

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

Dokumentacija za završni projekt iz predmeta Multimedijски sustavi

FOUR IN A ROW

Bjanka Bašić

Jelena Lončar

Voditelj: *Goran Igaly*

Zagreb, siječanj 2019. godine

Sadržaj

1. Opis programa.....	1
2. Grafički dio.....	2
3. Opis funkcija.....	5
4. Moguća poboljšanja.....	8
5. Literatura.....	9

1. Opis programa

Ideja je bila iskoristiti naučeno na kolegiju Multimedijски sustavi u svrhu izrade jednostavne igrice, koja koristi umjetnu inteligenciju za implementaciju poteza računala. Željeli smo izraditi vlastitu verziju postojeće, poznate igre s određenim grafičkim elementima, koje bismo napravile i uredile pomoću alata i programa korištenih na Multimedijским sustavima te koja se može implementirati u Processingu, koristeći znanje koje smo stekle.

Na kraju smo odabrale *Four in a Row*, odnosno *Connect Four* (hrv. Četiri zaredom), igru u kojoj je cilj, kao što samo ime kaže, formirati horizontalnu, vertikalnu ili dijagonalnu liniju sastavljenu od 4 diska/žetona identične boje. Osnovna premisa igre pojavljuje se i u raznim drugim verzijama, od križić-kružića pa do mnogih današnjih mobilnih aplikacija kao što je *Candy Crush*. To je jedan od razloga zašto nam se činila idealnom za izraditi.

Trebale smo riješiti nekoliko idejnih problema:

1. Grafički izgled igre
2. Implementaciju svih potrebnih funkcija
3. Kako će računalo raditi poteze



2. Grafički dio

Trebale smo odlučiti kako će točno grafički igra izgledati i koje ćemo boje koristiti. Odlučile smo unutar igre ponuditi dvije različite teme, između kojih igrač prije upuštanja u igranje bira jednu, osobno preferiranu. Prva je tema takva da što više podsjeća na klasičan izgled igre: ploča (rešetka) je plava, a žetoni su žuti i crveni. Druga je tema manje konvencionalna, svjetlija te se sastoji od sive ploče i ljubičastih te zelenih žetona. Osnovni gradivni element ploče jest jedna malena pločica. Taj je način izgradnje ploče veoma fleksibilan jer omogućava naknadno mijenjanje i prilagođavanje dimenzija ploče, ovisno o potrebama. Glavni grafički dijelovi koje je trebalo izgraditi bili su:

- Ploča
- Žetoni za oba igrača
- Pozadina
- Grafički elementi za slučajeve pobjede, poraza i izjednačenja
- Strelica (pomoć - pokazuje igraču koji novčić treba pomaknuti u prvoj igri)

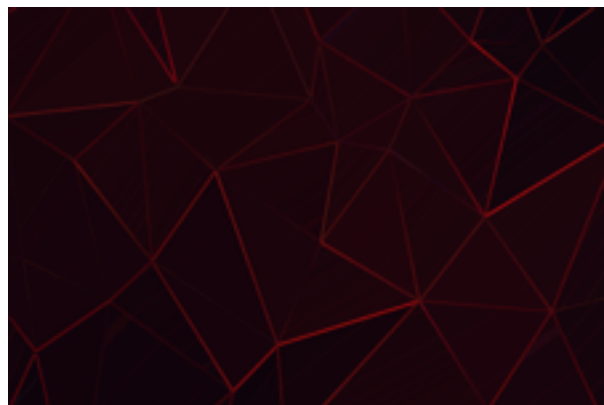
Ploču i žetone izradile smo u programu GIMP. Prvi je korak bio oblikovati osnovne objekte, kojima se onda po potrebi može modificirati boja te tako dodavati nove teme. Inače jednostavan proces bio nam je donekle otežan zbog nastojanja da objekti budu osjenčani, kako bi ostavljali dojam trodimenzionalnosti. Na sljedećim se slikama može vidjeti jedan gradivni element ploče, malena pločica, kao i žetoni. Pločica te žetoni u prvome retku dijelovi su osnovne, klasične teme, dok oni u drugome čine elemente alternativne, svjetlije teme.



Za izradu grafičkih elemenata za izvještavanje o pobjedi, porazu i izjednačenju te pomoćne strelice, osim GIMP-a korišten je i online program za kreiranje i uređivanje slika, PicMonkey (<https://www.picmonkey.com/home>). Dizajn tih elemenata napravljen je pomoću bogate ponude alata PicMonkeyja, a potom je konačan izgled dobiven oblikovanjem i otklanjanjem vidljive pozadine pomoću GIMP-a. Grafike za obavještanje o izjednačenju, pobjedi igrača (čovjeka), gubitku (pobjedi kompjutera) te strelice vidljive su na sljedećim slikama:



Pozadina je izrađena na temelju slike pronađene na internetu, koju smo modificirale pomoću GIMP-a, nastojeći postići da se dobro uklapa uz ostale grafičke dijelove. Bojanjem u GIMP-u napravljene su dvije verzije, jedna tamnija, za osnovnu temu, a druga svjetlija, za alternativnu temu. Obje pozadine mogu se vidjeti na sljedećim slikama:



U Processing kodu igre koristile smo int varijablu *gameScreen*, koja može poprimiti 4 različite vrijednosti. Kada je vrijednost varijable *gameScreen* jednaka 0, tada se crta Main Menu, odnosno početni izbornik, pomoću kojega igrač može odabrati temu, težinu, odabrati opciju čitanja pravila ili započeti igranje. Izgled početnog izbornika ostvaren je pomoću pozadinske slike izrađene u PicMonkeyju, kao i grafički elementi za izvješćavanje o ishodu po završetku igre, kao i pomoću rect (rectangle) elemenata, koji su poslužili kao gumbi pomoću kojih igrač odabire preferirane opcije. Odabirom opcije prikazivanja pravila igre, vrijednost varijable *gameScreen* postaje 3 te se, umjesto glavnog izbornika, crta nova pozadina, također izrađena pomoću PicMonkeyja, a koja igrača informira o pravilima. Pritiskom miša, igrač se ponovno vraća u početni izbornik. Odabirom opcije Play, igra započinje, a *gameScreen* poprima vrijednost 1 te se crtaju svi prethodno opisani elementi same igre, među kojima su najistaknutiji pozadina (prikazana na slikama na stranici 3), ploča te žetoni. Po završetku igre, *gameScreen* poprima vrijednost 2, a na ekranu se prikazuju grafički elementi za obavješćavanje o pobjedi, gubitku te izjednačenju, također prikazani na stranici 3. Na sljedećim su slikama vidljive pozadina glavnog izbornika te grafika opcije pravila:



3. Opis funkcija

void setup()

Zove se pri samom pokretanju programa pa inicijaliziramo sve PImage i pomoćne varijable. Pretpostavlja se da igrač po prvi put igra igru pa se vrijednost boolean varijable *isFirstGame* postavlja na true. Naime, ako je vrijednost *isFirstGame* jednaka true, tada se na ekranu crta pomoćna strelica, koja igraču olakšava snalaženje u igri. Nadalje, vrijednost varijable *gameScreen* postavlja se na 0, što znači da će se inicijalno prikazati glavni izbornik.

void draw()

Ovisno o ekranu koji želimo crtati, izvršava različite stvari. Postoje četiri mogućnosti:

1. *gameScreen=0* – prvi ekran koji vidimo kad pokrenemo program. Pozadina mu je Main Menu slika, opisana u dijelu 2. Grafički dio, te crta sve gumbе, pritiskom na koje možemo mijenjati postavke igre.
2. *gameScreen=1* – ekran u kojem se odigrava stvarno igranje igre. Prvo se vrši odabir igrača koji će prvi vući potez. Ako je vrijednost varijable *isFirstGame* jednaka true, prvi igrač automatski postaje računalo, kako bi čovjek, koji potencijalno igru igra po prvi put, mogao vidjeti kako izgleda povlačenje poteze te se time lakše snašao u igri. U protivnom, ako se ne radi o prvoj igri, igrač koji prvi vuče potez bira se nasumično, korištenjem *random* funkcije. Ono što točno funkcija radi u svakom trenutku uvelike ovisi o tome koji je igrač na redu (računalo ili čovjek). Pozivaju se mnoge pomoćne funkcije, o kojima će detaljnije biti riječ kasnije.
3. *gameScreen=2* – ekran koji se prikazuje kada završi igra. Klikom miša igrač se vraća na glavni izbornik, a pritiskom tipke escape automatski izlazi iz igre.
4. *gameScreen=3* – ekran s pravilima igre.

boolean overRect(int x, int y, int width, int height)/ boolean overCircle(int x, int y, int diameter)

Provjerava nalazi li se miš iznad pravokutnika ili kruga određenog parametrima funkcije. Koristimo za kreiranje funkcionalnosti gumba u glavnom izborniku.

void mouseClicked()

U slučaju da je vrijednost varijable *gameScreen* jednaka 2 ili 3, omogućava da se klikom miša igrač vrati na glavni izbornik.

void darkTheme()/void lightTheme()

Funkcije koje služe za učitavanje elemenata teme koju je igrač odabrao kao svoju preferiranu.

void makeMove(int[][] currBoard, int player, int column)

Ovisno o stupcu u koji se ispusti žeton, funkcija u matricu *currBoard*, koja reprezentira ploču koja se koristi za igranje, u dani stupac i redak koji odgovara prvom slobodnom mjestu u rešetci upisuje oznaku igrača koji je žeton ubacio.

void drawBoard(int[][] currBoard)

Funkcija crta ploču određenu podacima spremljenim u matrici *currBoard*.

void drawTile(int x, int y, int player)

Funkcija crta žeton igrača *player* s koordinatama lijevog gornjeg kuta jednakim (x,y).

void getNewBoard()

Funkcija koja služi za pražnjenje ploče, odnosno brisanje podataka o žetonima koji su bili u ploči.

void mouseDragged()

Funkcija koja se koristi u svrhu animiranja povlačenja žetona po ploči od strane ljudskog igrača (za vrijeme njegovog poteza). Ako su vrijednosti varijabli *humanMove* i *draggingToken* jednake true, tj., ako je čovjekov red i ako je krenuo povlačiti žeton s hrpe žetona odgovarajuće boje prema rešetci, odnosno njenom ciljanom stupcu, tada se pri svakom pomaku pritisnutog miša mijenjaju koordinate *xcor* i *ycor*, na kojima se crta dani žeton.

void mouseReleased()

Služi za mijenjanje vrijednosti boolean varijable *mouseR*. Kada čovjek, nakon što je neko vrijeme povlačio žeton po ekranu za vrijeme svog poteza, otpusti tipku miša, vrijednost varijable *mouseR* poprima vrijednost true. Nakon toga, animira se ispuštanje žetona u odgovarajući stupac rešetke.

int getComputerMove(int diff)

Funkcija koja vraća indeks stupca u koji će kompjuter, kada je njegov red, ubaciti svoj žeton. Pozivom funkcije *getPotentialMoves*, opisane u nastavku, generira se polje svih potencijalnih poteza koje kompjuter može povući. Prolaskom po tome polju, određuje se koji su potezi optimalni za računalo, odnosno koji ga vode k najboljem mogućem ishodu. Generira se polje svih optimalnih poteza, a potom se jedan od njih, onaj koji će se u konačnici i povući, određuje nasumično, pomoću funkcije *random*.

void deepCopy(int[][] board1, int[][] board2)

Pomoćna funkcija koja vrši “duboko kopiranje”, tj. sadržaj matrice *board1* kopira u matricu *board2*. Zove se u funkciji *getPotentialMoves* pri generiranju svih mogućih scenarija igre, određenim svim mogućim potezima dvaju igrača.

int[] getPotentialMoves(int[][] currBoard, int player, int diff)

Rekurzivna funkcija koja je ključna za odabir sljedećeg poteza računala. Ovisno o vrijednosti varijable *diff*, odnosno težini koju je igrač odabrao, pretražuje prostor svih mogućih stanja do određene dubine, određene spomenutom varijablom. Korištenjem funkcije *deepCopy* generira nove ploče, od kojih svaka odgovara scenariju koji slijedi nakon

određenog povučenog poteza. Inicijalno se numeričke vrijednosti “isplativosti” svih poteza postavljaju na 0. Ako se ispostavi da neki potez rezultira pobjedom igrača *player*, njegova se numerička vrijednost postavlja na 1, ako rezultira pobjedom suparnika igrača *player*, postavlja se na -1, a ako rezultira punom pločom, tj. neriješenim stanjem, vrijednost ostaje 0. Inače, vrijednost poteza postaje usrednjena vrijednost sume vrijednosti svih poteza. Funkcija vraća polje s numeričkim vrijednostima “isplativosti” svih mogućih valjanih poteza.

int getLowestEmptySpace(int[][] currBoard, int column)

Funkcija za dani stupac, određen varijablom *column*, vraća indeks prvog slobodnog stupca odozgo u matrici *currBoard*, koja reprezentira trenutno stanje na ploči igre.

boolean isValidMove(int[][] currBoard, int column)

Funkcija vraća boolean vrijednost true ili false, ovisno o tome je li potez određen ubacivanjem žetona u stupac s indeksom *column* valjani potez. Potez nije valjan ako je vrijednost varijable *column* manja od nule, veća ili jednaka *BOARDWIDTH* (širini ploče) (indeksiranje počinje od nule) ili je referencirani stupac već pun.

boolean isBoardFull(int[][] currBoard)

Funkcija provjerava je li ploča puna, tj. je li ispunjena cijela matrica *currBoard* (reprezentacija stanja na ploči).

boolean isWinner(int[][] currBoard, int player)

isWinner vraća vrijednost true ako je igrač određen varijablom *player* pobjednik igre, a false ako nije. Točnije, provjerava je li dani igrač uspio ostvariti neprekidnu liniju četiri svoja žetona u nizu negdje u ploči (tj. matrici *currBoard*).

4. Moguća poboljšanja

- Dodati pozadinsku glazbu
- Padanje žetona u rešetku popratiti odgovarajućim zvučnim efektom
- Povećati spektar opcija za težinu igre (po mogućnosti dodati težinu lakšu od sadašnje *Easy*)
- Dodati mogućnost biranja dimenzija ploče za igranje
- Implementirati multiplayer opciju, tj. omogućiti da igru igraju dva ljudska igrača jedan protiv drugoga (umjesto da čovjek igra protiv računala)
- Omogućiti igraču da bira koja će boja žetona, od dvije ponuđene za određenu temu, biti njegova (umjesto da mu boja bude dodijeljena po izboru teme)
- Omogućiti veću prilagodljivost grafičkog aspekta igre, tj. da igrač ima više ponuđenih tema, više opcija za boje žetona, rešetke i pozadine
- Budući da je igra riješena, odnosno poznata je optimalna strategija, koja onome igraču koji je prvi na redu omogućava da uvijek pobjedi, može se igraču ponuditi opcija igranja protiv računala koji igra upravo na optimalan način. Tako bi igrač mogao sam naučiti kako igrati optimalno. Alternativno, mogao bi se u igri implementirati čisto edukativan tutorial, koji igraču pokazuje slijed optimalnih poteza bez da se od njega traži da sam igra i vuče poteze
- Nakon pritiska na gumb za odabir teme ili težine u glavnom izborniku, na neki način vizualno označiti da je upravo taj gumb pritisnut, tj. odabran (ostaviti oznaku sve dok korisnik potencijalno ne pritisne na neki gumb kojim odabire alternativnu opciju)
- Tijekom samog igranja igre negdje na ekranu naznačiti koja je težina trenutno odabrana
- Na ekran (kada je *gameScreen=1*) postaviti gumb za izlazak iz same igre natrag u glavni izbornik
- Omogućiti igraču da se predomisli neposredno nakon povlačenja nekog svog poteza, dok protivnički žeton još putuje prema svom odredištu u rešetci (dok još nije ubačen, tj. dok još nije jasno u koji će ga točno stupac računalo ubaciti)
- Ako igrač igra više partija neposredno jednu za drugom (bez izlaska iz igre i ponovnog ulaženja), ispisivati na ekranu broj pobjeda igrača, odnosno računala do tekućeg trenutka

5. Literatura

<https://processing.org/reference/>

<https://discourse.processing.org>

https://en.wikipedia.org/wiki/Connect_Four

<http://inventwithpython.com>