

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tine Benko

REALIZACIJA IGRE POKER NA MIZI RFID

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Mentor: doc. dr. Mira Trebar

Ljubljana, 2012

Rezultati diplomskega dela so intelektualna lastnina študenta – avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.



Št. naloge: 00231/2012

Datum: 04.04.2012

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **TINE BENKO**

Naslov: **REALIZACIJA IGRE POKER NA MIZI RFID**
POKER GAME REALIZATION ON RFID TABLE

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj pregleda in analizira tehnologije in programska orodja za izvedbo igre poker na mizi RFID. V diplomski nalogi naj opiše načrtovanje mize RFID, pravila igre in ustrezno strojno opremo za razvoj aplikacije. Za realizirano rešitev naj bo na predstavitvenem zaslonu prikazan potek igre z nekaj primeri in opisane naj bodo njene prednosti in morebitne slabosti.

Mentor:


doc. dr. Mira Trebar

Dekan:


prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Tine Benko,

z vpisno številko 63050012,

sem avtor diplomskega dela z naslovom:

Realizacija igre poker na mizi RFID

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
doc. dr. Mire Trebar
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 01.07.2012

Podpis avtorja:

ZAHVALA

Doc. dr. Miri Trebar se zahvaljujem za mentorstvo in pomoč pri izdelavi diplomske naloge.

I would like to express a special acknowledgement to Mr. Roger Webber, Chief Executive Officer at Magellan Technology Pty Ltd, who made it possible for us to use Magellan's technology for academic use.

Staršem se zahvaljujem za podporo med študijem.

Dodatno se zahvaljujem vsem, ki ste mi polepšali čas študija.

KAZALO VSEBINE

POVZETEK	1
ABSTRACT	2
1 UVOD	3
2 TEHNOLOGIJE IN UPORABLJENA ORODJA	5
2.1 RFID	5
2.1.1 Opis in zgodovina	5
2.1.2 Frekvenčna območja	6
2.1.3 Značka	6
2.1.4 Čitalnik in antene	7
2.2 Digitalno oglaševanje	8
2.2.1 Urejanje vsebine	8
2.2.2 Predvajanje	9
2.2.3 Programska oprema G-Digital	9
2.3 Java	9
2.4 NetBeans	10
3 IGRA POKER IN PRIKAZ NA ZASLONU	11
3.1 Dogajanje na turnirju	11
3.1.1 Potek igre	12
3.1.2 Delo organizatorjev	14
3.2 Prikaz podatkov na zaslonu	14
4 IZVEDBA MIZE RFID	17
4.1 Uporabljena strojna oprema	18
4.1.1 Čitalnik	18
4.1.2 Antene	19
4.1.3 Značke	20

4.2	Izdelava mize.....	21
4.2.1	Razporeditev anten.....	21
4.2.2	Izdelava okvirja.....	23
4.2.3	Izdelava vmesnika za delivca.....	23
4.2.4	Namestitev anten in čitalnika.....	25
4.2.5	Konfiguracija čitalnika.....	26
4.3	Izdelava programa	31
4.3.1	Opis	31
4.3.2	Shema delovanja	32
4.3.3	Določitev izhodnih podatkov	33
4.3.4	Komunikacija s čitalnikom	34
4.3.5	Obdelava prejetih podatkov	38
4.4	Prikaz na zaslonu.....	41
4.4.1	Urejevalnik vsebine.....	41
4.4.2	Predvajalnik	46
5	SKLEPNE UGOTOVITVE.....	49
6	LITERATURA	51

UPORABLJENE KRATICE

BB	<i>Big Blind</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
HF	<i>High Frequency</i>
LAN	<i>Local Area Network</i>
LF	<i>Low Frequency</i>
RCA	<i>Radio Corporation of America</i>
RFID	<i>Radio-Frequency Identification</i>
SB	<i>Small Blind</i>
UHF	<i>Ultra-High Frequency</i>
USB	<i>Universal Serial Protocol</i>
XML	<i>Extended Markup Language</i>

POVZETEK

V diplomskem delu smo predstavili izdelavo mize za igro pokra, ki z uporabo tehnologije RFID omogoča pridobitev podatkov o poteku igre in njihov prikaz na oddaljenih zaslonih. Predstavili smo dinamiko organizacije in igre pokra na turnirju. Ugotovili smo, da je zaradi lastnosti igre težko spremljati potek, kar omeji organizatorje pri prikazovanju informacij na zaslonih le na osnovne podatke o turnirju. Zato smo želeli vzpostaviti sistem samodejnega zajema, obdelave in prikaza podatkov na zaslonih, ki na takšen način postanejo zanimivi širšemu krogu ljudi. Preučili smo delovanje naprav z uporabo tehnologije digitalnega oglaševanja in RFID ter zasnovali delovanje sistema. Izdelali smo mizo na kateri se odvija igra pokra, v njej pa so vgrajene antene RFID, ki omogočajo prepoznavo igralnih pripomočkov z vgrajenimi značkami RFID. Uporabo in nastavljanje strojne opreme smo načrtovali z mislijo na hiter in zanesljiv način delovanja aplikacije za obdelavo podatkov. Z uporabo pravil igre pokra smo pripravili datoteko z informacijami o poteku igre, do katere dostopajo predvajalniki za digitalno oglaševanje in jih prikažejo na zaslonu. Tako gledalci lahko neposredno spremljajo grafični prikaz igre na oddaljenih zaslonih.

Ključne besede: RFID, poker, turnir, digitalno oglaševanje, oddaljeni zaslon

ABSTRACT

In this diploma thesis we presented the development of a poker table for live tournaments, which allows to obtain and to present data about the game on remote displays through RFID technology. We described the dynamics of organization and game flow involved in live poker tournaments. We have found out that it is difficult for organizer to monitor the game flow on a particular table that limits him to presentation of only basic tournament data on the displays. This is why we wanted to establish a system of automated detection, processing and graphic presentation of data on remote display, making the game more interesting to watch. We have examined devices used in RFID technology and Digital Signage and prepared the basics for the system design. After that, we have built a poker table with integrated antennas used to detect movement of gaming accessories with built-in tags. The RFID hardware devices were configured and used to meet the requirements of data processing application on speed and reliability. Implementing poker rules in the application was necessary to generate a file that could be accessed by digital signage players. Thus the audience is able to monitor the graphic presentation of the game on remote displays.

Keywords: RFID, poker, tournament, digital signage, remote display

1 UVOD

Poker je zadnje čase vedno bolj priljubljena igra, ki jo večina igralcev igra preko interneta, drugi ga igrajo v igralnicah, najboljši pa na velikih turnirjih. Zaradi velikih denarnih vložkov je igra zanimiva za veliko število ljudi.

Igro spremljajo prireditelji, igralci in obiskovalci. Zanimajo jih podatki o številu igralcev v igri, številu žetonov določenega igralca, s kakšnimi kartami je odigral igro, na katerem mestu je v primerjavi z ostalimi igralci. Zaradi velikega števila miz opazovalec ne more spremljati vseh na enkrat in potrebuje prikaz podatkov na zaslonu. Za zbiranje podatkov je potrebno najti način za detekcijo vseh parametrov igre.

Tehnologija RFID (ang. *Radio Frequency Identification*) omogoča označevanje predmetov z značkami (ang. *RFID tag*), ki jih čitalnik (ang. *RFID reader*) lahko brezkontaktno prebere. Vsaka vsebuje svojo unikatno kodo in pomnilnik, ki ga lahko poljubno programiramo. Na takšen način čitalnik omogoča identifikacijo predmetov v svojem dosegu in komuniciranje z njimi.

Orodja za digitalno oglaševanje (ang. *Digital Signage*) omogočajo prikaz multimedijskih vsebin na oddaljenih zaslonih, ki jih je mogoče poljubno konfigurirati. Predvajalniki prenesejo potrebne datoteke iz centralnega strežnika ali oddaljenih virov in nato vsebino lokalno predvajajo.

Cilj naloge je izdelati mizo za igro pokra z živim prikazom igre. Miza naj prepozna žetone in karte na posameznem igralnem mestu in spremlja premike po igralni površini. Iz dobljenih podatkov naj aplikacija spremlja potek igre in stanje žetonov vseh igralcev. Podatke osvežuje z vsakim premikom značke in jih izvozi v podatkovno strukturo, do katere dostopajo predvajalniki in prikazujejo potek igre na posameznem zaslonu.

2 TEHNOLOGIJE IN UPORABLJENA ORODJA

Pri izdelavi naloge smo uporabili tehnologijo RFID za izdelavo mize ter digitalno oglaševanje za grafični prikaz igre na zaslonu. Za delovanje obeh posameznih komponent smo razvili program v programskem jeziku Java z uporabo integriranega razvojnega okolja NetBeans.

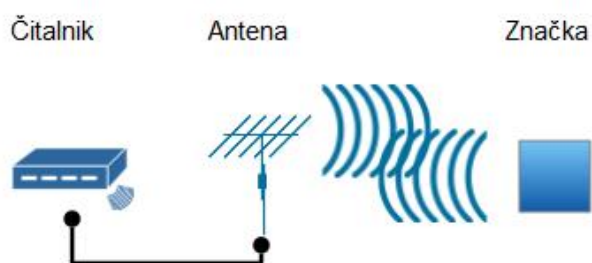
2.1 RFID

RFID je tehnologija, ki uporablja radio-frekvenčno in elektromagnetno polje za brezkontaktni prenos podatkov med značko in čitalnikom. Značka vsebuje enolično oznako, ki jo čitalnik lahko prebere in se uporablja za identifikacijo in sledenje predmetov [1].

2.1.1 OPIS IN ZGODOVINA

Začetki tehnologije segajo v obdobje druge svetovne vojne. Aplikacija »*Identify Friend of Foe*« je bila razvita v Angliji za potrebe vojske in je bila namenjena ločevanju med lastnimi in sovražnikovimi letali. Letalo je po sprejetem signalu oddajnika odgovorilo s svojo kodo, ki jo je oddajnik lahko prepoznal. Z uporabo sistema so odkrili pomanjkljivosti kot je zamenjava lastnega za sovražnikovo letalo v primeru okvare oddajnika na letalu ali izgube ključa. Kasneje je bila uporabljena za sledenje vojaške opreme in osebja [2].

V letu 1973 je Mario Cardullo patentiral prvi pasivni RFID sistem z uporabo pasivne značke. Sestavljen je iz baze (čitalnik RFID in antena) in sprejemnika (značka RFID). Baza odda izpraševalni signal (*interrogation signal*), ki ga zazna oddaljen sprejemnik (slika 1). Vsebuje spremenljiv pomnilnik, ki ga lahko na osnovi prejetega signala spremeni ali pa odda podatke nazaj bazi. Sprejemnik dobi energijo za delovanje iz prejetega zasliševalnega signala [3].



Slika 1: Sistem RFID.

Razvoj se je začel z uporabo tehnologije v komercialne namene. Laboratorij *Los Alamos Scientific Laboratory* je izdelal napravo za sledenje prometa, ki jo je naročil *Port Authority of New York and New Jersey*. Kasneje so bile razvite aplikacije za identifikacijo izdelkov v trgovinah, zaščito proti kraji, kontrolo dostopa, transport in logistiko ter identifikacijo predmetov [1].

2.1.2 FREKVENČNA OBMOČJA

Značke in čitalniki uporabljajo za komunikacijo različna frekvenčna območja [1]:

- LF 125-134 kHz (ang. *Low Frequency*)
- HF 13,56 MHz (ang. *High Frequency*)
- UHF 433MHz (ang. *Ultra-High Frequency*)
- UHF 868 in 915MHz (ang. *Ultra-High Frequency*)
- Mikrovalovi 2,54 in 5,8 GHz (ang. *Microwave*)
- Mikrovalovi 3,1 in 10 GHz (ang. *Microwave*)

Pri uporabi različnih frekvenc se moramo soočiti z različnimi fizikalnimi lastnostmi. Za nizke frekvence velja, da imajo manjši doseg in manjšo odpornost na motnje v okolju. Zato so predvsem uporabljene s pasivnimi značkami v neposredni bližini antene – do enega metra.

Uporaba visokih frekvenc se izogne problemu z motnjami v okolju in omogoča daljši doseg, vendar navadno zahteva uporabo aktivnih značk.

Izbira primerne frekvenčnega območja delovanja je odvisna od zahtev aplikacije s stališča razporeditve značk in anten v prostoru, hitrosti branja in pisanja na značke, možnih motenj v okolju in varnosti.

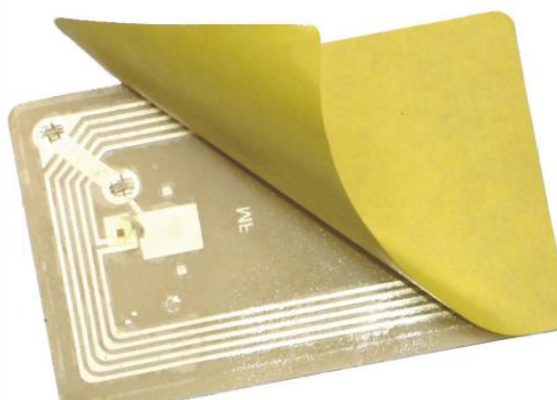
2.1.3 ZNAČKA

V tehnologiji RFID se uporabljata dve skupini značk:

- Aktivne značke
- Pasivne značke

Aktivno značko sestavljata mikročip z anteno in baterija za napajanje. Zaradi dodatne baterije je značka dražja in nameščena na večjem nosilcu kot pri pasivnem RFID. Prednost uporabe aktivnih značk je predvsem večji domet, vendar postanejo ob iztrošeni bateriji neuporabne, kar je poleg cene glavna omejitev pri uporabi značk takšne vrste.

Pasivna značka je podobno kot aktivna sestavljena iz mikročipa in antene, ki sta nameščena na nosilcu (slika 2). Mikročip dobi napajanje za delovanje in oddajo odgovora iz elektromagnetnega polja čitalnika. Iz tega sledi, da mora biti značka za svoje delovanje v dosegu antene. Po eni strani to onemogoča delovanje na dolge razdalje, po drugi strani pa omogoča, ob pravilni postavitvi anten, natančno spremljanje lokacije posamezne značke. Zaradi svoje velikosti je značka lahko nameščena na zelo majhen nosilec, kar omogoča enostavno uporabo v različnih okoljih [1].



Slika 2: Pasivna RFID značka.

2.1.4 ČITALNIK IN ANTENE

Čitalnik RFID značk je sestavljen iz antene in strojne opreme za modulacijo in demodulacijo signala pri komunikaciji z značkami. Za komunikacijo z uporabnikom ali drugimi aplikacijami mora imeti tudi vhodno – izhodne priključke. Vgrajen krmilnik skrbi za kontrolo strojne opreme in implementacijo komunikacijskih protokolov na zunanjih priključkih [4].

Antene so oblikovane namensko v odvisnosti od čitalnika in želene uporabe. Razvijalci jih stalno izboljšujejo, da bi z manj porabljene moči dosegli večji domet. Čitalnik lahko uporablja istočasno tudi dve anteni, eno za sprejemanje podatkov in drugo za pošiljanje.

Ker je v dosegu antene istočasno lahko tudi več značk, čitalniki za dostop do posamezne uporabljajo različne protokole za izogibanje kolizijam. Najbolj pogosto sta uporabljena verjetnostni in deterministični [1].

Verjetnostni algoritmi uporabljajo principe ALOHA protokola. Značka lahko pošilja podatke, če je komunikacijski kanal prost. Če je zaseden, ker več značk hkrati pošilja podatke, se promet ustavi, značke pa nato po naključno dolgem času ponovno pošljejo podatke in upajo,

da je kanal prost. Takšen algoritem ima omejeno maksimalno število značk na posamezni anteni, saj se verjetnost prostega kanala zmanjšuje ob dodajanju novih. Zaradi tega pride do več kolizij in je število zaznanih značk sorazmerno s hitrostjo komunikacijskega kanala.

Deterministični algoritmi pogosto uporabljajo sprehod po binarnem drevesu. Vsaka značka ima svojo unikatno kodo, po kateri jih naslavlja čitalnik. Tako lahko določi, katera značka bo odgovorila ob posameznem zasliševalnem signalu. Najprej zahteva, da se mu oglasijo vse značke s prvim bitom enakim 0. V primeru, da se oglasi več značk, čitalnik ne more razbrati podatkov, zato v naslednjih korakih pošlje vedno bolj specifično zahtevo, dokler se mu ne oglasi ena sama značka. Ko konča delo z izbrano značko, algoritem nadaljuje sprehod po drevesu. Na takšen način čitalnik stalno ve, katere značke ima v dosegu in jih lahko nato direktno naslavlja, kar pripomore k zmanjšanju kolizij in možnosti uporabe velikega števila značk na eni anteni.

2.2 DIGITALNO OGLAŠEVANJE

Zaslon za digitalno oglaševanje je namenjen prikazovanju televizijskih programov, menijev, informacij, posnetkov in drugih multimedijskih vsebin. Srečamo jih na javnih mestih, v trgovinah, hotelih, restavracijah in postajah javnega prevoza [5].

2.2.1 UREJANJE VSEBINE

Urejevalnik vsebine uporabljamo za uvoz multimedijskih datotek, konfiguracijo zaslona in pošiljanje datotek predvajalniku (slika 3).



Slika 3: Uporaba multimedijskih datotek na zaslonih za digitalno oglaševanje.

Uvozimo in pošljemo lahko datoteke, ki se predvajajo na predvajalniku (slike, filmi, glasbene datoteke) ali pa aktivne vsebine. Te omogočajo predvajalnikom pridobivanje informacij iz različnih virov, do katerih imajo dostop.

2.2.2 *PREDVAJANJE*

Predvajalnik sprejme datoteke in opis strukture predvajanja na zaslonu. Struktura predvajanja vsebuje informacije o postavitvi posameznih elementov na zaslon, njihovem vrstnem redu in trajanju.

2.2.3 *PROGRAMSKA OPREMA G-DIGITAL*

Programsko opremo G-Digital, ki smo jo uporabili pri izdelavi naloge razvija podjetje G-D d.o.o in je v prvi vrsti namenjena digitalnemu oglaševanju in informiranju [7]. Omogoča enostavno upravljanje z zasloni na oddaljenih lokacijah. Z uporabo vgrajenih orodij lahko napredno urejamo prikaz na posameznem zaslonu. Zaslon razdelimo na različne regije, v katerih hkrati prikažemo različne vsebine. Vsebine lahko predhodno uvozimo v sistem in so statične, lahko pa uporabimo dinamične, do katerih neodvisno dostopa posamezen predvajalnik. Na takšen način v konfiguracijo zaslona vključimo spletne strani in datoteke ter podatke, ki so dostopni na strežnikih. Tako predvajalnik podatke iz različnih podatkovnih baz prikaže kot smo mu vnaprej predpisali. Na zaslonu se nam nato vsebine dinamično spreminjajo brez potrebe po ročnem urejanju.

2.3 JAVA

Programski jezik Java so začeli razvijati v podjetju Sun Microsystems v letu 1990. Je objektno naravnan jezik, razvit na osnovi C++, vendar so pri razvoju poskrbeli za enostaven objektni model in manjše število nizko-nivojskih objektov. Javanski programi se v nasprotju z nekaterimi ostalimi programskimi jeziki ne prevedejo neposredno v strojno kodo, ki bi omogočila takojšnje izvajanje na strojni opremi, ampak v vmesno kodo (ang. *bytecode*, *class file*). Za izvajanje vmesne kode je potreben tolmač (ang. *interpreter*), ki le-to spremeni v strojno kodo, prilagojeno za okolje v katerem teče. Na ta način je dosežena prenosljivost med

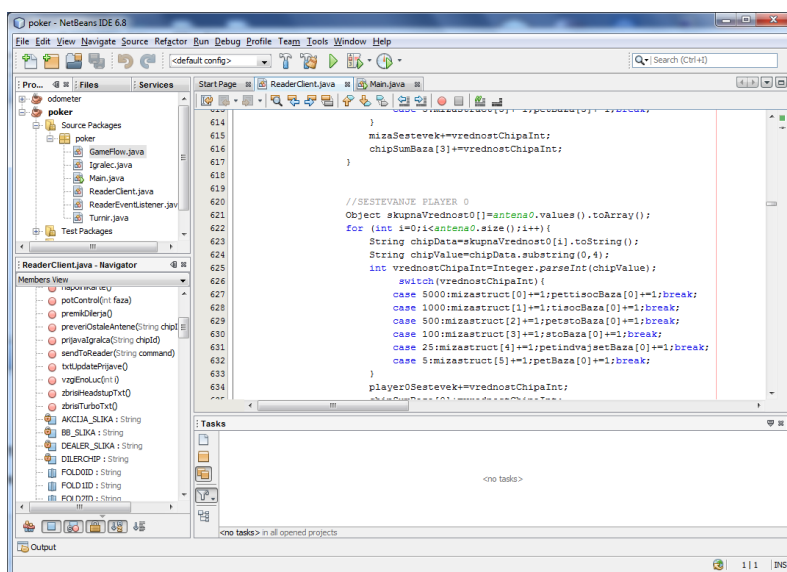
različnimi platformami, saj lahko tolmači na različnih platformah izvajajo isto vmesno kodo [8].

Java vsebuje veliko programsko knjižnico, ki jo pri delu uporabimo za hitro in zanesljivo izdelavo vmesnika za povezavo s strojno opremo. Z uporabo objektnega načina programiranja zagotovimo varnost in robustnost programa, hkrati pa nam omogoča enostavno nadgradnjo in razširitev z dodatnimi programskimi objekti.

2.4 NETBEANS

Za izdelavo programa v Javi, programerju zadostuje uporaba urejevalnika besedila, prevajalnika in tolmača. Za enostavnejše delo uporabimo integrirano razvojno okolje, ki vključuje vse našete potrebščine, dodatno pa programerju ponuja velik izbor orodij.

NetBeans je zbirka orodij za programiranje v Javi. Urejevalnik besedila poleg obarvanja kode programerju med pisanjem omogoča avtomatično dopolnjevanje in preverjanje sintaktične pravilnosti (slika 4). Z označevanjem delov, kjer se pojavlja določena spremenljivka, nam olajša pregled nad celotno kodo in njenim izvajanjem. Z uporabo razhroščevalnika dobimo še boljši vpogled v izvajanje kode in hitreje najdemo del kode, ki se ne izvaja po naših pričakovanjih. Z orodjem za vodenje projektov enostavno upravljamo z izvirno kodo, morebitnimi dodatnimi vključki in nastavitvami izvajanja posameznega projekta, ki jih potrebujemo ob izdaji programa. Ker napisano kodo v razvojnem okolju tudi izvajamo, lahko pravilno delovanje preverjamo med samim pisanjem kode [9].



Slika 4: Integrirano razvojno okolje NetBeans.

3 IGRA POKER IN PRIKAZ NA ZASLONU

Poker je igra, ki jo igralci igrajo s kartami in žetoni. Obstaja več različic igre, vendar za vse veljajo podobna osnovna pravila. Vsak igralec dobi določeno število kart, s katerimi sestavi svojo najboljšo kombinacijo. S to kombinacijo tekmuje proti ostalim igralcem in pri tem stavi svoje žetone. Kdor ima na koncu najboljšo kombinacijo, zmaga in pobere žetone.

Igra se je v zadnjem času zaradi dostopnosti širšemu krogu ljudi in enostavnosti prirejanja preselila na internet. Igra poteka na strežnikih prirediteljev, igralci pa odigrajo svoje poteze preko uporabniškega vmesnika. Tako organizatorji z nižjimi stroški prirejajo turnirje za veliko število ljudi, saj ne potrebujejo prostorov z mizami, ni jim potrebno zaposlovati osebja, igralci pa udobneje igrajo poker z uporabo domačega računalnika.

Veliki turnirji so se zato ohranili predvsem zaradi promocije internetnih ponudnikov. Organizatorji morajo skrbeti za informiranje igralcev in gledalcev na turnirju ter posneti video vsebine, ki jih nato predvajajo po televiziji. V kolikor hočejo gledalcem ponuditi atraktivno spremljanje igre, morajo posnetke naknadno obdelati s podatki o vsakem igralcu in poteku igre.

3.1 DOGAJANJE NA TURNIRJU

Pri turnirski izvedbi igre navadno sodeluje veliko število igralcev (slika 5). Pred začetkom turnirja se morajo igralci nanj prijaviti. Na podlagi prijav so igralci naključno razporejeni za mize, kjer prejmejo žetone za igro. Ko igralec izgubi svoje žetone, izpade iz igre. Na njegovo mesto se presede igralec z druge mize. Igra poteka, dokler ne izpade zadnji igralec in zmagovalec osvoji vse žetone. Organizacija takšnih dogodkov je zapletena, saj morajo prireditelji stalno spremljati koliko žetonov ima posamezen igralec, da ob presedanju ohranjajo ravnovesje med mizami.



Slika 5: Turnir pokra [10].

3.1.1 POTEK IGRE

Igra v živo poteka na igralni mizi. Igralci so razporejeni na posameznih igralnih mestih, na sredini pa ima prostor delivec kart (ang. *dealer*). Ta skrbi za deljenje kart in preverjanje pravilnega polaganja stav ter poteka igre. Deljenje kart opravlja namesto igralca, ki ga pri igri označimo z ploščkom (ang. *button*) na katerem je napis *dealer*. Ker sta oba izraza v angleškem jeziku enaka, bomo v nadaljevanju za označevanje igralca in njegove pozicije uporabljali izraz plošček, za osebo, ki poleg igralcev sodeluje pri igri pa delivec.

Celoten potek posamezne igre je drugačen ob vsaki ponovitvi, vendar lahko določimo osnovne elemente s pomočjo katerih opišemo igro:

- Identiteta igralca
- Število žetonov vsakega igralca
- Koliko žetonov je stavil posamezen igralec
- Koliko žetonov so stavili vsi igralci
- Kdo je plošček
- Kateri igralec je trenutno na vrsti
- Kateri igralec je odstopil
- Skupne karte na mizi
- Karte igralcev ob koncu igre

Identiteta igralca je pomemben podatek ob prijavi igralca na turnir in kasnejši razporeditvi za mize. Med igro jih organizatorji poimensko presedajo med različnimi mizami, na koncu turnirja pa je rezultat predstavljen v tabeli igralcev in njihovi uvrstitvi.

Igralci ob začetku igre prejmejo žetone različnih vrednosti. Pred seboj jih zlagajo v kupček, ki ga zmanjšujejo ob položitvi stav v igro, povečujejo pa ob zmagi v igri. Ker igra poteka hitro, igralec za lažje štetje žetone zлага na različne kupčke, ki jih sestavi z žetoni iste vrednosti. Tako zložene žetone enostavno preštejejo tudi ostali igralci na mizi, delivec in opazovalci. V primerih, ko ima igralec pred seboj zelo veliko žetonov ali pa so ti neurejeni, žetone lahko prešteje samo igralec sam.

Ko je posamezen igralec v igri na vrsti, lahko stavi. Navadno pove, koliko želi staviti, nato položi žetone iz kupčka na mizo. V primeru, ko igralec ne sestavi točnega zneska z svojimi žetoni, mu delivec ali soigralec lahko zamenja žetone za manjše vrednosti. Pravilno število žetonov preveri delivec in prepusti igro naslednjemu igralcu. Podatek o žetonih, ki jih je stavil posamezen igralec, je zanimiv za vse sodelujoče pri igri: za igralce, saj nočejo biti oškodovani v primeru premajhnega vložka, za organizatorje, saj morajo preverjati pravilnost igre in za gledalce, ki hočejo videti potek igre.

Delivec pred začetkom določi, kateri igralec bo plošček v prvi igri. Vsakemu igralcu razdeli po eno karto, plošček pa dobi igralec z najvišjo. Za spremljanje igre je to pomemben podatek, saj določa pozicijo igralca, od katere je odvisen potek igre. Določa igralca, ki začne igro ali pa mora pred igro vplačati slepo stavo (ang. blind). Delivec skrbi, da se plošček po vsaki igri premakne k naslednjemu igralcu, kar zagotavlja enakomerno porazdelitev igralcev na vseh igralnih pozicijah.

Igralci so na vrsti za igro v točno določenem zaporedju v odvisnosti od pozicije ploščka in faze igre. Zaporedje spremlja delivec in igralce po potrebi opominja ob morebitnem prehitevanju igralcev pred seboj.

Ko igralec noče več sodelovati v trenutni igri, odvrže svoje karte na mizo, od koder jih kasneje pospravi delivec. Hkrati mora spremljati, kateri igralci so že izstopili iz igre, saj jih pri nadaljnjem poteku igre preskoči v vrsti.

V posameznih fazah igre delivec obrne na sredino mize skupne karte, s katerimi igralci sestavijo svojo najboljšo kombinacijo. Za igralce je to ključen podatek, saj se na podlagi kart odločajo o naslednjih potezah, opazovalci pa na podlagi skupnih kart in stav posameznih igralcev sklepajo o poteku igre.

Ob koncu posamezne igre še sodelujoči igralci pokažejo svoje karte, s tem da jih obrnejo na mizo. Delivec preveri, kateri igralec ima zmagovalno kombinacijo, nato pa igralec lahko pospravi žetone iz mize na svoj kupček. Karte, ki so pokazane na mizi zaključujejo potek posamezne igre. Igralci pri tem dobijo informacije, s katerimi kartami so igrali soigralci, delivec in organizatorji preverijo pravilnost izpeljane igre, gledalcem pa to predstavlja vrhunec igre.

3.1.2 DELO ORGANIZATORJEV

Organizatorji morajo imeti med turnirjem pregled nad celotnim dogajanjem. Skrbijo za nemoten potek turnirja, zato so njihove naloge merjenje časa, razporeditev delivcev za mize, nadzor, predsedanje igralcev, informiranje igralcev o poteku turnirja. Za opravljanje vseh nalog je bistveno, da vsi izvajalci delujejo med seboj usklajeno.

Na začetku turnirja začnejo z merjenjem časa, ki je namenjen zviševanju slepih stav. Stave se višajo po vnaprej dogovorjeni lestvici. Igralci in delivci morajo biti obveščeni o trenutnem času in višini stave, saj to vpliva na igro.

Dogajanje dodatno spremljajo nadzorniki, ki prvenstveno skrbijo za pravilno delo delivcev. Predstavljajo pomemben člen v organizaciji, saj imajo pregled nad več mizami, podatke pa dodatno dobijo od posameznega delivca. Skrbijo za pravilno predsedanje igralcev med mizami, tako da igra poteka na vedno manj mizah.

3.2 PRIKAZ PODATKOV NA ZASLONU

Zaradi množice podatkov o dogajanju na posamezni mizi in načinov prenosa ter pridobivanja, jih organizatorji ne morejo nemudoma predstaviti gledalcem.

Zato na zaslonih namenjenih za informiranje prikazujejo le podatke, ki jih sami nadzirajo, to sta čas in višina stav (slika 6). Občasno jih uporabijo za naznanitev premora, šele po številu žetonov vseh igralcev pa lahko objavijo seštevke žetonov vseh igralcev in uvrstitev na skupni lestvici. Prikazane informacije so zanimive le za neposredne udeležence igre.



Slika 6: Prikaz informacij na turnirju [12].

Želimo si prikaz informacij na zaslonih, ki bi bile zanimive tudi za gledalce. Želimo jim dinamično prikazati dogajanje na turnirju z vsemi pomembnimi informacijami, zato je potrebno pohitriti in avtomatizirati zajem vseh podatkov, jih obdelati in predstaviti na zaslonih. Takšen zaslon organizatorji bolj učinkovito uporabljajo v promocijske namene, saj jih spremlja večji krog obiskovalcev.

4 IZVEDBA MIZE RFID

Vse predmete, ki jih igralci uporabljajo med igro, označimo z RFID značko. Na mizo namestimo antene, ki so razporejene na način, da omogočajo zaznavo poteka igre. Priključene so na čitalnik, s katerim komunicira aplikacija, ki nadzira igro in pripravi podatke, ki so pomembni za prikaz na zaslonu.

Značke so nameščene na ploščku, vseh žetonih, kartah in identifikacijskih karticah igralcev. Pri načrtovanju sistema moramo upoštevati, da je v danem trenutku na mizi zelo veliko število značk, ki jih želimo prepoznati, zato mora oprema omogočati hitro in zanesljivo delovanje.

Potek igre lahko prepoznamo z analizo premikanja značk po igralni površini. Vsak igralec ima pred seboj identifikacijsko kartico, žetone in karte. Ob stavi položijo žetone na sredino mize, kamor tudi odvržejo karte ob izstopu iz igre. Delivec ima na svojem mestu igralne karte. Prav tako v vsaki igri premakne plošček k naslednjemu igralcu. Iz zgoraj opisanega stanja na mizi ugotovimo, da potrebujemo antene na več ključnih lokacijah na mizi. Antena mora biti nameščena vsaj na vsakem igralnem mestu vključno z mestom delivca in na sredini mize. Poleg podatkov, ki jih lahko pridobimo z značkami, potrebujemo tudi dodaten uporabniški vhod za vnos podatkov o dogodkih, ko se ne zgodi noben premik značke, na primer ko igralec zaključi svojo potezo. V ta namen potrebujemo dodatno stikalo, ki ga upravlja delivec.

Čitalnik mora zato poleg priklopa anten omogočati tudi priklop zunanjih vhodov, saj se želimo izogniti dodatni opremi na mizi. Za povezljivost z aplikacijo uporabimo LAN (ang. *Local Area Network*) priključek za povezavo v lokalno omrežje.

Aplikacija za delo s čitalniki je ključnega pomena za delovanje celotnega sistema. Vzpostaviti mora povezavo s posameznim čitalnikom, kar ji omogoči sprejemanje podatkov o značkah. Za vsako mora določiti kaj predstavlja in spremljati njeno premikanje med posameznimi antenami. Z analizo premikov, uporabo pravil pokra in vhodnih podatkov delivca logično ureja informacije za prikaz na zaslonih, s katerimi redno osvežuje bazo izhodnih podatkov.

Na zaslonu grafično predstavimo pridobljene informacije. Možen je prikaz posamezne mize in poteka igre na njej, časa, trenutnih stav in lestvice igralcev.

4.1 UPORABLJENA STROJNA OPREMA

Pri izdelavi mize RFID je bila uporabljena strojna oprema, katere proizvajalec je Magellan Technology iz Avstralije. Lastnik ekskluzivnih pravic za uporabo opreme v igralniške namene pa je podjetje International Gaming Technologies iz Združenih držav Amerike [11].

4.1.1 ČITALNIK

RFID čitalnik MARS-24 (slika 7) je namenjen branju in pisanju na individualne značke tipa *ItemTag* in množico značk *StackTag*. Uporablja komunikacijski protokol skladen s standardom ISO/IEC 18000 – 3 Mode 2 in deluje z v območju visokih frekvenc (HF 13,56 MHz) [6].



Slika 7: Čitalnik MARS-24.

Lastnosti, pomembne za izvedbo naloge, so naslednje:

- Priklop do 24 neodvisnih anten
- Priklop uporabniškega vhoda in izhoda
- Povezljivost z uporabo USB ali LAN
- Prepoznavnost do 15000 značk v minuti
- Zmožnost takojšnje detekcije premika značke z območja ene antene v območje druge
- Zmožnost detekcije zloženih in prekrivajočih značk
- Zmogljiva platforma za samodejno delovanje

Večina čitalnikov omogoča priklop manjšega števila anten, pogosto ene same. Zaradi potrebe po uporabi več območij na mizi, kjer kontroliramo značke, je nujno potreben čitalnik z možnostjo priklopa več anten. Tako zagotovimo kompaktnost izdelka in enostavnejšo implementacijo.

Za čim večjo odzivnost sistema sta pomembna podatka o prepoznavi značk in detekciji prehoda značke iz ene antene na drugo. Prvi podatek nam pove, koliko značk in njihovih 96-bitnih identifikacijskih števil lahko čitalnik zazna, ko se pojavijo v območju dosega ene od anten. Višje število pomeni hitrejšo inicializacijo aplikacije do faze, ko je pripravljena spremljati igro. V tej fazi je za aplikacijo pomemben drugi podatek, saj pogojuje odzivnost aplikacije na dogodke na mizi.

Vhodno/izhodni vmesnik potrebujemo za priklop stikala. Omogoča priklop petih izhodov in sedmih vhodov, podatke o trenutnem stanju vhodov pa čitalnik lahko posreduje aplikaciji.

Komunikacijo z zunanjimi napravami omogočata priključka RJ 45 (*Ethernet*) za povezavo v internetno omrežje in USB (Universal Serial Bus)

4.1.2 ANTENE

Magellan Technology proizvaja široko paleto naprednih anten (slika 8). Izbor različnih dimenzij in načinov delovanja omogoča optimalno izbiro za uporabo v posamezni aplikaciji.



Slika 8: Antena za uporabo na čitalniku MARS-24.

Antene se glede na način izvedbe v osnovi ločijo v naslednji skupini:

- Antene z eno osjo (ang. *Single axis antenna*)
- Antene z dvema osema (ang. *Dual axis antenna*)

Antena z eno osjo je sestavljena iz trde podlage, na kateri je navitje, elektronski elementi in potenciometer za fino nastavljanje antene. Za priklop uporablja RCA (ang. *Radio Corporation of America*) kabel.

Antena z dvema osema se razlikuje od prve v navitju antene, ki je v tem primeru dvojno. Medtem ko prva omogoča le branje značk, ki so postavljene vzporedno z anteno, lahko ta prebere tudi značke, ki so postavljene navpično na anteno. Zaradi dveh navitij je potrebna uporaba dveh priklopnih kablov.

Pri izdelavi smo uporabili naslednje antene:

- 3x antena z eno osjo in dimenzijami 120x240 mm
- 1x antena z dvema osema in dimenzijami 228x450 mm
- 1x antena z dvema osema in dimenzijami 228x540 mm

4.1.3 ZNAČKE

Značke so vgrajene v žetonih, identifikacijskih karticah, kartah in ploščku. Čipe zanje proizvaja Magellan Technology, vgradnjo v igralne pripomočke pa opravijo drugi proizvajalci ob izdelavi le-teh. Glavne lastnosti značk so naslednje:

- Unikatna serijska številka
- Do 8 Kbit uporabniškega spomina
- Hitrost pisanja do 424 Kbit/s
- Neodstranljiva zaščita pisanja
- Velika odpornost proti zunanjim vplivom

Unikatna serijska številka zagotavlja visok nivo varnosti pri uporabi značk za identifikacijo predmetov z veliko vrednostjo. Velik uporabniški spomin in veliko hitrost prenosa podatkov lahko uporabimo za shranjevanje informacij o predmetu. Po zapisu podatkov na značko lahko uporabniški spomin zaklenemo in tako zagotovimo nespremenljivost ob nadaljnji uporabi. Pravilna vgradnja značke v nosilni medij prepreči morebitne poškodbe ob uporabi.

4.2 IZDELAVA MIZE

Mizo smo načrtovali in izdelali z uporabo razpoložljive strojne opreme. Ker smo imeli na voljo le pet anten, smo lahko izdelali mizo s tremi igralnimi mesti, četrto anteno smo uporabili za mesto delivca, peto pa za preverjanje položenih stav. Antene je potrebno namestiti na ključna mesta na mizi, zato smo po predhodnem načrtovanju izdelali okvir, v katerega smo jih vgradili. Povezali smo jih na čitalnik, ki je nameščen ob mizi. Nanj smo priklopili tudi uporabniški vmesnik, ki ga uporablja delivec, in omrežni kabel za komunikacijo čitalnika z ostalimi napravami. S pomočjo računalnika in programske opreme proizvajalca smo čitalnik nastavili za delo z antenami na mizi in značkami v igralnih pripomočkih. Na koncu je bilo potrebno še programiranje vseh značk s podatki, ki jih potrebuje aplikacija za svoje delovanje.

4.2.1 RAZPOREDITEV ANTEN

Za uspešno izdelavo sistema smo morali za mizo RFID najprej izdelati ustrezne načrte. Naloga mize je zaznava vseh podatkov o igri, ki jih nato posreduje aplikaciji. Zato je treba vanjo namestiti antene, s pomočjo katerih lahko določimo lokacijo igralnih pripomočkov. Za pravilno postavitve anten smo morali preučiti njihove premike po igralni površini. Pri tem smo ugotovili, da ni mogoče zaznati vseh parametrov igre le z značkami na igralni površini, zato je bilo potrebno načrtovati dodatni vmesnik, s pomočjo katerega lahko aplikaciji posredujemo manjkajoče podatke. V nadaljevanju bomo najprej predstavili potek in pravila igre pokra pri navadni igri. Pri tem bomo določili vrsto podatkov, ki jih želimo posredovati aplikaciji in način pridobivanja le-teh.

Igra pokra pozna več različic igre. Vsako določajo nekoliko različna pravila v poteku igre, številu uporabljenih kart in načina deljenja med igralce ter stavljenja žetonov. Pri prikazu rešitve smo se zaradi enostavnosti omejili na trenutno najbolj razširjeno različico igre *Texas Holdem*, vendar bi s podobnim pristopom lahko izdelali rešitev za vsako posamezno igro.

Za igro *Texas Holdem* je značilno, da se jo igra z 52 igralnimi kartami. Delivec jih na začetku igre premeša in vsakemu igralcu razdeli po eno karto. Igralec z najvišjo karto dobi predse plošček. Plošček se med posameznimi igrami premika med igralci v smeri urinega kazalca in določa vrstni red igralcev med igro. Navadno se igra z uporabo slepih stav, ki jih morata pred pričetkom igre postaviti igralca, ki sedita na levi strani ploščka. Prvi položi malo slepo stavo (ang. small blind), drugi pa veliko slepo stavo. Velika slepa stava (ang. big blind) je

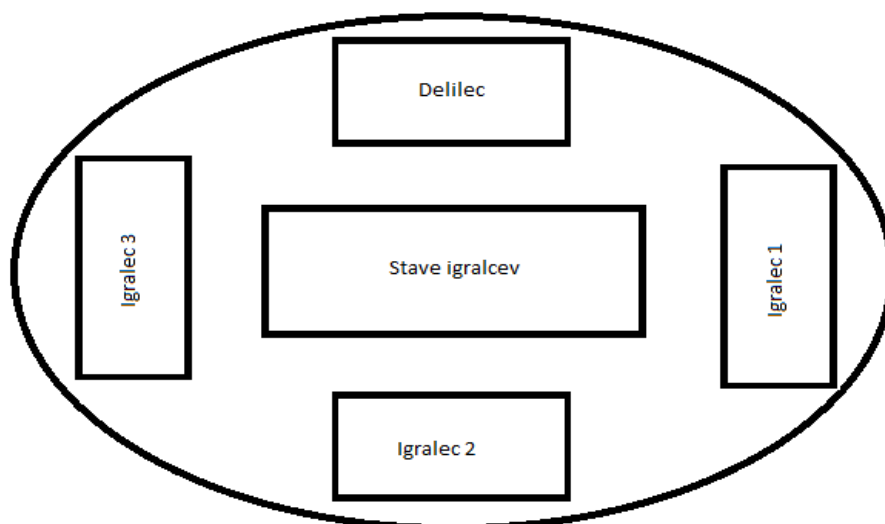
najmanjša vrednost, ki jo lahko igralci stavijo med igro, mala slepa stava pa navadno predstavlja polovico te vrednosti. Pri turnirski igri se stave s časom zvišujejo.

Ko je določen plošček in so slepe stave plačane, delivec premeša karte in jih razdeli med igralce. Vsakemu igralcu razdeli po eno karto naenkrat v dveh krogih, tako da vsak dobi dve karti. Prvo dobi igralec, ki je položil malo slepo stavo. Ko so karte razdeljene, igralci začnejo z igro, ki poteka v petih fazah. V prvi fazi igre (ang. *preflop*) se igralci lahko odločijo, če bodo sodelovali v igri. Prvi na potezi je igralec levo od igralca, ki je položil veliko slepo stavo. Za sodelovanje v igri mora vsaj izenačiti minimalno stavo, tako da položi zahtevano število žetonov na mizo. V kolikor noče sodelovati, lahko svoje karte odvrže na mizo in odstopi. V primeru da hoče igralec zvišati stavo, predse položi večjo vsoto žetonov. Naslednji igralci igrajo podobno, vendar zanje velja, da morajo v primeru višanja stave prejšnjega igralca za sodelovanje v igri staviti vsaj enako število žetonov. Ko so vse stave med sodelujočimi igralci izenačene, se lahko začne druga faza igre (ang. *flop*). Delivec na sredino mize obrne tri skupne karte. Igro začne igralec na levi strani ploščka. Odloči se lahko za stavo ali pa preveri (ang. *check*), nato je na vrsti naslednji igralec. V primeru stave jo lahko viša ali pa odstopi. Tako se igra nadaljuje dokler niso vse stave izenačene in delivec lahko obrne naslednjo skupno karto. Tretja (ang. *turn*) in četrta faza igre (ang. *river*) se odigrata na enak način kot druga. Ko so stave izenačene, igralci v peti fazi igre (ang. *showdown*) pokažejo svoji karti in jih primerjajo z nasprotniki, zmagovalec pa pobere vse žetone. Ko delivec preveri, da ima pri sebi vse karte in so vsi igralci pospravili žetone premakne plošček k naslednjemu igralcu in začne se nova igra.

Na podlagi poteka igre določimo razporeditev anten na mizi (slika 9). Za preverjanje žetonov, ki jih imajo igralci pred seboj, potrebujemo za vsakega svojo anteno. To anteno uporabimo tudi za identifikacijo igralcev, prepoznavanje njihovih kart in določanje pozicije ploščka.

Za preverjanje stav, ki jih igralci polagajo med igro, potrebujemo anteno na sredini mize. To anteno uporabimo tudi za prepoznavo skupnih kart, ki jih delivec obrača na mizo.

Za delivca potrebujemo posebno anteno. Po vsaki igri želimo preveriti, če so na njej vse karte, ki jih uporabljamo v igri. Tako zagotovimo sprotno preverjanje, kar onemogoča igranje z manjkajočimi kartami.



Slika 9: Razporeditev anten na različnih igralnih mestih.

4.2.2 IZDELAVA OKVIRJA

Namen okvirja je zagotovitev igralne površine za igralce, pod katero so varno nameščene antene za spremljanje njihove igre. Okvir izdelamo iz neprevodnega materiala, saj se želimo izogniti motnjam, ki bi jih lahko povzročili prevodni materiali v dosegu antene. Zato je najbolj primerna izdelava lesenega okvirja, saj ga lahko enostavno obdelamo in prilagodimo za uporabo pri igri pokra.

Za izdelavo uporabimo vezano ploščo, v katero na spodnji strani izrežemo utore za antene, na zgornjo stran pa namestimo igralno prevleko. Po površini razporedimo štiri antene za igralce in njihove stave, igralno površino za delivca pa naredimo ločeno iz lesenega u-profila. Ker mora biti območje vsakega igralca ločeno od sosednjega, morajo biti antene med seboj dovolj oddaljene. Za zaščito anten in priključkov za kable po spodnjem robu namestimo obrobo, na katero lahko postavimo pokrov. Tako je miza na zgornji strani pripravljena za igro, s spodnje strani pa zaščitena proti poškodbam.

4.2.3 IZDELAVA VMESNIKA ZA DELIVCA

Ker z analizo premikov igralnih pripomočkov ne moremo določiti vseh parametrov igre, moramo izdelati vmesnik, s katerim bo upravljal delivec in aplikaciji posredoval manjkajoče podatke. Pomembni so podatki ob naslednjih treh dogodkih:

- Ko igralec preveri

- Ko igralec odstopi
- Ko se zaključi igra in se na zaslonu pokažejo karte

Podatke potrebujemo za pravilen pregled nad potekom igre. Spremlja jih delivec za mizo, ki skrbi da igralci odigrajo igro v pravilnem vrstnem redu. Ko posamezen igralec odigra, se igra premakne na naslednjega igralca. V tem primeru je potrebno aplikaciji sporočiti, da je na vrsti naslednji igralec. Če igralec odstopi, ne sodeluje več v trenutni igri, zato ga do naslednje igre vedno preskočimo. To stori s položitvijo kart na sredino mize, kar lahko zaznamo v primeru, ko je tudi v karte vgrajen RFID čip, v nasprotnem primeru pa mora delivec tudi ta podatek posredovati aplikaciji. Ko se vse stave zaključene, mora delivec potrditi zaključek igre in na zaslonu se lahko prikažejo karte igralcev, ki so igrali do konca.

Čitalnik je opremljen s priključkom za zunanji vmesnik, ki zagotavlja več različnih vhodnih in izhodnih signalov. Pomen posameznih nožic na priključku je predstavljen v tabeli 1.

Številka nožice	Smer	Funkcija
1	-	Napetost 5V
2	Izhod	Signal alarma o napaki
3	Izhod	Oddajna serijska vrata za razhroščevanje
4	Izhod	Izhodna SPI vrata
5	Izhod	Signal ure SPI vrat
6	Izhod	Izhodna linija 0
7	Izhod	Izhodna linija 1
8	Izhod	Izhodna linija 2
9	Izhod	Izhodna linija 3
10	Izhod	Izhodna linija 4
11	Vhod	Sprejemna serijska vrata za razhroščevanje
12	Vhod	Vhodna linija 0
13	Vhod	Vhodna linija 1
14	Vhod	Vhodna linija 2
15	Vhod	Vhodna linija 3
16	Vhod	Vhodna linija 4
17	Vhod	Vhodna linija 5
18	Vhod	Vhodna linija 6
19	-	Ozemljitev
20	-	Ozemljitev

Tabela 1: Pomen nožic na priključku za uporabniški vhod in izhod.

Zanimajo nas predvsem priključki za zunanji vhod in izhod. Vhodne povezave nam omogočajo priklop tipk, izhode pa priklop LED diod. Oboje lahko krmilimo z aplikacijo, kjer lahko zaznamo pritisk tipke in določimo katera dioda sije v danem trenutku. Tako lahko

delivec aplikaciji posreduje podatke s pritiskom na tipko, aplikacija mu pa posreduje informacijo o igralcu, ki je trenutno na vrsti z vklopom posamezne LED diode.

Vmesnik naredimo z uporabo plastičnega ohišja, v katerega vgradimo po tri tipke in tri LED diode. S ploščatim kablom jih povežemo na priključke 6, 7, 8 in 12, 13 in 14. Vmesnik nato namestimo na okvir mize (slika 10).



Slika 10: Uporabniški vmesnik.

4.2.4 NAMESTITEV ANTEN IN ČITALNIKA

Antene pritrdimo v pripravljene utore na spodnji površini mize z uporabo obojestranskega lepilnega traku (slika 11). Nanje priključimo kable s priključkom RCA in jih skupaj s ploščatim kablom uporabniškega vmesnika pritrdimo na robu okvirja. Ko zapremo pokrov, je miza pripravljena za povezavo s čitalnikom. Po navodilih proizvajalca mora biti razdalja med antenami in čitalnikom ob delovanju minimalno 20 cm, zato izkoristimo vso dolžino kablov in ga namestimo čim dlje od mize. Kable vseh anten povežemo na ženske RCA priključke, ploščati kabel pa povežemo na priključek za uporabniški vhod in izhod. Za povezavo z računalnikom priključimo še LAN kabel. Na koncu priključimo še napajalni kabel in po vklopu v električno omrežje je čitalnik pripravljen za uporabo.



Slika 11: Okvir mize RFID z antenami.

4.2.5 KONFIGURACIJA ČITALNIKA

Pred uporabo RFID naprav v aplikaciji, jih je potrebno primerno nastaviti. Potrebno je vzpostaviti povezavo z računalnikom, nastaviti čitalnik za pravilno delo z različnimi antenami, vsako posamično nastaviti za optimalno delovanje, preveriti delovanje vseh RFID značk in nanje zapisati podatke, ki jih v nadaljevanju potrebujemo v aplikaciji. Pri opravi vseh nalog smo si pomagali s programom proizvajalca za delo s čitalniki *Reader Manager*.

Za namestitev potrebujemo računalnik z naslednjimi karakteristikami: operacijski sistem Microsoft Windows XP z drugim servisnim paketom, 500 MHz procesorske moči, 128 MB pomnilnika in omrežna kartica. Po uspešni namestitvi iz zgoščenke, ki je priložena čitalniku, preverimo še omrežne nastavitve. Privzeta nastavitve čitalnika za povezovanje v omrežje je nastavljena tako, da najprej poizkusi vzpostaviti povezavo z strežnikom DHCP (ang. *Dynamic Host Configuration Protocol server*). Zato ob prvi povezavi priključimo čitalnik in računalnik v omrežje z nastavljenim strežnikom DHCP. Po prvem zagonu programa izberemo zavihek *Connections*, kjer upravljamo povezavo s čitalnikom. Za program velja, da je v enem trenutku lahko povezan le z enim čitalnikom. Zato iz ponujenih povezav izberemo čitalnik, ki ga je program našel v našem omrežju. Po vzpostavitvi povezave se nam v statusni vrstici izpiše IP naslov in tip značk, ki so uporabljene v trenutni konfiguraciji. Za lažje nadaljnjo delo čitalniku določimo stalni naslov IP v oknu za omrežne nastavitve. Za uveljavitev sprememb čitalnik ponovno zaženemo.

Na čitalnik smo priklopili antene, ki se med seboj razlikujejo v dimenzijah in delovanju, zato je potrebno čitalnik nastaviti za pravilno delo z njimi. V zavihku *Tools* → *System* zaženemo *Reader Setup* in zažene se nam čarovnik za nastavitve skupin anten. V prvem koraku ustvarimo skupine posameznih osi. Vsaka os predstavlja svoje navitje na anteni za uporabo v aplikaciji pa želimo dobiti podatke iz posamezne antene kot celote. Zato antene razdelimo v štiri skupine:

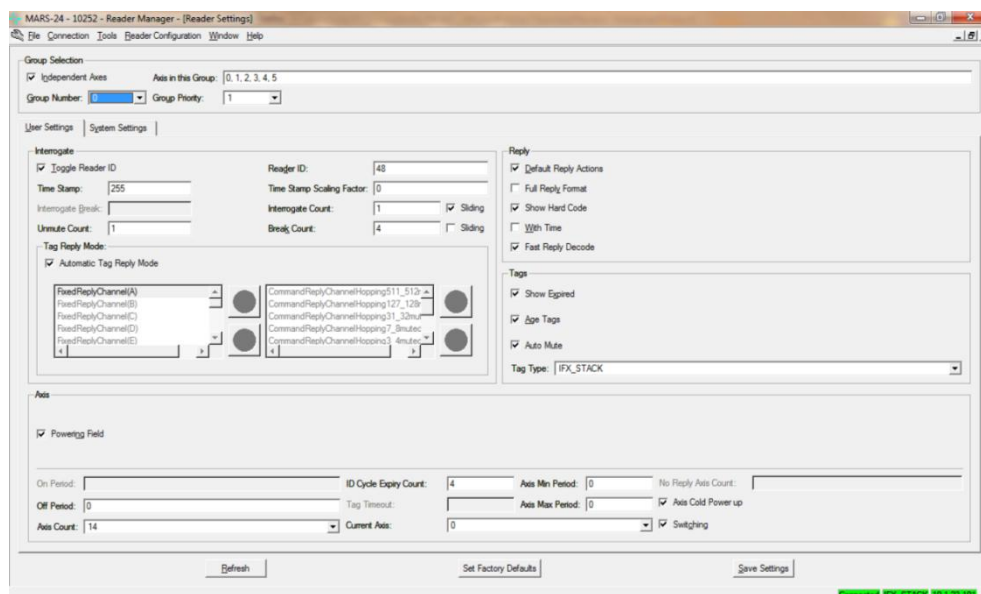
- Antene z eno osjo, ki jih uporabimo za dva igralca in delivca
- Antena z dvema osema 228x450 mm, ki jo uporabimo za tretjega igralca
- Antena z dvema osema 228x540 mm, ki jo uporabimo za stave na mizi
- Onemogočene

V prvo skupino uvrstimo prvih šest osi, kjer bomo v nadaljevanju uporabili manjše antene z eno osjo. V drugo in tretjo skupino dodamo po dve osi, ki jih bomo uporabili za dve večji anteni, v tretjo skupino pa dodamo ostale osi, ki jih nato onemogočimo.

V naslednjem koraku določimo način delovanja posamezne skupine. Za vsako določimo ali so osi neodvisne, če spreminjamo identifikacijo čitalnika in prioriteto delovanja. Na koncu moramo še za vsako os določiti tip antene, ki je priklopljena na čitalnik.

Dodatne nastavitve uredimo v oknu *Reader Settings* (slika 12). V njem lahko bolj natančno nastavimo uporabniške in sistemske nastavitve za vsako posamezno skupino. Nastaviti moramo vrsto značk, ki jih uporabljamo in način delovanja s čitalnikom. Izbira primernih nastavitvev je zelo pomembna, saj morajo biti usklajene z delovanjem aplikacije. V tem primeru najbolj izkoristimo prednosti strojne opreme. Za aplikacijo je pomembno, da stalno vzdržuje bazo s podatki na kateri anteni se nahaja posamezna značka, kar lahko izvedemo na dva načina. V prvem primeru čitalnik periodično pošilja poizvedbe o značkah na posamezni anteni in jih shranjuje. Zaradi velikega števila značk podatkov, ki jih želimo prebrati, je takšen način lahko počasen. V drugem primeru lahko podatke o značkah shranimo ob prvem zasliševanju, v nadaljevanju pa podatke z značke ponovno preberemo le ob premiku na drugo anteno. Pri tem izkoristimo način zasliševanja značk, ki nam ga omogoča uporaba čitalnika MARS-24. Ko se na anteni pojavijo nove značke, čitalnik z determinističnim načinom najde vse značke, ki so v dosegu in jih shrani. Šele ko so prepoznane vse značke, lahko vsaki posamezno pošlje ukaz za branje podatkov na njej. Ker je dovolj, da podatke preberemo enkrat, lahko značko v nadaljevanju utišamo (ang. *mute tag*), dokler se ne pojavi v dosegu druge antene. Takšen način je dosti hitrejši, obenem pa tudi zmanjša prenos podatkov med

čitalnikom in aplikacijo. V ta namen v nastavitvah dela z značkami označimo polje za samodejno utišanje značke. Vse ostale nastavitve lahko popolnoma nadzorujemo s pomočjo ukazov v kodi aplikacije in jih bomo uporabili med njenim izvajanjem.



Slika 12: Uporabniški vmesnik Reader Settings.

Pred prvo uporabo značk na antenah je potrebno vsako posamezno nastaviti za čim boljše delovanje v okolju, kjer je nameščena. Za ta namen uporabimo opremo za vzdrževanje anten, ki je sestavljena iz naprave za merjenje napetosti na anteni (slika 13), kabla za povezavo in merilca napetosti. Uporabljamo jo tako, da napravo položimo na anteno in s pomočjo merilnika odčitamo napetost, ki se pojavi na njej. Za boljše delovanje in večji doseg antene mora biti le-ta čim večja, kar dosežemo z nastavitvijo potenciometrov nameščenih na anteni. Med odčitavanjem napetosti na anteni ga s primernim izvijačem obrnemo do točke, ko je izmerjena napetost največja. Ko nastavimo vse antene, imamo mizo in čitalnik pripravljeno za delo z značkami.



Slika 13: Oprema za vzdrževanje anten.

Za delo z značkami uporabimo orodje *Tag Memory Map*, ki ga najdemo v zavihku *Tools* → *System*. Ko se nam odpre okno, je na levi strani prazen seznam značk, na desni pa tabela podatkov, ki so zapisani na posamezni znački. Ko značko premaknemo v doseg antene, se v seznamu prikaže njena unikatna oznaka, ki je najprej obarvana sivo. Po nekaj trenutkih čitalnik prebere podatke na znački in jih izpiše v tabeli na desni strani, oznaka pa se obarva v zeleno barvo, kar pomeni, da je pripravljena za delo.

Tabelo podatkov, ki so zapisani na znački, podrobno analiziramo, saj nam mora biti znana ob izdelavi aplikacije. Celoten pomnilnik je razdeljen na dva dela – sistemski in uporabniški. Sistemski del pomnilnika obsega prvih deset blokov in vsebuje osnovne podatke o znački. Pomen zapisa v posameznem bloku je naslednji:

- Blok 0: Rezervirano za proizvajalca
- Blok 1: Koda izdelave
- Blok 2: Določeni identifikator 1
- Blok 3: Določeni identifikator 2
- Blok 4: Identifikator skupine aplikacij
- Blok 5: Pogojni identifikator
- Blok 6: Nastavitvena beseda
- Blok 7: Geslo 0
- Blok 8: Geslo 1
- Blok 9: Geslo 2

Podatke lahko preberemo, spreminjamo pa lahko samo določene. Za prve štiri bloke je proizvajalec omogočil zaščito proti spreminjanju vsebine, tako da lahko spreminjamo le

ostale. Ko spremenimo vsebino v teh blokih, lahko tudi sami vzpostavimo zaščito pred spreminjanjem, geslo pa lahko uporabimo za zaščito blokov uporabniškega pomnilnika. Zaščite ni mogoče odstraniti in ker značk ne bomo uporabljali v razmerah, kjer bi bila zaščita nujna, te možnosti ne bomo uporabili.

Uporabniški del pomnilnika se nahaja od 11 do zadnjega razpoložljivega bloka. Namenjen je shranjevanju podatkov, ki jih aplikacija potrebuje pri svojem delu. Posamezne značke se med seboj razlikujejo po velikosti pomnilnika, ki je na voljo. Značke, ki jih imamo na voljo, vsebujejo pomnilnik velikosti 8 Kbit pri katerem so v blokih od 11 do 16 že zapisani podatki za uporabo v drugi aplikaciji. Ker nočemo po nepotrebnem uničiti podatkov, se odločimo, da bomo za potrebe naše aplikacije uporabili bloke na naslovih 32, 34, 35 in 36. V resnici ne potrebujemo toliko prostora, vendar se branje pomnilnika izvaja v blokih po štiri. Tako je to najmanjša in hkrati največja enota pomnilnika, s katerim lahko operiramo pri enkratnem dostopu. Vanje lahko shranimo 16 šestnajstiških znakov, kar nam omogoča vpis podatkov, ki nam bodo omogočili razlikovanje med posameznimi predmeti uporabljenimi v igri, v primeru vrednostnih žetonov pa tudi njihovo vrednost. Zato določimo tabelo (tabela 2), ki predstavlja preslikavo med predmetom, ki je uporabljen v igri, in podatki, ki so shranjeni na njegovi znački.

Igralni pripomoček	Podatki	Skupno število
Plošček	0000 0000 0000 0000	1
Karta	0000 0000 0000 0000	52
Žeton - 5000	5000 9999 ffff ffff	100
Žeton - 1000	1000 9999 ffff ffff	100
Žeton - 500	0500 9999 ffff ffff	100
Žeton - 100	0100 9999 ffff ffff	50
Žeton - 25	0025 9999 ffff ffff	50
Žeton - 5	0005 9999 ffff ffff	50
Identifikacijska karta	0000 0002 ffff ffff	20

Tabela 2: Podatki na različnih predmetih uporabljenih pri igri.

Iz tabele je razvidno, da je skupno število značk 523, kar pomeni da bi bilo zapisovanje na vsako posamezno značko z uporabo programa dolgotrajno opravilo. Zato na anteno položimo vse značke določenega tipa in odpremo konzolno okno, ki ga dobimo v zavihku *Tools* → *System*.

S pomočjo ukaza `Command((writeAddress=<int>,writeLength=<int>,data=<data>)` zapišemo podatke na vse značke v doseg anten. Ko se prepričamo, da so podatki zapisani na

vseh značkah, jih odstranimo in v ukazno vrstico napišemo ukaz za zapis podatkov na naslednji tip značk. Postopek ponavljamo dokler niso vse značke označene in pripravljene za uporabo v programu.

4.3 IZDELAVA PROGRAMA

Program v sistemu skrbi za komunikacijo s čitalnikom, pri kateri pridobi podatke o značkah v doseg in jih po pravilni obdelavi izvozi v podatkovno bazo, iz katere lahko orodje za digitalno oglaševanje na zaslonu prikaže potek igre.

4.3.1 OPIS

Pred začetkom izdelave programa smo se morali seznaniti z delovanjem čitalnika RFID in podrobno preučiti postopke za vzpostavitev povezave in delo z značkami ter uporabniškim vhodom in izhodom. Nato smo pripravili potrebne ukaze za delo s čitalnikom in definirali strukturo podatkov, ki se prenašajo med njim in programom. Iz podatkov lahko v programu razberemo, na kateri anteni se je pojavila določena značka in del njenega uporabniškega pomnilnika. Obe informaciji moramo v aplikaciji obdelati na primeren način, ki nam omogoča reprodukcijo trenutnega dogajanja na mizi. Pri obdelavi moramo zato upoštevati pravila igre, razbrati vrsto značke in smer njenega premika. Z razlikovanjem značk po predmetu, ki ga predstavljajo, ob posameznem premiku določimo potezo, ki jo je odigral igralec, kar nam omogoča spremljanje celotne igre. Ko vzpostavimo pregled nad igro, lahko pripravimo podatke, ki jih želimo prikazati na zaslonu in jih v pripravljeni strukturi shranimo v izhodno datoteko. Na podlagi opisanih postopkov lahko zapišemo zahteve, ki jih mora program izpolnjevati:

- Vzpostavitev povezave s čitalnikom
- Možnost pošiljanja ukazov čitalniku
- Sprejemanje sporočil s čitalnika in razčlenitev vsebine
- Vpeljava logike za igro pokra
- Orodje za delo z uporabniškim vhodom in izhodom na čitalniku
- Vodenje celotnega dogajanja na mizi
- Osveževanje izhodne datoteke v najkrajšem možnem času

Program izdelamo z upoštevanjem naslednjih načel načrtovanja:

- Enostavnost

Program mora biti kolikor je mogoče enostaven za uporabo, vzdrževanje in morebitno nadgrajevanje funkcij.

- Povezljivost

Omogočati mora povezljivost z vsemi napravami v sistemu.

- Prenosljivost

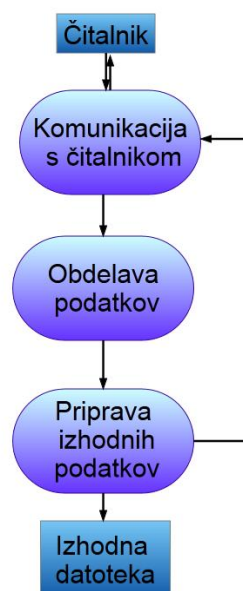
Program mora biti prenosljiv med različnimi operacijskimi sistemi.

4.3.2 SHEMA DELOVANJA

Z diagramom delovanja predstavimo osnovno delovanje programa (slika 14), ki ga logično lahko razdelimo na tri komponente:

- Komunikacija s čitalnikom
- Obdelava podatkov
- Priprava izhodnih podatkov

Vsaka posamezna komponenta za svoje delovanje potrebuje ostali dve, zato jih je potrebno uskladiti. V ta namen najprej določimo vrsto izhodnih podatkov, nato pa določimo postopke s pomočjo katerih lahko vhodne podatke pretvorimo v zahtevan izhod. Izhodni podatki poleg datoteke za prikaz na zaslonu vsebujejo tudi ukaze za čitalnik, s pomočjo katerih krmilimo uporabniški izhod.



Slika 14: Shema delovanja programa.

4.3.3 DOLOČITEV IZHODNIH PODATKOV

Izdelave programa se lotimo šele, ko določimo strukturo podatkov, ki jih želimo pridobiti. Ker hočemo prikazovati trenutno stanje na mizi, potrebujemo popolne podatke o prisotnih igralcih, premikih in trenutni poziciji igralnih pripomočkov ter poteku igre. Na mizi so štiri antene, ki predstavljajo štiri območja igre, zato za vsako od njih določimo vrsto podatkov, ki so predstavljeni v tabeli 3.

Vrsta podatka	Igralec 1	Igralec 2	Igralec 3	Miza	Tip podatka
Igralec aktiven					Slika
Igralec na vrsti					Slika
Delivec					Slika
Mali slepec					Slika
Veliki slepec					Slika
Seštevek stav					Število
Trenutna stava					Število
Seštevek vrednosti žetonov					Število
Število žetonov vrednosti 5000					Število
Število žetonov vrednosti 1000					Število
Število žetonov vrednosti 500					Število
Število žetonov vrednosti 100					Število
Število žetonov vrednosti 25					Število
Število žetonov vrednosti 5					Število
Skupne karte turn					Slika
Skupne karte flop					Slika
Skupne karte river					Slika
Ime igralca					Besedilo
Slika igralca					Slika
Karte igralca					Slika
Karte igralca					Slika
Informacije					Besedilo

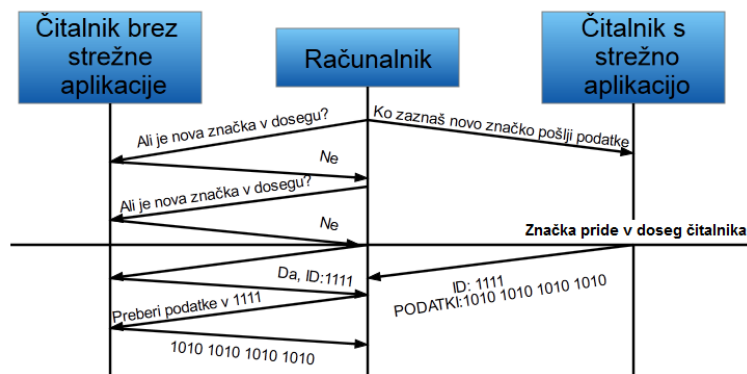
Tabela 3: Podatki v izhodni datoteki.

Iz tabele je razvidno, da kot podatkovni tip nastopajo besedilo, število in slika. Čeprav vsi podatki v datoteki nastopajo kot besedilo, je razlog razlikovanja v načinu prikaza na zaslonu. V orodju za prikaz lahko v posameznih okvirjih prikazujemo dinamične vsebine v obliki besedila ali slike. Medtem ko besedilo lahko poljubno oblikujemo ali obarvamo, lahko za sliko določimo pot do datoteke, kjer je shranjena. Tako lahko vse datoteke za delovanje zaslona shranimo na predvajalniku.

Kljub temu, da določenih polj v tabeli ne potrebujemo (obarvani s črno barvo), tudi za ostale velja, da ni nujno da so v določenem trenutku zapolnjeni. Ker vsebina odraža dinamiko dogajanja na mizi, se stalno spreminja. Tako se na primer skupne karte izpišejo šele po tem, ko so obrnjene na mizi, karte igralcev pa šele, ko je igra zaključena. Za ohranitev enostavnosti se odločimo, da podatke predstavimo v tekstovni datoteki, v kateri je vsaka vrsta podatka v novi vrstici, ločeni pa so z vejicami. Pomen vsakega posameznega polja bomo prikazali pri izdelavi programa in končni uporabi na zaslonu.

4.3.4 KOMUNIKACIJA S ČITALNIKOM

Na čitalniku je nameščena strežna aplikacija (*ReaderServer*). Nadzoruje delovanje strojne opreme, pošilja ukaze značkam, sprejema in obdeluje njihov odziv ter zagotavlja uporabniški vmesnik za zunanje aplikacije. Komunikacijo vzpostavimo z uporabo vtičnice v programu, s pomočjo katere najprej vzpostavimo povezavo strežnik-odjemalec (ang. *client-server connection*), nato pa pošiljamo ukaze in sprejemamo sporočila. Podatki se prenašajo v obliki nizov. Vsi ukazi so sintaktično napisani v programskem jeziku Python, saj čitalnik uporablja njegov prevajalnik. S pošiljanjem ukazov čitalniku upravljamo njegovo delovanje in s tem določimo odziv značk v njegovem dosegu ter delovanje uporabniškega vhoda in izhoda. Prejeti podatki so v obliki sporočil ob spremembah, za katere določimo, da nam čitalnik pošlje odziv. Pri tem se uporabljen čitalnik pomembno razlikuje od ostalih, kjer se navadno uporabljajo kot naprava, ki predstavlja komunikacijski kanal med značkami in nadzornim računalnikom. Pri takšnem načinu je za doseg istega cilja potrebno veliko več komuniciranja med napravama, kot pri uporabi vnaprej določenih ukazov (slika 15). Z zmanjšanjem prenosa podatkov tako veliko pridobimo na odzivnosti celotnega sistema.



Slika 15: Komunikacija s čitalniki.

4.3.4.1 Vzpostavitev povezave

Strežna aplikacija na čitalniku ima implementirano vtičnico, ki posluša na vratih 8023, zato definiramo razred *ReaderClient* v katerem implementiramo odjemalca, ki se poveže na ta vrata. S podatki o omrežnem naslovu čitalnika in vratih ustvarimo nov objekt tipa *Socket*. Hkrati ustvarimo še objekta tipa *BufferedReader* in *BufferedWriter*, ki nam bosta služila za branje oziroma vnos podatkov na vtičnico. Izvorna koda je predstavljena na sliki 16, razred *ReaderClient* pa uporabimo v glavnem razredu.

```
public class ReaderClient
{
    private final int READER_PORT = 8023;
    private Socket clientSocket;
    private BufferedReader input;
    private BufferedWriter output;
    public ReaderClient()
    {
    }
    public void connectToReader(String ipAddress)
    {
        clientSocket = new Socket(ipAddress, READER_PORT);
        input = new BufferedReader( new InputStreamReader(clientSocket.getInputStream()));
        output = new BufferedWriter( new OutputStreamWriter(clientSocket.getOutputStream()));
        handleConnection();
    }
    private void handleConnection()
    {
        if (input.ready())
        {
            System.out.println("Received: " + input.readLine());
        }
    }

    public void sendToReader(String command)
    {
        if (output != null && clientSocket.isConnected())
        {
            output.write(command);
            output.flush();
        }
    }
    public void closeConnection()
    {
        input.close();
        output.close();
        clientSocket.close();
    }
}
```

Slika 16: Razred *ReaderClient*.

4.3.4.2 Pošiljanje ukazov

Za pošiljanje ukazov uporabimo metodo *sendToReader*, ki ji kot argument podamo željen ukaz. Argument mora biti v obliki niza, ki na koncu vsebuje znak za zaključek vrstice. Ob

zagonu programa želimo na čitalniku pripraviti uporabniški vhod in izhod za delovanje ter določiti način izpraševanja značk. Za dosego tega cilja čitalniku pošljemo naslednje ukaze:

- `readerClient.sendToReader("MonitorInput(IO_DEFAULT) \n\r");`
- `readerClient.sendToReader("CheckInputEvery(100) \n\r");`
- `readerClient.sendToReader("InterrogateAction(readAddress=32,readLength=4) \n\r");`

S prvim ukazom čitalniku sporočimo, da hočemo stalno nadzorovati uporabniški vhod. Tako nam bo čitalnik posredoval informacijo o vsaki spremembi stanja.

Z uporabo drugega ukaza nastavimo interval preverjanja uporabniškega vhoda na 100 ms. Ker je na vhod priključena tipka, mora biti interval dovolj majhen, da zaznamo pritisk.

Tretji ukaz nastavi čitalnik na željen način izpraševanja značk. Z uporabo ukaza *InterrogateAction* čitalniku povemo, da hočemo informacije o znački in podatke od 32. do 35. bloka pomnilnika.

4.3.4.3 Sprejemanje podatkov

Izvorno kodo, ki smo jo napisali do tega trenutka, lahko preizkusimo. Program se bo povezal s čitalnikom, poslal ukaze, ki smo jih definirali v prejšnjem poglavju, in nato čakal na morebiten odziv. V metodi *handleConnection* smo definirali, da se odziv izpiše na sistemskem izhodu, zato program zaženemo v konzoli integriranega razvojnega okolja. Opravimo preizkus delovanja z različnimi igralnimi pripomočki in uporabo uporabniškega vhoda. Na vsako od štirih anten na mizi položimo različne predmete: vrednostni žeton, plošček, karto in identifikacijsko kartico. Nato na uporabniškem vmesniku pritisnemo vsako od treh tipk in preverimo odziv, ki je predstavljen v tabeli 4.

Igralni pripomoček	Antena/tipka	Odziv
Žeton vrednosti 500	Igralec 2	Info,Reply:Axis 6, Timestamp 42ff, SpecificID 801fb814,ReadAddress 0032, Data: 0500 9999 ffff ffff
Plošček	Igralec 1	Info,Reply:Axis 0, Timestamp 7aff, SpecificID c00153ae, ReadAddress 0032, Data: 0000 0000 0000 0000
Karta	Igralec 3	Info,Reply:Axis 1, Timestamp 4cff, SpecificID c02ea73d, ReadAddress 0032, Data: 0000 0000 0000 0000
Identifikacijska kartica	Miza	Info,Reply:Axis 8, Timestamp 01ff, SpecificID c07c2451, ReadAddress 0032, Data: 0000 0002 ffff ffff
Gumb preveri	Uporabniški vhod 0	Info,External:Input 1 changed state to 0
Gumb pokaži karte	Uporabniški vhod 1	Info,External:Input 0 changed state to 0
Gumb odstopi	Uporabniški vhod 2	Info,External:Input 2 changed state to 0

Tabela 4: Odziv čitalnika ob prisotnosti značk.

Vidimo, da smo dobili odziv, kakršnega smo si želeli. Mogoče je razlikovati med sporočili, ki jih povzroči sprememba na uporabniškem vhodu, kjer za vsak pritisk tipke dobimo različen odziv in sporočili o prepoznanih značkah v dosegu anten. Pri teh lahko razberemo podatke, ki smo jih zahtevali od čitalnika, pomen posameznih delov pa je naslednji:

- *Info, Reply*

Takšen začetek sporočila nam pove, da je v njem predstavljen odziv značke.

- *Axis*

S številko je predstavljana os, na kateri je bila zaznana značka.

- *Timestamp*

Časovni žig čitalca, ob katerem je zaznal značko.

- *SpecificID*

Unikatna identifikacijska številka posamezne značke.

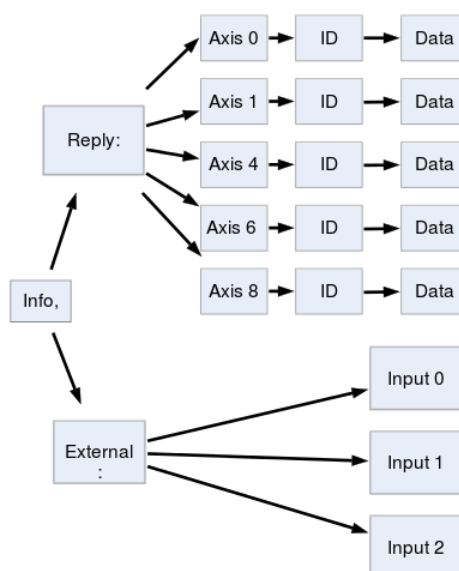
- *ReadAddress*

Naslov dela pomnilnika, iz katerega so prebrani podatki.

- *Data*

Podatki iz pomnilnika značke.

Pred obdelavo podatkov, moramo vsak odziv razčleniti in iz njega izluščiti podatke. V ta namen v razredu *readerClient* ustavimo metodo, v kateri uporabimo orodja za delo z nizi, s pomočjo katere sporočila razčlenimo na način, ki je prikazan v diagramu na sliki 17.



Slika 17: Razčlenjevanje podatkov.

Iz diagrama je razvidno, da odzive najprej razvrstimo glede na vrsto. Pri uporabniškem vhodu nas zanima le podatek, na katerem vhodu se je zgodila sprememba. Odzive značk dodatno razčlenimo na posamezne osi, pri vsaki pa potrebujemo še unikatno identifikacijsko številko značke in prebrane podatke. Podatke nato posredujemo metodi, ki skrbi za obdelavo podatkov.

4.3.5 OBDELAVA PREJETIH PODATKOV

V tem razdelku bomo predstavili pristope, ki smo jih uporabili pri obdelavi podatkov. Zaradi obsežnosti uporabljenih metod in procedur, se bomo posvetili logičnemu vidiku delovanja programa. Z opisom ključnih elementov bomo skozi potek igre predstavili način priprave podatkov za izvoz v izhodno datoteko.

4.3.5.1 Baza podatkov

Najpomembnejši element pri obdelavi podatkov igre pokra je podatkovna baza. V njej vodimo seznam značk in njihovo razporeditev po antenah. Ko s čitalnika prejmemo nov odziv, preverimo, kje se je značka nahajala v trenutku pred tem. Tako določimo premik igralnega pripomočka, kar predstavlja glavni dogodek pri analizi dogajanja na mizi. Dodatno v bazo shranimo še podatke o kartah in izkaznicah, ki jih uvozimo iz pripravljene tekstovne datoteke, saj so ti podatki znani že pred začetkom igre. Vsaka identifikacijska številka značke enolično določa posamezno izkaznico ali kartico, zato v tabelah vodimo povezavo med številko in ostalimi podatki. Tako za vsako karto določimo pot do njene slike, izkaznici pa določimo ime in pot do slike lastnika.

4.3.5.2 Od premika ploščka do pritiska tipke

Za uspešno spremljanje igre moramo nekje začeti. Ob zagonu programa se nam informacije o vseh značkah shranijo v bazo, kar nam že omogoča izračun nekaterih podatkov. Tako lahko na primer seštejemo, kolikšna je vrednost žetonov na posamezni anteni. Za nas so takšni podatki seveda pomembni, vendar želimo spremljati celotno igro pokra. Ko so vsi igralni pripomočki pripravljeni, se nova igra začne s premikom ploščka.

Premik ploščka je ključni dogodek pri spremljanju igre. Lahko ga neposredno določimo, saj je plošček le eden, kar pomeni, da moramo spremljati le eno identifikacijsko številko. Ko ga delivec premakne na mizi od enega igralca do drugega, pomeni, da se je začela nova igra. V bazi podatkov preverimo, če so vse igralne karte na anteni delivca in na mestu za stave ni

nobenega žetona. V nasprotnem primeru v polje za informacije na izhodu napišemo napako v kateri je zapisano na katerem igralnem mestu so nepričakovani igralni pripomočki.

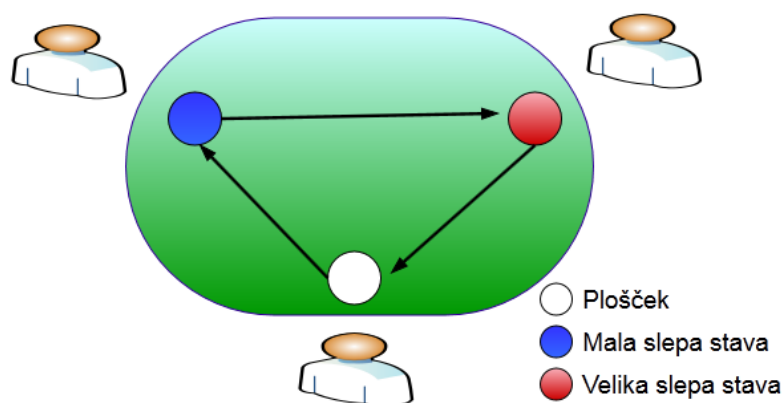
Pred začetkom igre moramo nastaviti še odštevalnik časa za višanje slepih stav. Ker se plošček za delivca navadno nahaja na mizi, izkoristimo to lastnost in odštevalnik zaženemo v trenutku, ko se plošček pojavi na anteni pred delivcem. Predhodno nastavimo interval spremembe in višino slepih stav, oba podatka pa zapišemo v polje za informacije na izhodu.

Ob začetku nove igre preverimo, kateri igralci v njej sodelujejo. Prvi pogoj je, da ima igralec seštevek žetonov večji od 0. V ta namen seštejemo vrednosti vseh žetonov na antenah. Pri tem vodimo tudi evidenco koliko žetonov posamezne vrednosti ima vsak igralec. Oba podatka zapišemo na izhod.

Ko seštevamo vrednosti žetonov v bazi, preverimo tudi katera izkaznica je na posameznem igralnem mestu. Informacije o imenu in poti do slike igralca uporabimo na izhodu.

Ko so pogoji za začetek igre izpolnjeni, lahko iz lokacije ploščka določimo naslednje podatke o igri (slika 18):

- Kateri igralec mora položiti malo slepo stavo
- Kateri igralec mora položiti veliko slepo stavo
- Kateri igralec je prvi na vrsti



Slika 18: Odvisnost igralnih pozicij od lokacije ploščka.

Podatek o višini velike in male slepe stave je že znan, zato igralcema na teh dveh mestih na izhodu pripravimo pot do slike, ki predstavlja malo oziroma veliko slepo stavo. Ko igralca na

mizo postavita žetone v vrednosti slepe stave, pošljemo na izhod prazen niz in tako слиki zbrisemo.

Ko so slepe stave postavljene, delivec lahko začne z deljenjem kart. Vsakemu igralcu razdeli po dve karti, igro pa začne igralec za slepo stavo. V tem trenutku lahko na izhodu pripravimo podatke za polje karte v roki, ki jih prikažemo vsakemu igralcu, ki sodeluje v igri. Igralec, ki je na vrsti, odigra svojo potezo, pri tem pa spremljamo premikanje žetonov z njegove antene na anteno za stave. Spremembe seštevka žetonov sproti pripravljamo na izhodu.

Po odigrani potezi je na vrsti naslednji igralec. Delivec sporoči spremembo s pritiskom na gumb na uporabniškem vhodu. Ko zaznamo pritisk, na izhod pošljemo podatek, kateri igralec je na vrsti.

Naslednji igralec lahko ob preveliki stavi prejšnjega odstopi iz nadaljnje igre. To lahko zaznamo s premikom igralnih kart od igralčeve antene na anteno za stave ali pa s pritiskom delivca na uporabniškem vhodu. V tem primeru si moramo to zapomniti in v naslednjih korakih igralca preskočiti. Na izhod sporočimo izbris kart igralca.

Ko vsi igralci odigrajo, nastopi naslednja faza igre. Delivec obrne tri skupne karte na anteno za stave, igro pa v trem primeru začne igralec za ploščkom. Z uporabo baze podatkov na izhodu pripravimo slike za vse tri karte in sporočimo spremembo igralca, ki je na vrsti.

Nato se z uporabo do sedaj opisanih postopkov odigrata še dve fazi igre, ker poleg premikanja žetonov na izhod zapišemo še obe skupni karti, dokler ne pride do zaključka igre. Ko imajo v zadnji fazi igralci izenačene stave, delivec pritisne na gumb na uporabniškem vhodu. Na antenah igralcev, ki še sodelujejo v igri, preverimo identifikacijske številke kart in jih pošljemo na izhod.

Zmagovalec na koncu premakne žetone na svojo anteno in igra se s premikom ploščka začne znova.

4.3.5.3 Izhodni podatki

Podatke, ki jih pridobimo na izhodu metod uporabljenih za njihovo obdelavo, ob vsaki spremembi shranimo v izhodno datoteko. Ob tem skrbimo, da je struktura datoteke skozi čas ista, saj lahko le tako podatke uporabimo za prikaz na zaslonu. Ker bodo do datoteke dostopali zaslони iz različnih lokacij, vključimo datoteko v skupno rabo.

Hkrati izdelamo še metodo, s pomočjo katere krmilimo uporabniški izhod na čitalniku. Z uporabo podatkov o igralcu, ki je trenutno na vrsti na uporabniškem izhodu, prižgemo lučko,

ki prikazuje tega igralca. Čitalniku pošljemo zato serijo treh ukazov, v katerih mu povemo, kateri dve lučki naj ugasne in katero prižge. Ukazi so za primer drugega igralca naslednji:

- `sendToReader("ExternalOutputBit(2,0, IO_DEFAULT) \n\r");`
- `sendToReader("ExternalOutputBit(3,1, IO_DEFAULT) \n\r");`
- `sendToReader("ExternalOutputBit(4,0, IO_DEFAULT) \n\r");`

4.4 PRIKAZ NA ZASLONU

Za prikaz igre na zaslonu uporabimo orodje za digitalno oglaševanje. Programska oprema je sestavljena iz dveh delov, urejevalnika vsebine in predvajalnika. Z uporabo urejevalnika vsebine določimo razporeditev posameznih elementov na zaslonu. Ko je vsebina pripravljena, nastavitve pošljemo predvajalniku. Z uporabo datoteke, ki smo jo pripravili s programom, se na zaslonu prikazuje potek igre.

4.4.1 UREJEVALNIK VSEBINE

Najprej namestimo urejevalnik vsebine, G-Digital Manager. Za delovanje potrebujemo računalnik z operacijskim sistemom Microsoft Windows in nameščenimi orodji za .NET. Namestitev je enostavna, poleg lokacije imenika za programske datoteke določimo še imenik, v katerem bodo shranjene multimedijske datoteke. Po uspešni namestitvi program zaženemo. Ob prvem zagonu se pojavi okno za registracijo izdelka, po vnosu licenčne kode pa se nam pojavi glavno okno programa.

V glavnem oknu dostopamo do orodij, ki jih bomo uporabili za izdelavo konfiguracije zaslona:

- Uvoz datotek
- Dinamični podatki
- Priprava zaslona

Na zaslonu bomo uporabili različne slike, zato jih moramo najprej uvoziti v program. Za uvoz datotek kliknemo na gumb *Add Files*. V oknu, ki se nam odpre, izberemo datoteke, ki se nato prenesejo v imenik, ki smo ga pripravili ob namestitvi. Ko so vsebine uspešno uvožene, se nam prikažejo v glavnem oknu programa.

Dinamične podatke, ki jih želimo prikazati na zaslonu v našem primeru predstavlja datoteka, ki smo jo pripravili s programom. Zato moramo v urejevalniku vsebine najprej določiti način

dostopa in obdelave datoteke. To storimo z vgrajenim orodjem *DataWatcher*. V meniju programa izberemo *Dynamic Data* → *DataWatcher*. V prvem oknu ustvarimo nov vir vsebine, nato pa v sedmih korakih nastavimo vse potrebno za dostop in uporabo pri pripravi zaslona. To so:

- Osnovne nastavitve

V osnovnih nastavitvah določimo ime in interval preverjanja vira. Ime lahko poljubno izberemo, interval preverjanja pa nastavimo v odvisnosti od hitrosti spreminjanja izvornih podatkov. Ker se v našem primeru datoteka spremeni ob vsakem premiku značke na anteni, vrednost nastavimo na dovolj majhno, da ne izgubimo na odzivnosti sistema. Izberemo interval 100 ms.

- Vir podatkov

Pri nastavitvi vira podatkov določimo tip in pot do njega. Tip izberemo iz ponujenih možnosti: datoteka, imenik, spletni vir, podatkovna baza. Način dostopa je odvisen od vira, saj moramo za dostop do podatkovne baze vnesti podatke o strežniku, uporabniku, geslu, podatkovni bazi in iskalni poizvedbi, za ostale tri pa le pot do želenega vira. Ker je v našem primeru datoteka v javno dostopnem imeniku na računalniku, kjer se izvaja program, navedemo pot za dostop do nje.

- Razčlenjevanje podatkov

Vhodno datoteko moramo pred uporabo razčleniti. Izbiramo med tremi različnimi razčlenjevanji dokumentov: XML (ang. *Extensible Markup Language*), HTML (ang. *HyperText Markup Language*) in *delimiter field*. Z uporabo prvih dveh lahko razčlenimo dokumente posamezne vrste, mi imamo pa pripravljeno datoteko za tretjo izbiro. Določimo, na kakšen način so podatki v datoteki ločeni po vrsticah in kako po stolpcih. Za razlikovanje stolpcev tako izberemo znak za vejico, za vrstico pa novo vrstico.

- Razporeditev polj

Za vsak stolpec v vhodni datoteki določimo ime, ki ga bomo uporabljali pri nadaljnji obdelavi. Podatki se namreč iz vira z uporabo razčlenjevanja shranijo v notranjo podatkovno strukturo, na katero se sklicujemo pri uporabi v urejevalniku prikaza.

- Preizkus delovanja

V petem koraku preizkusimo delovanje nastavitev. S pritiskom na gumb Test se izvede dostop do vira, razčlenjevanje in uvoz v podatkovno strukturo. Prikaže se nam izpis pridobljene tabele, ki je prikazan na sliki 19.

- Obdelava podatkov

Z uporabo skript lahko dobljene podatke še dodatno obdelamo. Ker imamo podatke v vhodni datoteki v obliki, kakršno lahko uporabimo na zaslonu, ta korak preskočimo.

- Prikaz obdelanih podatkov

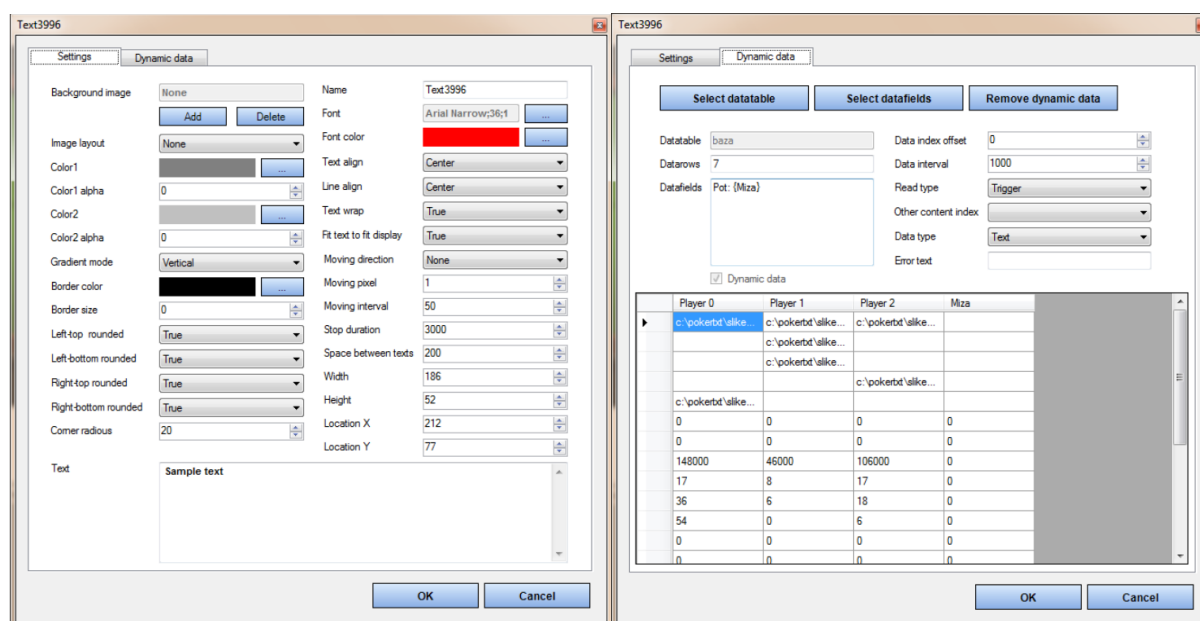
Ker v prejšnjem koraku nismo izvedli nobene obdelave, se podatki prikažejo v enaki obliki kot v petem koraku.

Player 0	Player 1	Player 2	Miza
c:\pokertxt\slikeMiza\karte.jpg	c:\pokertxt\slikeMiza\karte.jpg	c:\pokertxt\slikeMiza\karte.jpg	
	c:\pokertxt\slikeMiza\laVrsti.jpg		
	c:\pokertxt\slikeMiza\dealer.jpg		
		c:\pokertxt\slikeMiza\smallBlind.jpg	
c:\pokertxt\slikeMiza\bigBlind.jpg			
0	0	0	0
0	0	0	0
148000	46000	106000	0
17	8	17	0
36	6	18	0
54	0	6	0
0	0	0	0
0	0	0	0
0	0	0	0
Preflop postavi in placaj blinde			
c:\pokertxt\slikeMiza\karteVRoki.jpg	c:\pokertxt\slikeMiza\karteVRoki.jpg	c:\pokertxt\slikeMiza\karteVRoki.jpg	

Slika 19: Orodje DataWatcher.

Ko imamo pripravljene vse vsebine, lahko začnemo s pripravo zaslona. V meniju glavnega okna izberemo *Players* → *Add player* in odpre se nam okno za urejanje vsebin na zaslonu predvajalnika. V njem najprej določimo ime, ločljivost in pot do računalnika, na katerem se

izvaja predvajalnik, nato začnemo pripravljati zaslon z uporabo pripravljenih gradnikov. Najprej kliknemo na gumb *Add background image* in z izbiro slike, ki smo jo predhodno uvozili med vsebine, določimo ozadje zaslona. Uporabimo sliko mize na črnem ozadju. V naslednjih korakih na ozadje postavimo osnovne gradnike – okvirje za vsebino. To je samostojno polje, ki se nam pojavi na zaslonu z klikom na gumb *Add text*. Kot lahko razberemo iz imena, je v osnovi namenjen prikazu besedila, vendar ga lahko uporabimo tudi za prikaz dinamičnih vsebin. Z desnim klikom na okvir izberemo možnost *Settings* in odpre se nam okno za urejanje nastavitev in vsebine okvirja (slika 20).



Slika 20: Nastavitve okvirja besedila.

V prvem zavihku urejamo nastavitve oblike okvirja in oblike ter vsebine besedila. Tako lahko izberemo dimenzije, barvo ozadja in morebitno obrobo okvirja. Besedilu, ki ga vnesemo v okno lahko določimo vrsto pisave in način prikaza. Poleg tega, da je besedilo prikazano statično, lahko izbiramo med različnimi načini premikanja. Na takšen način lahko na majhnem prostoru prikažemo daljša besedila, ki bi bila ob uporabi manjše pisave težko berljiva. V tem oknu odstranimo obrobo, nastavimo barvo ozadja na prozorno in velikost določimo glede na element igre, ki ga predstavlja.

Vsebino določimo z uporabo dinamičnih podatkov. V drugem zavihku tako izberemo tabelo podatkov, ki smo jo pripravili z orodjem *DataWatcher*. Vsebino okvirja lahko zdaj določimo z besedilom v posameznem polju tabele. Z izbiro podatkovnega tipa, lahko dodatno izberemo način prikaza vsebine. Privzeto je nastavljen na besedilo, izberemo pa lahko tudi možnost

slike. V tem primeru je v okvirju namesto besedila prikazana slika, do katere vodi pot zapisana v besedilu polja.

Z uporabo teh lastnosti naredimo okvirje za vse elemente igre in jih razporedimo po mizi. Posamezne elemente lahko združimo in tako prikažemo (slika 21):

1. Višina male in velike slepe stave ter čas do naslednjega zvišanja. Oboje prikažemo kot besedilo.
2. Podatki o posameznem igralcu. Obrobo predstavlja okvir sive barve, ki se lahko spremeni v belo, kar prikazuje kateri igralec je na vrsti. V njem prikažemo sliko igralca, v dveh poljih z besedilom pa njegovo ime in število žetonov.
3. Pred vsako igralno mesto postavimo elemente, ki s sliko prikazujejo kateri igralec mora postaviti malo oziroma veliko slepo stavo. Dodamo še dva elementa za prikaz kart. Prvega uporabimo za prikaz, ko igralec sodeluje v trenutni igri in ima karte pred seboj, drugega pa za morebiten prikaz njegovih kart ob koncu igre.
4. Podatke o višini stave posameznega igralca in skupnemu seštevku žetonov na mizi prikažemo kot besedilo na sredini mize.
5. Skupne karte, ki so obrnjene na mizi prikažemo z slikami.



Slika 21: Konfiguracija zaslona.

Tako pripravljeno konfiguracijo pošljemo na predvajalnik, ki smo ga že namestili in pripravili za delovanje.

4.4.2 PREDVAJALNIK

Predvajalnik G-Digital Player smo namestili na računalnik, preverili povezavo do računalnika, kjer se izvaja program in izhodne datoteke ter ga zagnali. Predvajalnik se na začetku poveže z urejevalnikom vsebine od katerega prenese datoteke, ki jih potrebuje za delovanje. Tako na lokalni disk shrani vse slike in nastavitve zaslona. Slike so v obliki slikovnih datotek, nastavitve posameznih elementov in virov dinamičnih podatkov pa v XML datoteki. Od tega trenutka naprej lahko predvajalnik samostojno dostopa do vira podatkov in prikaže potek dogajanja pri igri pokra.

Na naslednjih slikah je prikaz poteka ene igre pokra. Na igralnih mestih so razporejeni igralci Mojca, Janez in Mateja. Plošček je pred Janezom, vsak igralec ima na svojem mestu tudi identifikacijsko kartico. Mateja mora položiti malo slepo stavo, Mojca pa veliko, zato je pred njima znak, ki označuje da stava še ni položena (slika 22).



Slika 22: Začetek nove igre.

Ko igralca položita stave, napis na njunem igralnem mestu izgine in delivec razdeli karte med igralce (slika 23). Pri tem opazimo napis o posameznih stavah. Na sredini mize je prikazan skupni seštevek stav, pred vsakim igralcem pa je vsota, ki jo je položil v igro.



Slika 23: Igralca položita slepe stave in igra se začne.

Nato sledijo stave posameznih igralcev, ko so med seboj izenačene pa delivec na mizo obrne tri skupne karte (slika 24). Prva na vrsti v tem krogu je Mojca, tako je okvir okoli njene slike osvetljen.



Slika 24: Prve skupne karte.

V naslednjih dveh krogih poteka igra podobno kot v prvem. Do konca ostanejo v igri vsi trije igralci. Delivec pritisne na tipko na uporabniškem vhodu in na zaslonu se prikažejo karte igralcev (slika 25).



Slika 25: Zaključek igre.

5 SKLEPNE UGOTOVITVE

Pri izdelavi diplomskega dela smo uspešno vzpostavili sistem za zajem, obdelavo in prikaz podatkov na zaslonu pri igri pokra. S primernim načrtovanjem smo izdelali mizo za poker in jo opremili z napravami RFID. Ob izdelavi aplikacije smo vzpostavili komunikacijo s čitalnikom in izdelali metodo za razčlenjevanje vhodnih podatkov. Tako pripravljene podatke smo nato uporabili v metodi, v kateri smo morali ob upoštevanju pravil pokra vpeljati logiko za prepoznavo igre pokra na osnovi zaznave in pomnjenja pozicij posameznih igralnih pripomočkov na različnih antenah. Z uporabo metode za obdelavo podatkov smo pripravili izhodno datoteko namenjeno uporabi v orodju za digitalno oglaševanje. Z urejevalnikom vsebine smo določili način dostopa in razčlenjevanja podatkov v izhodni datoteki in jih uporabili za grafični prikaz informacij na zaslonu. Iz podatkov smo ustvarili posamične elemente in jih z uporabo slike ozadja sestavili v grafično podobo mize. Konfiguracijo zaslona smo poslali na predvajalnik, ki je v nadaljevanju samostojno dostopal do podatkov in jih redno osveževal. Ob stalnem spreminjanju podatkov o igralcih, njihovih žetonih in premikih igralnih pripomočkov gledalec dobi vtis neposrednega spremljanja žive igre.

Pri celotni izvedbi je največ časa predstavljala izdelava metode za obdelavo podatkov. Na podlagi pravil pokra smo morali pripraviti proceduro za spremljanje posamezne igre, nato pa s testiranjem preveriti pravilno delovanje. Zaradi velikega števila izjem pri igri je bil to najbolj obsežen del programa.

Ker smo bili omejeni s strojno opremo, smo lahko izdelali le eno mizo s tremi igralnimi mesti. Zato celotnega sistema nismo mogli uporabiti na način, kakršnega smo predvideli na začetku, za izpeljavo turnirja z večjim številom miz. Kljub temu smo mizo izdelali z mislijo za takšen način uporabe. Kompaktnost mize smo dosegli z vgraditvijo vseh sestavnih delov v okvir mize, tako da za pripravo mize za delovanje potrebujemo le priključka za napajanje in povezavo v omrežje.

Prednosti uporabe mize RFID pri igri pokra se najbolj izkažejo pri organizaciji turnirjev. Z uporabo mize organizatorji dosežejo boljše informiranje igralcev in obiskovalcev med potekom turnirja, s prikazom igre na zaslonih pa dosežejo večji krog gledalcev, kar lahko izkoristijo za promocijske namene. S spremljanjem celotnega poteka igre na centralni lokaciji se zmanjša potreba po velikem številu nadzornikov, saj lahko celoten sistem uporabimo za nadzor poteka igre. Iz podatkov v sistemu lahko organizatorji vidijo podatke o posameznem

igralcu in razporeditvi žetonov po mizah, kar jim omogoča enostavno presedanje igralcev na prosta igralna mesta. Podatke v sistemu lahko uporabijo tudi za drugačen način prikazovanja igre. Tako lahko na osnovi istih podatkov igro prikažejo v uporabniškem vmesniku za igro pokra preko interneta ali pa jih uporabijo za enostavnejšo kasnejšo obdelavo video vsebin s prikazom igre.

Vse opisane možnosti uporabe hkrati predstavljajo tudi izhodišča za nadaljnje delo, saj bi za njihovo uporabo morali razviti celoten sistem z več različnimi izhodnimi podatki in možnostmi za uporabnike. Potrebna bi bila izdelava uporabniškega vmesnika z orodjem za zbiranje prijav za turnir in vodenje podatkov o igralcih. Celoten potek turnirja bi prikazali v uporabniškem vmesniku in organizatorjem omogočili določeno stopnjo samodejnega vodenja in nadzora turnirja.

6 LITERATURA

[1] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, John Wiley & Sons, Ltd., 2010

[2] (2012) The History of RFID. Dostopno na:
http://www.transcore.com/pdf/AIM%20shrouds_of_time.pdf

[3] (2012) M. Cardullo, W. Parks, US3713148 (A) - Transponder Apparatus And System. Dostopno na:
http://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=3713148&KC=&FT=E&locale=en_EP

[4] B. Glover, H. Bhatt, RFID Essentials, O'Reilly, 2006

[5] (2012) Digital Signage: The Right Information in All The Right Places. Dostopno na:
<http://www.itu.int/en/ITU-T/techwatch/Pages/digital-signage-standards.aspx>

[6] (2012) Multi Antenna Reader System. Dostopno na:
http://www.magellan-rfid.com/res/PDF/Brochures%20and%20Documents/027-01-010-DOC-2MARS-24AT_PB.pdf

[7] (2012) Programska oprema G-Digital. Dostopno na:
http://www.g-d.si/casino/file/G-Digital_slo.pdf

[8] (2012) The Java Language: An Overview. Dostopno na:
<http://java.sun.com/docs/overviews/java/java-overview-1.html>

[9] (2012) NetBeans Editor. Dostopno na:
<http://netbeans.org/features/ide/editor.html>

[10] (2012) Foxwoods Casino Poker Room. Dostopno na:
http://poker.about.com/od/cardroomscasinos/ss/foxwoodstour_3.htm

[11] (2012) Magellan Technology – Gaming. Dostopno na:
<http://www.magellan-rfid.com/applications/gaming>

[12] (2012) The Tournament Director Screenshot. Dostopno na:
<http://www.thetournamentdirector.net/screenshots/blackAndGreen.gif>