PRIRODNO-MATEMATIČKI FAKULTET

UNIVERZITET U SARAJEVU

Projekat iz predmeta Razvoj softvera

Igra 2048

Predmetni mentori: Student:

Mr. Vedad Letić Emir Bračković

Doc.dr. Damir Hasić

# **UVOD**

2048 je igrica za jednog igrača koja je dizajnirana od strane Web develpoera Gabriele Cirullia iz Italije. Cilj igrice je pomjerajući blokove brojeva i kombinirajući iste doći do broja 2048. Ova igrica je originalno napisana u programskom jeziku JavaScript i CSS, ali danas postoje i verzije (klonovi) koji su napisani u C++ i Vala.

Igra je napravljena za Android i IOS sistem, odnosno mobilne uređaje. Pošto je talijanski stručnjak napravio igru samo za jedan vikend stavio je da je besplatna. Vrlo brzo nakon toga igra je postala viralna i doživjela je ogroman uspjeh. U manje od sedam dana imala je preko 4 miliona klikova.

Nakon nekog vremena, počele su izlaziti druge verzije te igre i vremenom su se dodavale raznije opcije poput „undo“ dugmeta , rang liste svih igrača poredanih po njihovom rezultatu i tako dalje. Također vremenom su dodane i reklame u igrici , tako da su se mnogi autori kopija obogatili.

2048 se igra na mreži 4x4 polja. Komande na telefonu su povlačenje prstom u jednu od strana (gore , dolje, lijevo, desno). Nakon toga svi blokovi idu u tu stranu. Igra počinje sa 2 bloka u kojima je vrijednost 2. Spajanjem 2 ista bloka pomoću metode prevlačenja objedinjuju se ta 2 bloka i u njemu je njihov zbir. Pri svakoj komandi, bilo da smo uspjeli neke blokove spojiti ili ne, nastaje novi blok negdje na ploči vrijednosti 2, a njegova lokacija je odlučuje slučajnim odabirom. Spajanjem blokova, cilj igrice je da se dođe do broja 2048, odnosno posljednji korak je spojiti 2 bloka vrijednosti 1024.

U originalnoj verziji kada bi korisnik došao do bloka vrijednosti 2048, igrica bi završavala i izašalo bi obavještenje da je korisnik pobjedio. Danas ima mnogo verzija u kojima se igrica i nakon toga nastavlja i bitan je samo rezultat odnosno „score“ koji je korisnik napravio. Nakon završetka igre, ukoliko je korisnik na mreži njegov rezultat se šalje u bazu podataka i zatim on dobija svoje mjesto na rang listi svih korisnika.

Ova igra nije ni danas izgubila na svojoj popularnosti upravo zbog raznih verzija koje izlaze pa se korisnici i koji su se zasitili igrom, vrate se njoj kada vide da se pojavilo nešto novo. Primjer toga je da u mnogo verzija korisnik više nije ograničen na ploču dimenzija 4x4 , nego postoje i 2x2, 3x3, 5x5, 6x6 i slično.

# **O IZRADI PROJEKTA**

Cilj projekta je da u toku semestra koristeći metode koje smo naučili na predavanjima i vježbama pokušamo napraviti Igru 2048. Pored predavanja i vježbi dozvoljeno i neophodno je bilo korištenje interneta i dugo istraživanje o raznim metodama koje bi pomogle pri izradi programa.

Projekat je urađen u programskom jeziku Java i prilagođen je samo za računare/laptope. Potrebno je napraviti konzolu koja će sadržavati 16 polja odnosno 4X4. Blokovi (polja) na početku popunjeni su samo na 2 mjesta sa brojem „2“ . Pozicija tih blokova je izabrana slučajnim odabirom (random metodom).

Korisnik upravlja programom (igrom) pomoću tastature odnosno strelica (gore , dolje , lijevo , desno). Svaki put kada se izvrši potez (ako je to uopće moguće) nastaje novi blok vrijednosti „2“ koji također ima poziciju izabranu random metodom, ali samo na mjestima koja su slobodna.

Bilo je potrebno napraviti da ukoliko se u istom redu jedan do drugog nađu dva bloka iste vrijednosti, ako bi korisnik kliknuo strelicu lijevo ili desno, oni bi se spojili. Isto tako ako bi bili u istoj koloni jedan do drugog, od ta dva bloka nastao bi jedan. Vrijednost tog bloka je njihov zbir.

Nakon toga potrebno je napraviti da kada se pojavi blok koji ima vrijednost 2048 igrica završava. Nakon kraja igrice sa tim ishodom, iskočiti će poruka sa natpisom „ Čestitamo, pobjedili ste !“. Isto tako prije svakog poteza korisnika mora se provjeriti da li je uopće moguće ikakvo micanje kroz polja odnosno da li moguće uraditi bilo koju komandu. Ukoliko nije moguć ni jedan potez onda izlazi poruka „Kraj igre !“.

Pri završetku igre dodana je metoda da se igra počne iz početka (restart igre). To je korisniku omogućeno ako klikne dugme „ESC“.

Ostalo je još uljepšati igricu, pa sam tako za brojeve iste vrijednosti, koristio istu pozadinu u bloku. U mom projektu korištene su razne nijanse plave/ljubičaste boje, tako da blok sa brojem „2“ je najsvjetliji, blok sa brojem „4“ je tamniji i tako dalje do broja 2048 koji je specijalne boje. Pored toga dodano je da ukoliko je kraj igre, ploča sa poljima postaje svjetlija da bi bilo lakše pročitati poruku.

Posljednji korak je „scoreboard“ odnosno rezultat koji je korisnik uspio ostvariti. On stoji u donjem desnom uglu i svaki put kada se spoje dva bloka, poveća se za vrijednost njihovog zbira.

# **METODE I FUNKCIJE**

U sljedećem dijelu dokumentacije, pojasnit cu većinu metoda i funkcija koje sam koristio za kreiranje igre 2048.

U glavnoj (main) klasi implementirao sam JFrame metode koje služe za konzolu. Tu sam iskoristio:

***setTitle*** - služi za postakvu naslova konzole

setSize – služi da se postavi veličina konzole

setResizable (false) – onemogućava korisnika da mijenja veličinu konzole

setLocationRelativeTo(null)- centrira konzolu na ekranu

setDefaultCloseOperation- kontrolira gašenje prozora (konzole)

setVisible(true)- prikazuje konzolu na ekranu

Drugi dio programa je Emir2048 koji sadrži:

addKeyListener(new KeyAdapter()

Ovaj dio generalno služi za kontrole u igri I prihvaćanje komandi sa tastature. Ukoliko korisnik klikne neko dugme koje nije definisano, neće se desiti ništa, a u sljedećem dijelu koda dodao sam opcije za ESC dugme, strelicu lijevo, desno, gore I dolje.

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_ESCAPE) {

resetIgre();

Kao što sam već naveo, klik na dugme ESCAPE pokreće reset igre. Klik dugmeta se registruje sa metodom “getKeyCode” I onda ispitujemo da li je taj klik jedan od komandi. Na početku koda postavljeni su parametri pobjeda I poraz na false, odnosno to govori da još nije došao kraj igre.

if (!pobjeda && !poraz) {

switch (e.getKeyCode()) {

case KeyEvent.VK\_LEFT:

left();

break;

case KeyEvent.VK\_RIGHT:

right();

break;

case KeyEvent.VK\_DOWN:

down();

break;

case KeyEvent.VK\_UP:

up();

break;

}

U ovom dijelu koda definisano je da ukoliko nije pobjeda true I ukoliko nije poraz true, pomoću switch metode primaju se komande sa strelica tastature.

public void resetIgre()

Funkcija resetIgre , restartuje , odnosno resetuje čitavu igru. Ona briše stare podatke i kreira novu mrežu polja u igri formata 4x4.

private void novoPolje()

private List < Polje > slobodnoMjesto()

Ove funkcije služe za generisanje random broja koje služi za određivanje pozicije novog polja vrijednosti “2”. U okviru ovih funkcija ispitao sam da li je to mjesto koje je slučajnim odabirom dobijeno slobodno, ukoliko jeste na tom mjestu nastaje novo polje.

public void left()

public void right ()

public void up()

public void down ()

U okviru ovih funkcija omogućeno je kretanje naših blokova (polja) metodom rotacije svih polja I njihovim pomjeranjem. Tačnije omogućeno je pozivanje funkcija koje su napravljene sa tim zadatkom.

private boolean compare(Polje[] prvo, Polje[] drugo)

Funkcija koja služi za upoređivanje 2 polja odnosno da li su im iste vrijednosti ili ne. Kao parameter prima 2 polja.

private Polje[] rotate(int angle)

Funkcija koja služi za rotaciju polja. Kao parameter prima ugao . Rotacija je omogućena pomoću matematičkih metoda implementiranih u okviru ove funkcije.

private Polje[] moveLine(Polje[] staraLinija)

private Polje[] mergeLine(Polje[] staraLinija)

Ove funkcije služe za pomjeranje i spajanje polja. U okviru njih provjerava se da li je mjesto u stranu koju želimo pomjeriti slobodno. Ako jeste onda se pomjeranje izvršava. Ako nije, to polje ostaje na mjestu, a pomjeraju se ona koja imaju mogućnost. U okviru funkcije mergeLine ubacio sam da svaki put kada se 2 bloka spoje, na rezultat se doda vrijednost bloka koji je nastao. Također odmah tada se ispituje da li postoji blok vrijednosti 2048. Ukoliko postoji dolazi do kraja igre.

boolean KrajDaNe()

Postavljena je na false. Kada nema više mogućih poteza, vrati true i igra se prekida.

public void paint(Graphics g)

private void crtajPolje(Graphics g2, Polje Polje, int x, int y)

Došli smo do grafike. U okviru ovih funkcija pomoću raznih metoda uradio sam crtanje polja I konzole igre. Npr. setColor , postavlja boju pozadine polja, fillRect ispunjava to polje I tako dalje.

Pored ovih metoda za crtanje dodao sam da ukoliko dođe do pobjede ili poraza , konzola mijenja boju I postaje svjetlija I ispisuje se poruka “Kraj igre “ ili “Cestitam ! Pobjedili ste 😊 “. Tu stu koristene funkcije drawString, setFont, setColor I slično.

public Color getBackground()

Ovo je posljednja metoda koju ću navesti. U okviru nje napravio sam više opcija I lično mi je bilo najzanimljivije raditi sa njom. Kreirao sam switch koji prima kao parameter vrijednost polja I vraća koje boje to polje mora biti. Lično sam birao svaku boju. U Javi boje se unose ili imenom npr. “black “ ili hexadecimalnim putem, samo što pri unosu treba staviti “0x” na početak umjesto “#”.