoritmom *backpropagation* ili genetičkim algoritmom. Razlog tomu je što cijeli proces treniranja znatno usporava prikaz partija te mehanika igre koja iz drugih razloga mora provjeravati je li potez igrača legalan, iako je igrač vratio jedan od poteza koji je ista ta mehanika ranije generirala. Treniranje Juniora (samo 9000 odigranih partija) upravo je iz toga razloga potrajalo puna četiri dana.
- Odabir heuristike i značajki pozicije koje će se ocjenjivati bitne su ne samo za odabir najboljeg poteza, već i kod efikasne eliminacije određenih grana stabla prostora stanja podrezivanjem alfa-beta. S druge strane, ispravan odabir značajki pozicije koje će se postavljati na ulaz

neuronske mreže može bitno smanjiti neuronsku mrežu, a time i broj potrebnih partija, i vrijeme potrebno za treniranje te neuronske mreže.

Uz te smjernice, učinjenu studiju te opis izrađenih igrača, mislim da je moguće izraditi progresivno jače igrače najprije za igru dame, a zatim i za halmu i bilo koje druge igre koje se razvijaju u sklopu ovog sustava u budućnosti. Sve to znači da su svi edukacijski ciljevi ostvareni, odnosno dani su temelji za izgradnju kompetentnijih rješenja. Njih pak ne bi bilo moguće ostvariti da nije uspješno ostvareno i sučelje putem kojeg je moguće simulirati igru između dva različita rješenja. Stoga tvrdim da su svi ciljevi ovog rada ostvareni.

Kao budući rad istaknuo bih manje promjene na sučelju prema korisniku radi dodavanja nekih funkcionalnosti (npr. predaja partije u slučaju kada igra čovjek protiv računala ili drugog čovjeka trenutno nije moguća, a u vrednovanju igračih algoritama pokazala se iznimno poželjnom).

Nadalje, opisana iskustva pri projektiranju rješenja igračih algoritama uzrokovala su više korijenitih promjena u dizajnu, no sada mogu poslužiti kao činjenice koje ukazuju na probleme na koje će se naići pri odabiru pojedinog dizajna. Time se ti problemi mogu lakše anticipirati i riješiti prije nego se u izradi rješenja naiđe na njih.

## 7. LITERATURA

- [1] K. Chellapilla, D. B. Fogel, "Evolving Neural Networks to Play Checkers without Relying on Expert Knowledge" *IEEE Trans. Neural Networks*, Vol. 10:6, pp. 1382-1391 (1999).
- [2] M. Lacković, "Programsko okruženje za izvođenje turnira igračih algoritama" Diplomski rad br. 1838, FER (2010).
- [3] S. Lončarić, "Neuronske mreže: Uvod" Nastavni materijali predmeta Neuronske Mreže, FER (2008).
- [4] G. F. Luger, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving" Addison-Wesley (2005).
- [5] T. M. Mitchell, "Machine Learning" McGraw-Hill Science/Engineering/Math (1997).
- [6] J. von Neumann, "Zur Theorie der Gesellschaftsspiele" *Math. Annalen.* **100**, pp. 295-320 (1928).
- [7] A. Rand, "Requiem for Man" u: A. Rand et al., *Capitalism: The Unknown Ideal*, Centennial Edition, Signet (1986).
- [8] S. Russell, P. Norvig, "Artificial Intelligence – A Modern Approach" Prentice Hall (1995).
- [9] G. Tesauro, "Practical Issues in Temporal Difference Learning" *Machine Learning* **8**, pp. 257-277 (1992).

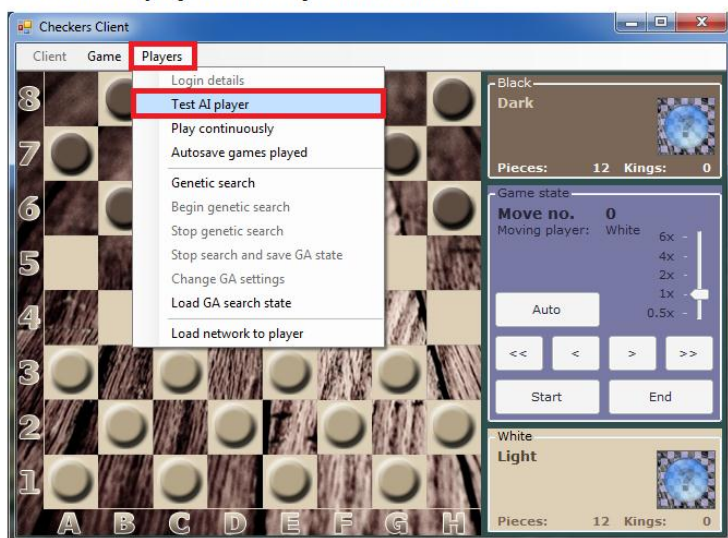
## A) Dodatak: Učitavanje i izvršavanje testnog primjera putem sučelja (dame)

Potrebno je pokrenuti izvršnu datoteku klijenta za igru dame. U izborniku kliknuti "Players" te u padajućem izborniku odabrati "Test AI player". Pokazat će se ekran na kojem se odabire igrač za testiranje, pa je potrebno odabrati igrača (dvostrukim klikom na njega).

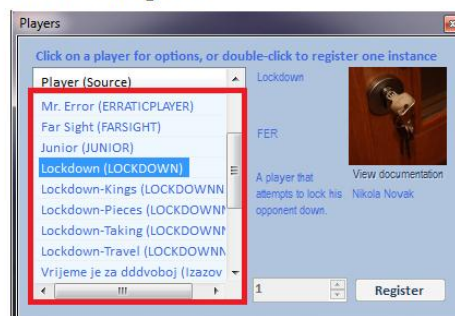
Nakon toga prikazuje se sučelje za odabir testova. Klikom na tipke "Prev" i "Next" prolazi se kroz testove. Moguće je s testa pod brojem 1 doći direktno na posljednji test klikom na "Prev". Glavni prozor pokazuje početnu poziciju svakog testa. Među testovima se nalazi mnogo njih koji su korišteni za testiranje igrača na sposobnost sudjelovanja na turniru, pa se testovi izrađeni uz ovaj diplomski rad nalaze pri kraju liste.

Nakon što je odabran željeni test, potrebno je kliknuti na tipku "Run test", i test se tada pokreće. Slike koje slijede grafički prikazuju ovdje opisane korake koje je potrebno obaviti za pokretanje testa.

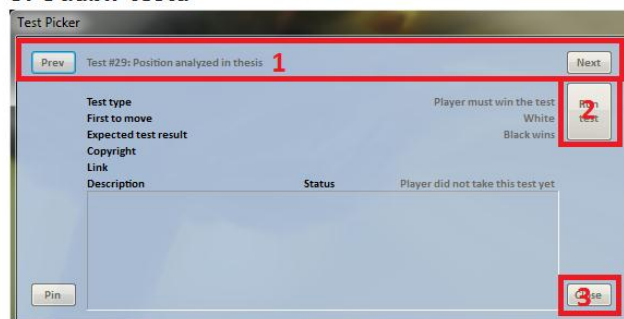
### 1. Odabir opcije testiranja u izborniku



### 2. Odabir igrača



### 3. Odabir testa



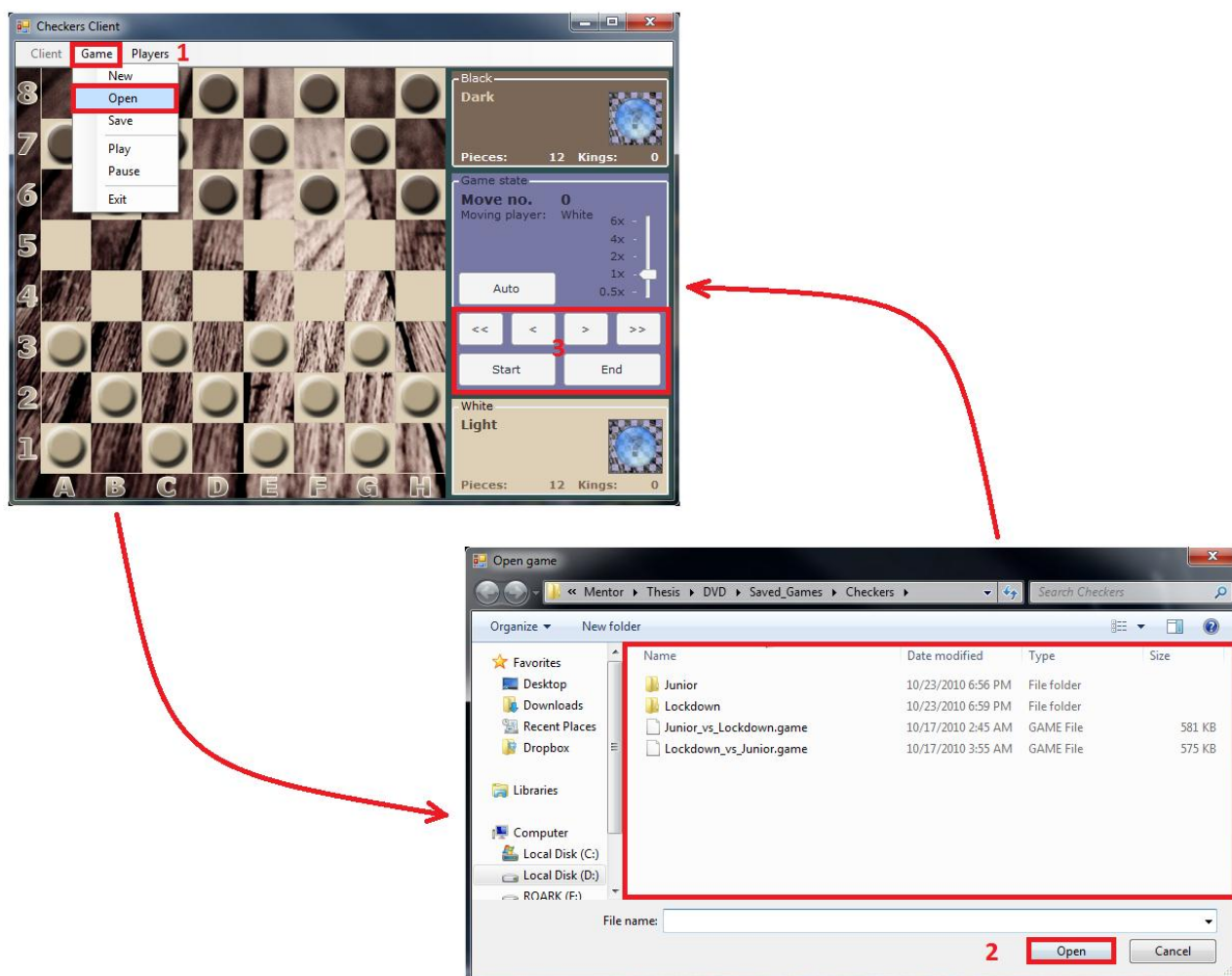
Slika 11. Grafički prikaz uz opis pokretanja testnog primjera.

## B) Dodatak: Učitavanje i pregled spremljene igre putem sučelja (dame)

Učitavanje postojeće igre dosta je intuitivno. Dovoljno je pokrenuti sučelje (klijent) za dame, odabrati "Game" u izborniku te "Open" u padajućem izborniku. Zatim je potrebno potražiti igru koju želite učitati na disku te ju odabrati. Pokazuje se početna pozicija te avatari i nazivi igrača koji su tu igru igrali.

Pregled igre moguć je pritiskom na prikladne strelice (moguć je pregled unaprijed te kretanje prema natrag, prema slijedu pozicija kako su se one događale). Klikom na tipku "Start" pregled se vraća na početnu poziciju, dok se klikom na "End" pokazuje posljednja pozicija u igri.

Donje slike grafički prikazuju opisane korake.

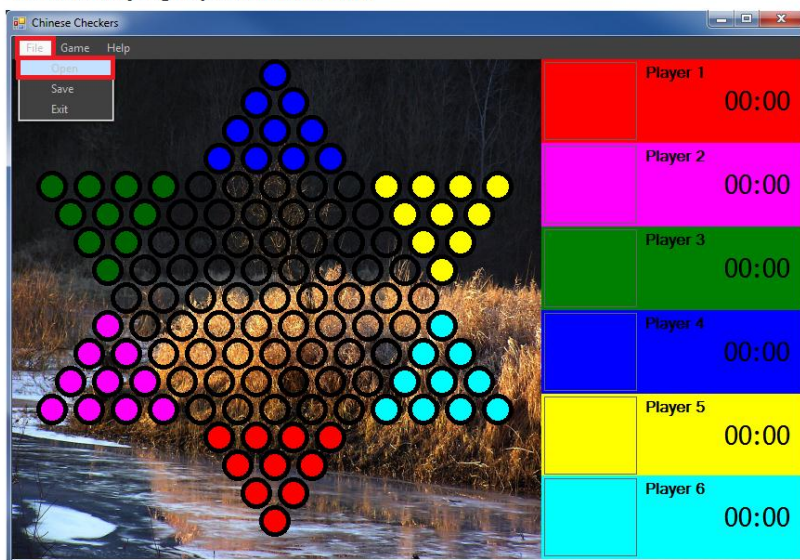


Slika 12. Grafički prikaz uz opis učitavanja igre dame.

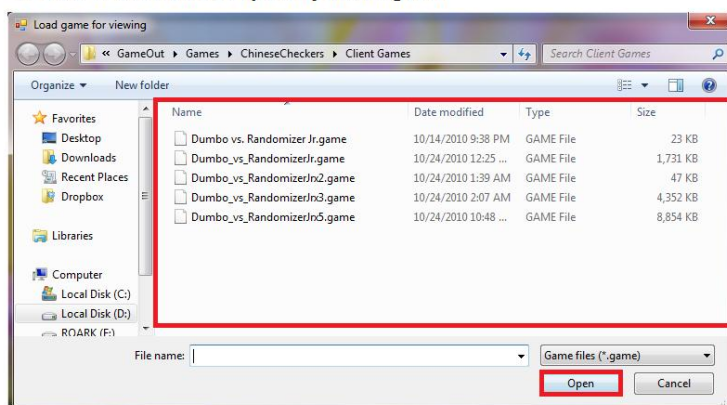
### C) Dodatak: Učitavanje i pregled spremljene igre putem sučelja (halma)

Učitavanje odigrane igre slično je kao i kod igre dame. Učitanoj igri, međutim, nije moguće ručno pregledavati (kretanjem naprijed-natrag), već se potezi automatski prikazuju na ekranu. Grafički je to prikazano na donjim slikama.

#### 1. Učitavanje igre putem izbornika



#### 2. Odabir datoteke sa spremljenom igrom



Slika 13. Grafički prikaz uz opis učitavanja igre halma.

## D) Dodatak: Lokacije spremljenih turnira i pojedinih igara na priloženom DVD-u

Svi odigrani turniri nalaze se na DVD-u u direktoriju \Tournaments\, dok su pojedinačne igre spremljene u direktorij \Saved\_Games\.

Direktorij	Sadržaj direktorija
▲ Saved_Games	Pojedinačne igre
▲ Checkers	Pojedinačne igre dame
Junior	Pojedinačne igre igrača Junior
Lockdown	Pojedinačne igre igrača Lockdown
▲ Chinese Checkers	Pojedinačne igre halme
Dumbo	Pojedinačne igre igrača Dumbo
▲ Tournaments	Turniri
▶ All_Players	Turnir svih igrača dama zajedno
▲ Junior	Turniri igrača Junior
▶ Junior_Test	Testni turnir igrača Junior
▲ Lockdown	Turniri igrača Lockdown i svih njegovih varijanti
▶ Lockdown_Ablation_NoKingEval	Turnir varijacije igrača Lockdown bez evaluacije dama
▶ Lockdown_Ablation_NoMoveEval	Turnir varijacije igrača Lockdown bez evaluacije poteza
▶ Lockdown_Ablation_NoPieceEval	Turnir varijacije igrača Lockdown bez evaluacije pijuna
▶ Lockdown_Ablation_NoTravel	Turnir varijacije bez heuristike za kretanje prema protivniku
▶ Lockdown_Ablation_Original	Turnir izvornog igrača Lockdown

Tablica 10. Popis direktorija na priloženom DVD-u koji sadrže turnire i pojedinačne igre.

Tablica 10 prikazuje popis svih direktorija i kratki opis njihovih sadržaja. Svaki direktorij koji sadrži neki od odigranih turnira kao poddirektorij sadrži direktorij \Games\ koji sadrži pojedinačne igre odigrane u tom turniru. Njih je moguće pojedinačno učitati putem prikladnog klijenta.

## SAŽETAK

Algoritmi umjetne inteligencije mogu se primijeniti u izradi igračih algoritama od jednostavnih igara s malim prostorom stanja, do kompleksnijih poput šaha. Potpuno razumijevanje principa po kojem funkcioniraju i njihovo poboljšavanje nije moguće bez iskustva u izradi njihovih osnovnih inačica. Čak i tada, bez mogućnosti usporedbe sposobnosti vlastitog rješenja, nije moguće znati koliko je algoritam zaista dobar. Sučeljavanje različitih rješenja, pa i sučeljavanje algoritma s čovjekom, predstavlja zanimljivi način da se sazna stvarna snaga danog algoritma.

Rezultat ovog rada i provedenih studija je upravo jedno okruženje putem kojeg je moguće provoditi takva testiranja izrađenih algoritama. Izrađene su igre dame i halma, i igrači za svaku od njih. Provedene su studije nad igračima dama pomoću kojih se ispitala njihova kompetentnost u igrama protiv već postojećih igračih algoritama, ali i protiv čovjeka. Time se uz minimalno truda omogućava testiranje drugih rješenja u igri protiv algoritama čija je jačina već poznata. Iako je fokus stavljen na rješenja igračih algoritama za igru dame, izrađen je i igrač halme koji se može koristiti kao svojevrsna polazna točka u izradi algoritama koji pretražuju ogroman prostor stanja. Sučelja samih igara prema korisniku intuitivna su i jednostavna za korištenje, a za obje je igre također izrađena komponenta koja ih spaja s okruženjem MGMTM čime se omogućuje i organizacija turnira. Obje igre i svi igrači implementirani su u programskom jeziku C#.

*Ključne riječi: umjetna inteligencija, igrači algoritam, neuronska mreža, genetski algoritam, minimax, podrezivanje alfa-beta, dame, halma*

## ABSTRACT

Artificial intelligence algorithms can be used in creating game playing algorithms for the simplest of games that have a small state space, to the more complex ones like chess. A complete understanding of the principles of their function as well as their improvement is not possible without first-hand experience in creating their most basic variations. Without being able to compare the capabilities of a solution to some reference, even then it is impossible to know how good an algorithm actually is. Playing a game against some other algorithm, or even against a human then presents an interesting way to find out the real strength of a given algorithm.

The result of this thesis and the studies presented within is an environment which enables just such testing. The games that were created for this purpose are checkers and Chinese checkers (Stern-halma) and the players for each. Studies that were done on checkers players examined their competence in games against existing playing algorithms and against human players. This enables the testing of new algorithms against algorithms whose strength is already known with almost no effort. Although checkers players were the focus of the studies, a Chinese checkers player was also created and it can be used as a kind of starting point in the making of the playing algorithms that search a vast state space. User interfaces of the games are fairly simple and intuitive, and a component was made for each game which connects them to the MGMTM environment. This enables the organization of tournaments. Both games and all players were implemented using C# programming language.

*Keywords: artificial intelligence, playing algorithm, neural network, genetic algorithm, minimax, alpha-beta pruning, checkers, chinese checkers*