SVEUČILIŠTE U ZAGREBU

**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

Dokumentacija za studentski projekt iz kolegija Multimedijski sustavi

**IGRICA S LOPTICOM**

Tea Poljanić

Dorada dokumentacije: Tihana Štifanić

Dorada dokumentacije: Ivka Ćaćić, Maja Pavičić i Magdalena Potočnjak

**Sadržaj**

[**Opis programa** 1](#_Toc155744431)

[**Implementacija** 2](#_Toc155744432)

[**main.pde** 2](#_Toc155744433)

[**Navigacija.pde** 2](#_Toc155744434)

[**PrvaIgra.pde** 3](#_Toc155744435)

[**Rezultati.pde** 4](#_Toc155744436)

[**DrugaIgra.pde** 4](#_Toc155744437)

[**Gumb.pde** 5](#_Toc155744438)

[**Loptica.pde** 5](#_Toc155744439)

[**PowerUp.pde** 5](#_Toc155744440)

[**Text.pde** 6](#_Toc155744441)

[**Zvuk.pde** 6](#_Toc155744442)

[**Moguća poboljšanja** 7](#_Toc155744443)

**Opis programa**

U ovom projektu implementirane su dvije jednostavne igrice koristeći Processing. Radi se o igricama u kojima se koriste loptice. Jedna verzija je za jednog igrača dok je druga za dva igrača.

Igrica za jednog igrača (Skupi loptice) sastoji se od tri vrste loptica: crvena, plava i zelena. Crvena je naša loptica i njom upravljamo pomoću miša. Cilj je pomoću nje dotaknuti što više zelenih loptica. Dodirom na zelenu lopticu, naš rezultat se povećava, a broj plavih loptica se povećava za jedan. Igra traje sve dok ne izgubimo sve živote, a na početku ih imamo tri. Povremeno se među lopticama pojavljuju i powerUps-i odnosno žute loptice čijim sakupljanjem dobivamo neke benefite i ljubičaste loptice čijim sakupljanjem dobivamo neke kazne. Najboljih 10 igrača plasira se na rang listu.

Druga igrica (Pong) sastoji se od jedne loptice te dvije pločice s kojima udaramo lopticu. Svaki igrač upravlja jednom pločicom i miče je tako da uspije udariti lopticu. Cilj je poslati lopticu protivniku tako da je on ne uspije udariti. Pobjednik je onaj koji bude u tome uspješan 5 puta.

Potrebne dodatne biblioteke: controlP5 za tekstni okvir i Minim za zvuk.

**Implementacija**

Prilikom implementacije, trebalo je odvojiti različite „zaslone“ igrice. Razlikujemo tako početni zaslon, zaslon s pravilima, zaslon s postavkama, zaslone prve i druge igrice te zaslone koji dolaze nakon prve i druge igrice. Također, postoje dva različita zaslona koji se prikazuju prije prve, odnosno druge igre, a služe za upis imena jednog, odnosno dva igrača. Trenutni zaslon pratimo pomoću varijable *prozor* dok varijabla *prethodniProzor* pamti prozor iz kojeg smo došli do trenutnog zaslona. Zaslon s postavkama prikazuje se prilikom pritiska na gumb „POSTAVKE“ ili, ako je igra u tijeku, pritiskom na tipku P.

Kompletan kod je refaktoriran i raspoređen u više manjih datoteka radi lakšeg snalaženja pa ćemo redom opisivati komponente svake od pojedinih datoteka. Isto tako, sve multimedijske datoteke korištene u projektu koje nisu *.pde* su stavljene u mapu *data*. To su redom:

* slika *candy.png* koja nije korištena, ali je ostavljena za primjer moguće upotrebe stavljanja na gumb
* font *NerkoOne-Regular.ttf* preuzet s GoogleFontsa koji se koristi prilikom ispisa svog teksta u projektu
* Excel tablica *rang.csv* u koju se sprema rang lista prve igre
* slika *svemir.jpg* koja je korištena za pozadinu
* mapa *Zvuk* u kojoj su datoteke korištene za pozadinski zvuk odnosno zvučne efekte

## **main.pde**

Ovo je glavna datoteka iz koje se pozivaju sve ostale funkcije. Prilikom pokretanja programa najprije se pokreće funkcija *setup()*. U njoj se postavlja veličina i pozadina prozora te instanciraju neki potrebni objekti, kao što su tablica za rang listu i tekstni okviri za upis imena igrača te varijable za zvuk i pokretanje pozadinske glazbe.

Funkcije *mouseClicked()* i *keyPressed()* nam služe za navigiranje među prozorima ukoliko je miš pritisnut i nalazi se unutar određenog gumba odnosno za otvaranje postavki te kontrolu pločica u drugoj igrici ako je pritisnuta određena tipka.

Nakon ove funkcije, slijedi funkcija *draw()* koja se izvršava ciklički i poziva razne druge funkcije ovisno o prozoru u kojem se nalazimo.

## **Navigacija.pde**

Funkcija *prikaziPocetniZaslon()* na početni prozor, koji se otvara prilikom pokretanja programa, ispisuje naslov „Igrice s lopticom“ te stvara četiri gumba s naslovima „SKUPI LOPTICE (1 igrač)“, „PONG (2 igrača)“, „PRAVILA“ i „POSTAVKE“. Klikom na određeni gumb, otvara se traženi prozor.

Zatim imamo funkciju *prikaziPostavke()* koja iscrtava zaslon postavki. Na njemu su prikazani gumbi za regulaciju pozadinske glazbe, zvuka te za navigaciju među prozorima. Gumbi koji su sive boje nisu aktivni, odnosno pritiskom na njih se neće dogoditi ništa. Dok gumbi crvene boje mogu biti pritisnuti te prilikom pritiska obavljaju određenu funkciju. U slučaju gumba „MUSIC ON“ ili „MUSIC OFF“ uključuje se ili isključuje reprodukcija pozadinske glazbe, a u slučaju gumba „SOUND ON“ ili „SOUND OFF“ uključujemo odnosno isključujemo reprodukciju zvukovnih efekata kao što su (ne)plasiranje na rang listu u prvoj igri te udarac loptice u pločicu za igru Pong. Gumbi mijenjaju boje u odnosu na trenutne postavke pozadinske glazbe i zvučnih efekata čije stanje pohranjujemo u varijable *music* odnosno *sound*. Ukoliko smo u zaslon postavki došli iz prozora jedne od igara, tada je aktivan gumb „IGRAJ PONOVNO“ koji će resetirati trenutnu igru na početne postavke. Gumb „NAZAD“ ima funkciju vraćanja na prethodni zaslon te ukoliko je to zaslon za igru, služi kao „*resume“*.

Nadalje, imamo funkciju *prikaziPravila()* kojom se stvara prozor s pravilima. Tada se prikazuju pravila za obje igrice te gumb pomoću kojeg se vraćamo na početni izbornik. Na samom kraju funkcije ostavljena je za primjer budućeg poboljšanja zakomentirana linija koja na prikazanom gumbu crta sliku bombona spremljenog pod „candy.png“ u mapi *data*.

Također, ovdje je definirana funkcija *osvjeziIgre()*, koja postavlja neke početne postavke, ovisno o igri koja je odabrana (tj. prozoru u kojem se nalazimo). Unutar *osvjeziIgre()*, za prvu igricu postavljamo varijable *rezultat* te *dodatniBodovi* na nulu te inicijaliziramo prvu zelenu i plavu lopticu. U varijablu *brojZivota* stavljamo vrijednost 3, a u *prosloVrijeme* vrijednost 0 jer je igra tek krenula. Za drugu igricu postavljamo lopticu na početnu poziciju, kao i pločice.

## **PrvaIgra.pde**

Ova datoteka sadrži globalne funkcije potrebne za prvu igru. Najprije se susrećemo s funkcijom *prijavaPrvaIgra()* koja crta zaslon za upisivanje imena igrača te gumbe za navigaciju. Pritiskom gumba za odabir prve igrice na početnom zaslonu, iscrtava se tekstni okvir u koji korisnik može upisati svoje ime. Ono se može sastojati od najviše 14 znakova. Također se prikazuje gumb „IGRAJ!“, kojim korisnik započinje igranje prve igrice. Iscrtavaju se i gumbi „NAZAD“ kojim se vraćamo na prethodni zaslon te gumb „POSTAVKE“ koji nas vodi u postavke za zvuk.

Za prvu igricu funkcija *prikaziPrvuIgru()* najprije iscrtava sve loptice koje se trebaju nalaziti na ekranu (jedna crvena, jedna zelena i određeni broj plavih). Ispisuje se trenutni rezultat, proteklo vrijeme igranja igrice te preostali broj života. Također provjerava je li došlo do dodira crvene loptice s nekom drugom i u tom slučaju radi određeni sljedeći korak – ako je dotaknuta zelena loptica, broj plavih se povećava kao i trenutni rezultat, a ako je dotaknuta plava loptica gubi se jedan život pri čemu dolazi do efekta „shake-anja“ crvene loptice. Ako su izgubljeni svi životi, igra završava te se ažurira rang lista igrača na temelju broja bodova i vremena igranja igrice (ukoliko je to potrebno, pamtimo samo najboljih 10 igrača). Isto tako, ukoliko je aktivan neki powerUp funkcija će ispisati tekst o kojem se powerUp-u radi.

Nakon gubitka svih života, funkcija *prikaziKrajPrveIgre()* ispisuje postignuti rezultat, plasman na rang listu ukoliko je do toga došlo te samu rang listu uz gumbe „POČETNI IZBORNIK“, „IGRAJ PONOVNO“ i „POSTAVKE“.

## **Rezultati.pde**

U ovoj datoteci su dvije funkcije koje se brinu o ažuriranju rang liste nakon završetka prve igre te njezinom prikazu na zaslon. Prva od njih *updateRangTable()* računa konačan rezultat te na njega dodaje proteklo vrijeme po formuli

koja je prilagođena silaznom sortiranju najprije po broju bodova, a zatim po vremenu trajanja igre. Broj 10 000 000 je odabran zato što se vrijeme mjeri u milisekundama te je 1 000 000 milisekundi jednako 16,667 minuta pa bi svako igranje igre dulje od tog vremena poremetilo poredak na rang listi, a trajanjeIgre smo morali oduzeti od tog broja jer sortiramo silazno pa bi pri samom zbrajanju dobili da veće vrijeme bude na višem mjestu na rang listi što ne želimo. Također, u funkciji provjeravamo plasman trenutno završene igre na rang listu te ovisno o tome ažuriramo Excel tablicu *rang.csv*. Funkcija *prikaziRangListu()* ispisuje redom redove navedene tablice na ekran.

## **DrugaIgra.pde**

Slično kao za gumb „SKUPI LOPTICE“, vrijedi i za pritisak gumba za odabir druge igrice „PONG“, osim što se tada, u sklopu funkcije *prijavaDrugeIgre()*, prikazuju dva tekstualna okvira za unos imena igrača koja ne smiju biti duža od 14 znakova.

U slučaju druge igrice, funkcija *prikaziDruguIgru()* poziva sljedeće funkcije:

* *nacrtajLopticu()* - crta lopticu na sredini ekrana, to je njena početna pozicija te se ta loptica kasnije miče
* *pomakniLopticu() -* funkcija pomoću koje se loptica miče ovisno o brzini koja je u tom trenutku zadana
* *provjeriOdbijanjeLoptice()* - ova funkcija provjerava je li loptica udarila u bočne strane (ako je, došlo je do kraja igre) ili u gornju i donju stranu (u tom slučaju se samo odbija)
* *nacrtajPlocicu()*  - kreira dvije pločice pomoću kojih igrač udara lopticu
* *pomakniPlocicu()* - u ovisnosti o pritisnutim tipkama - W i S za lijevog igrača te strelica ↑ i ↓ za desnog igrača, pomiče pločicu gore - dolje
* *plocicaUZid()* - provjerava je li pločica došla do vrha ili dna, ukoliko je ne dozvoljava više kretanje u tom smjeru
* *dodir()* - funkcija pomoću koje provjeravamo je li loptica udarila u pločicu, u slučaju da je mijenja joj se brzina te se loptica nastavlja kretati većom brzinom
* *ispisiRezultat() -* pomoću ove funkcije se u svakom trenutku ispisuje trenutni rezultat u vrhu ekrana, tj broj postignutih bodova
* *provjeriKraj() -*  ova funkcija označava kraj igre kada jedan od igrača postigne 5 bodova

Nakon što se odredi pobjednik, funkcijom *prikaziKrajDrugeIgre()* prikazuje se poruka nakon odigrane igrice koja otkriva pobjednika te dva gumba. Jedan gumb je za ponovno igranje iste igrice, a drugi za povratak na početni izbornik. Naravno, tu je i gumb za postavke.

## **Gumb.pde**

Klasa Gumb pohranjuje podatke o poziciji, širini, visini i boji gumba te tekstu i boji teksta koji je napisan unutar samog gumba. Trenutno su svi implementirani gumbi jednake veličine 160x100, većina su crvene boje te jednake boje i veličine teksta pa stoga imamo konstruktor koji prima samo poziciju (x i y koordinate) te tekst koji će biti ispisan unutar gumba. Ostavljen je još jedan konstruktor s mogučnošću zadavanja pozicije, visine i širine gumba.

Funkcija članica klase *nacrtajGumb()* crta gumb sa prethodno zadanim parametrima i ispisuje tekst crne boje koji je centriran u sredinu gumba. Navedena funkcija preopterećena je s još dvije koje omogućuju crtanje gumba na drugoj poziciji (tada se ta pozicija pamti u varijablama *xPom* i *yPom* te varijabla *pomaknut* označava da je gumb nacrtan na drugim koordinatama) odnosno u drugoj boji (trenutno korištena siva boja za detekciju aktivnih gumba) što dobiva kao argument tipa float odnosno color.

Kako bismo detektirali pritisnuti gumb, funkcija članica *unutar()* nam vraća vrijednost true ako je trenutna pozicija miša unutar okvira gumba, odnosno false ukoliko nije ili ako je gumb sive boje pa je neaktivan.

Ostavljena je za primjer i buduću nadogradnju funkcija članica *prikaziSliku(String slika)* pomoću koje je moguće unutar gumba iscrtati sliku čiji izvor pošaljemo kao argument funkcije.

## **Loptica.pde**

Klasa Loptica služi nam za implementaciju svih zelenih i plavih loptica u prvoj igri. Sadrži podatke o trenutnoj poziciji, brzini po obje osi, radijusu i boji. Prilikom konstruiranja objekta potrebno je navesti sve te parametre. Ažuriranje loptica vrši se funkcijom članicom *update()* koja ujedno brine i o odbianju loptica od rubova ekrana odnosno područja igre (gornji rub je spušten ispod razine ispisanog teksta s rezultatom i brojem života).

Ostale funkcije članice obavljaju točno ono što im i samo ime kaže pa ih neću zasebno navoditi. Ukratko, implementirano je pomicanje loptice na nove koordinate, „shake“, povećanje ili smanjenje radijusa loptice te njezine brzine pomicanja po ekranu.

Nadalje, ovdje imamo i globalne funkcije koje vraćaju objekt klase Loptica s random generiranim koordinatama i brzinama pomicanja defaultne plave ili neke druge boje dobivene u argumentu (zasad se koristi samo za zelenu). Funkcija *crtajLoptice()* brine za pravilno prikazivanje loptica na ekranu u svakom trenutku te generiranje powerUp loptica.

## **PowerUp.pde**

Klasa PowerUp novo je implementirana klasa koja nasljeđuje klasu Loptica uz dodatne varijable za *ime* kako bismo znali o kojem se powerUp-u iz liste ponuđenih radi i varijable *a* koja generira oscilaciju kretanja te loptice po y-osi. Kao i kod klase Loptica imamo konstruktor koji traži da navedemo sve parametre, funkciju *update()* koja radi na sličan princip te funkciju *unistiAkoPresaoScreen()* koja brine o recikliranju naših loptica nakon što nestanu s ekrana.

Ostale globalne funkcije u ovoj datoteci su:

* *napraviPowerUp* – generira novu powerUp lopticu na random koordinatama pri vrhu zaslona
* *dodajPowerUp –* kontrolira u kojem trenutku će se stvoriti novi powerUp te nasumično odabire koji će to powerUp biti
* *vrijeme* – određuje je li prošlo unaprije određeno vrijeme za kreiranje novog powerUp-a
* *prikaziPowerUp* – crta powerUp lopticu sve dok je unutar okvira zaslona
* *detektiranaKolizijaPowerUp* – vraća true kada crvena loptica dohvati powerUp, inače vraća false
* *removePowerUp* – resetira potrebne varijable i pokreće novi ciklus brojanja vremena za sljedeći powerUp
* *aktivirajPowerUp* – prilikom dohvaćanja powerUp-a crvenom lopticom ostvaruje dobiveni powerUp

Ostale funkcije su implementacija pojedinih powerUp-ova ili pomoćne funkcije za njih. Djelovanje je jasno iz imena funkcija i po potrebi je opisano u komentarima.

## **Text.pde**

Klasa Text nam služi za jednostavnije ispisivanje teksta na ekran i pamti poziciju, veličinu, sadržaj i boju teksta. Jedan od konstruktora prima sve parametre, a drugi koristi unaprijed zadanu veličinu 30 i crnu boju teksta. Funkcija *ispisiText()* centrirano po sredini ispisuje zadani sadržaj teksta na zaslon.

U ovoj datoteci imamo i dvije pomoćne funkcije koje se tiču ispisa teksta na ekran, a to su funkcija *pretvoriVrijeme(int t)* koja dobivene milisekunde pretvara u string oblika mm:ss, pri čemu su mm minute, a ss pripadajuće sekunde te funkcija *deklinacija(int ukupniRezultat)* koja na temelju dobivenog rezultata vraća string „bod“ (npr. 1 bod, 21 bod itd.), „boda“ (npr. 2 boda, 3 boda, 24 boda itd.) ili „bodova“ za sve ostale slučajeve.

## **Zvuk.pde**

Klasa Zvuk je novostvorena za kontrolu zvučnih efekata u igrici. Konstruktor prima točnu putanju do željenog zvuka. Klasa ima funkciju članicu *reproduciraj()* koja ovisno o varijabli *sound,* koju korisnik može mijenjati u zaslonu postavki, reproducira zvukovni efekt na kojem je funkcija pozvana. Također, tu je i funkcija *setupZvuka()* koja na samom početku izvođenja programa postavlja potrebne varijable za zvuk i pokreće pozadinsku muziku koja se izvodi kao petlja - svaki put kada završi kreće ponovno ispočetka.

**Moguća poboljšanja**

* ~~Dodati upisivanje imena pojedinog igrača.~~
* ~~Dodati spremanje podataka za svakog igrača, spremanje njegovog uspjeha u pojedinoj igrici.~~
* ~~Kreiranje rang liste za pojedine igrice.~~
* Dodati još igrica.
* U prvoj igrici postaviti da plave loptice mijenjaju veličinu i brzinu. 🗹
* Postaviti da crvena loptica iza sebe „vuče“ onoliko loptica koliko je zelenih dotakla (kao zmija).
* U drugoj igrici dodati neke prepreke za zanimljiviju igru.
* Dodati mogućnost biranja boje loptice ili pločice s kojom igramo.
* Omogućiti odabir teme same igrice.
* Dodati mjerenje vremena u obje igre.🗹
* Korištenje izmjerenog vremena u kombinaciji s brojem osvojenih loptica kao kriterij prilikom formiranja rang liste za prvu igricu. 🗹
* Kreiranje rang liste za drugu igricu s kriterijem izmjerenog vremena.
* Implementacija igrice Pong za 2 igrača tako da može igrati preko mreže, a ne samo lokalno.
* Dodati na početni zaslon gumb „RANG LISTA“ koji će doći na prozor 3 s rang listom
* Dodati ikone umjesto teksta na nekim gumbima, npr. kotačić za postavke, zvučnik za sound, kućicu za početni izbornik i slično
* Popraviti da prilikom shake-anja crvene loptice ne možeš gubiti živote par sekundi
* PowerUps allGreen popraviti tako da se sa sudarenom lopticom zbog koje je dobiven bod dogodi da nestane i ponovno pojavi kasnije kao plava loptica na random mjestu
* Dodati još powerUps-a, primjerice nemogućnost gubljenja života određeni period te postaviti negdje vidljivo odbrojavanje koliko traje ta nevidljivost
* Napraviti gumb i novi zaslon s izborom više pozadina ili rangirati pozadine ovisno o postignutim rezultatima, tj napraviti razine ili razne svijetove igrice te sukladno tome mijenjati i pozadinsku muziku
* Dodati još zvukovnih efekata, npr. prilikom dohvaćanja pojedinog powerUp-a i pobjede u drugoj igri
* Promijeniti izgled powerUpsa, npr. oblik kocke i/ili stavit sliku preko objekta npr. slika bombona na lopticu
* Dodati neki od vizualnih efekata poput eksplozije loptice
* Osmislit da se bodovi za svakog pojedinog igrača spremaju negdje i sumiraju kroz više igara te na taj način uvesti „novac“ za kupovanje dodatnih mogućnosti poput dodavanja korisničke slike, otključavanje postavke za dodatni zvuk ili sliku koju može odabrati.Također se može proširiti na powerUpse - ukoliko ostvari dodatan broj bodova, igrač može odabrati koji powerUp se neće koristiti tijekom igre ili odabirat koji powerUp će mu biti dodijeljen u početku igre (čime dobiva prednost).