Catch



Dorada dorade projekta iz kolegija Multimedijski sustavi

Ivana Sablić

Prosinac, 2023.

Sadržaj

[1. Opis projekta](#_Toc153745606)

[2. Implementacija](#_Toc153745607)

[2.1. Game](#_Toc153745608)

[2.2. Home](#_Toc153745609)

[2.3. FirstGame](#_Toc153745610)

[2.4. FallingObject](#_Toc153745611)

[2.5. Settings](#_Toc153745612)

[2.6. GameOver](#_Toc153745613)

[2.7. Highscores](#_Toc153745614)

[3. Poboljšanja u odnosu na prvu verziju](#_Toc153745615)

[4. Poboljšanja u odnosu na prethodnu verziju](#_Toc153745616)

[5. Moguća daljnja poboljšanja](#_Toc153745617)

# 

# Opis projekta

Catch je igra napravljena pomoću Processing-a kao završni projekt iz kolegija Multimedijski sustavi.

Cilj igre je hvatati objekte, odnosno hranu koja pada s vrha ekrana u košaru kojom igrač upravlja pomoću miša ili tipkovnice. Uhvaćena hrana donosi bodove, ali ako ju ne uspijemo uhvatiti smanjuje se broj života igrača. Također, potrebno je izbjegavati bombe koje isto smanjuju broj života, no ukoliko igrač uhvati srce, može nadoknaditi izgubljeni život. Kako vrijeme prolazi, objekti počinju brže padati. Igra završava kada igrač izgubi sve živote.

# Implementacija

Pri izradi ove igrice, različiti dijelovi, odnosno ekrani, predstavljeni su objektima pripadajuće klase. Primjerice početni ekran koji se otvara prilikom pokretanja igrice predstavljen je objektom klase Home.setup()*.* U ovom poglavlju ćemo detaljnije proći kroz klase i funkcije koje su korištene prilikom implementacije. Korištene klase su Home, FirstGame, GameOver, HighScores i Settings.

## 2.1. Game

Osnovna datoteka u projektu je Game.pde. U njoj se nalaze standardne funkcije setup() i draw(*),* neke pomoćne funkcije, deklaracije varijabli, zvukova, slika i slično*.* Sada ćemo detaljnije objasniti uloge pojedinih funkcija.

U funkciji setup() inicijaliziramo sve potrebne varijable i objekte koje koristimo u daljnjem kodu. Učitavamo slike, font i zvučne datoteke. Unutar funkcije setup() poziva se funkcija init() u koju je prebačen dio inicijalizacije iz setup() funkcije koji ovisi o veličini prozora kako bi se kasnije mogla mijenjati njegova veličina.

Dvije najznačajnije funkcije koje ćemo koristiti su myDraw() i mousePressed().

Svaka klasa ima funkciju myDraw() koja služi za iscrtavanje potrebnog sadržaja. Način na koji se manipulira različitim ekranima jest pomoću pripadnih boolean varijabli (homeB, firstGameB, highscoresB, gameOverB, settingsB, isPaused*)*. Dakle, unutar funkcije draw() poziva se funkcija za iscrtavanje samo one klase čija je varijabla postavljena na true. mousePressed() je funkcija koja reagira na klik miša te poziva funkciju myMousePressed() iz klase koja je trenutno aktivna, odnosno čija je pripadna boolean varijabla postavljena na true.

Još neke korisne funkcije su overRect(float x, float y, float width, float height) koja prima koordinate gornjeg lijevog kuta pravokutnika te širinu i visinu pravokutnika te vraća true ukoliko se miš nalazi iznad zadanog pravokutnika, a false inače, overCircle(float x, float y, float diameter) koja prima koordinate centra kruga te njegov promjer i vraća true ukoliko se miš nalazi iznad zadanog kruga, a false inače te appendTextToFile(String filename, String text) koja prima ime datoteke i tekst te dodaje tekst na kraj datoteke.

## 2.2. Home

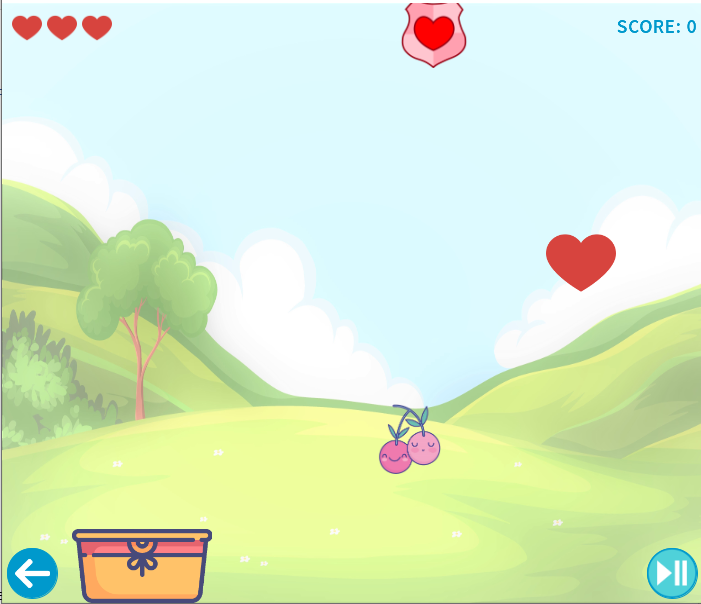


Pomoću varijabli a, b, rectW i rectH rasporedit ćemo elemente na početnom (Home) ekranu.

U ovom slučaju,tj u klasi Home, funkcija myDraw() iscrtava pozadinu, logo te gumbe Play, Settings, Highscores i Exit. Prelaskom miša preko navednih gumba dobivamo tzv. hover efekt, tj. boja gumba iz svijetlo plave prelazi u tamno plavu.

Pomoću myMousePressed() se vrši provjera koji je gumb odabran na početnom zaslonu te se sukladno tome mijenjaju odgovarajuće boolean varijable homeB, firstGameB, settingsB, highscoresB*.* Ukoliko je odabran gumb Highscores*,* dodatno se, poziva i metoda calculate() iz klase Highscores.Provjerava se je li pritisnut neki gumb te se sukladno tome mijenjaju odgovarajuće boolean varijable. Pritiskom na gumb Exit igrica se zatvara. Pritiskom na gumb Highscores poziva se i metoda calculate() iz klase Highscores.

## 2.3. FirstGame



Najprije ćemo nabrojati i pojasniti varijable koje koristimo u ovoj klasi. Varijable su sljedeće:

rnd – slučajno generirani broj pomoću kojeg se odabire padajući objekt (element iz liste objectImages)

lives – broj preostalih života igrača

obj – instanca klaseFallingObject

objects – lista objekata tipa FallingObjects

time1 – vrijeme u kojem je generiran novi objekt

time2 – vrijeme, računa sa svakim pozivom funkcije myDraw()

timespeed1 – vrijeme promjene brzine

timespeed2 – vrijeme, računa se svakim pozivom funkcije myDraw()

time – vremenski razmak za generiranje novih elemenata

low – donja granica za određivanje vremena do generiranja novog objekta

up – gornja granica za određivanje vremena do generiranja novog objekta

secondsToFall – vrijeme potrebno da objekt padne

durationOfBiggerBasket – trajanje povećane košare

durationOfProtection – trajanje zaštite od bombi

timeBasketResized – vrijeme kad je košara povećana

timeProtectionStarted – vrijeme kad je počela zaštita od bombi

originalBasketWidth – originalna širina košare

originalBasketHeight – originalna visina košare

Korištenjem myDraw() iscrtava se pozadina, rezultat i prazna ili puna srca ovisno o tome koliko igrač ima preostalih života. Košara se iscrtava na dnu ekrana s pozicijom na x-osi određenom položajem miša. Ukoliko je igra pauzirana animacija je zamrznuta.

Svakim pozivom funkcije prolazi se kroz listu padajućih objekata te se provjerava njegovo stanje. Ukoliko je uhvaćeni objekt:

* hrana, objekt se miče iz liste te se ažuriraju bodovi
* srce, objekt se miče iz liste te ako broj života igrača manji od tri dodaje mu se život
* puž, objekt se miče iz liste te se igra usporava na način da se varijable low, up i secondsToFall postave na početne vrijednosti
* bomba, objekt se miče iz liste te se broj života igrača smanjuje za jedan

Ako je objekt došao do dna ekrana, odnosno nije uhvaćen, miče se iz liste objekata te se smanjuje broj života igrača za jedan.

Inače, objekt se pomiče prema dolje po sljedećoj formuli

te se iscrtava.

Ukoliko je prošlo vrijeme zadano varijablom time, generira se novi objekt klase FallingObjects i dodaje se u listu objects. Na slučajan način odabire se koji je objekt u pitanju, odnosno odabire se njegova slika. Slike objekata su numerirane, npr. bomba ima naziv 0, jabuka ima naziv 1 itd. Pozicija objekta na x-osi određuje se nasumično. Nakon što je generiran novi objekt, ažurira se vrijeme time1 te se na slučajan način odabire vrijeme do generiranja idućeg objekta, ali unutar intervala [low, up].

Ovisno o postavljenoj razini težine igre svako toliko vremena smanjuje se varijabla secondsToFall te se smanjuju donja i gornja granica, odnosno varijable low i up.

Kao i prije, mousePressed() provjerava je li pritisnut gumb za nazad, odnosno gumb za pauziranje/pokretanje igrice. Ako je stisnut gumb za nazad mijenja boolean varijable homeB i firstGameB te postavlja score na 0, a ako je stisnut gumb za pauziranje/pokretanje mijenja boolean varijablu isPaused te pauzira/pokreće pozadinsku glazbu.

Funkcija setGameOver() rezultat igre sprema u datotetku *scores.txt*, te firstGameB postavlja na false, gameOverB na true kako bi se prikazao Game over ekran.

## 2.4. FallingObject

FallingObject je klasa koja predstavlja objekte koji padaju s vrha ekrana, dakle hrana, srca, puževi, bombe, ruke, štitovi i zvijezde. Klasa sadrži sljedeće varijable:

width – širina slike objekta

height – visina slike objekta

posX – pozicija na x-osi

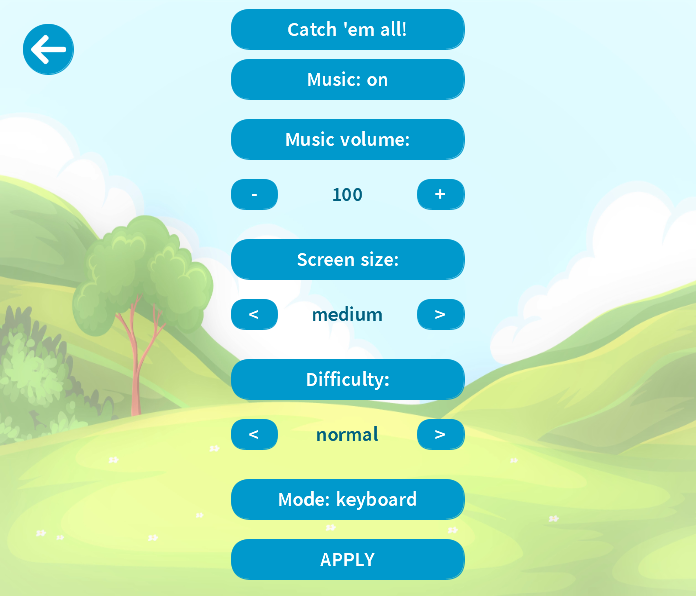
posY – pozicija na y-osi

points – koliko bodova donosi hvatanje objekta

bomb – boolean koji je true ako je objekt bomba, false inače

index – indeks slike iz polja slika objectImages

## 2.5. Settings



U postavkama igrice imamo razne mogućnosti, možemo manipulirati veličinom ekrana, pozadinskom glazbom, možemo birati težinu, način hvatanja (hoćemo li hvatati svu hranu ili samo cupcakes) te hoćemo li igrati pomoću miša ili tipkovnice.

U ovoj klasi su korištene varijable sizetmp i difficultytmp koje služe za pohranjivanje privremenih promjena veličine ekrana i težine igrice. Vrijednosti tih varijabli se nakon pritiska na gumb APPLY pohranjuju u globalne varijable size i difficulty.

Varijable a, b, rectW i rectH su varijable koje služe kao pomoć za raspoređivanje elemenata na ekranu.

Pomoću funkcije myDraw() iscrtavaju se svi elementi vidljivi na gornjoj slici. Ovisno o vrijednosti varijable sizetmp (0, 1, 2, 3) ispisuje se small, medium, large, very large, analogno i za varijablu difficultytmp, ispisuje se easy, normal, hard, extreme.

Funkcija myMousePressed() ima ulogu u sljedećim manipulacijama. Ukoliko je pritisnut gumb Music:On/Off muzika se zaustavlja ili pokreće. Također je moguće podešavati glasnoću muzike pritiskom na gumbe + ili - . Pritiskom na strelice mijenja se veličina ekrana, odnosno težina igrice te konačno, pritiskom na gumb APPLY, sve promjene se pohranjuju.

## 

## 2.6. GameOver

Graphical user interface, website

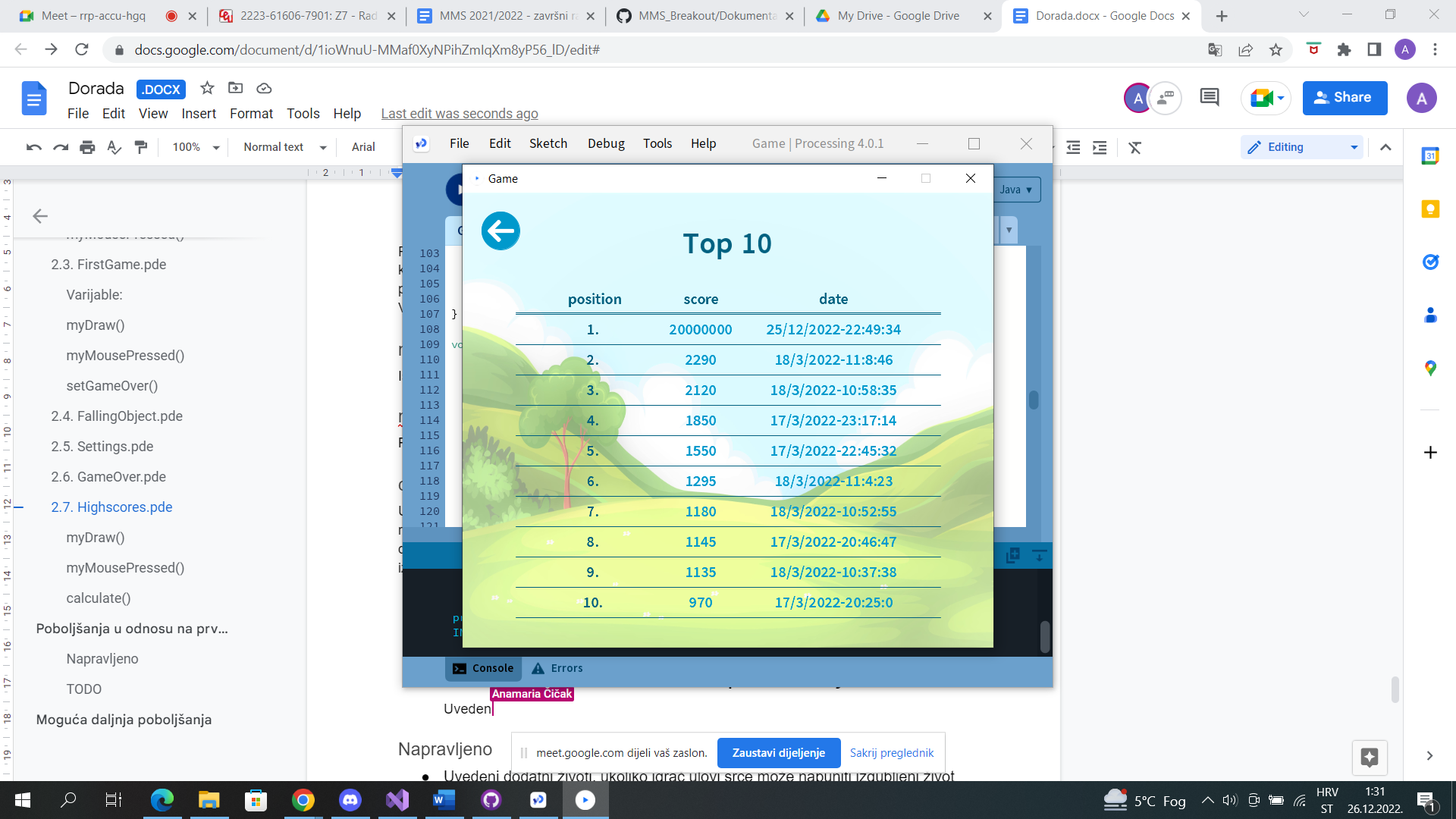
Description automatically generated

Varijable a, b, rectW i rectH su pomoćne varijable za raspoređivanje elemenata na ekranu. Funkcija myDraw() iscrtava pozadinu i sav tekst i gumbe vidljive na slici gore.

myMousePressed() provjerava je li kliknut gumb Play again ili gumb Home. Ukoliko je pritisnut gumb Play again stvara se novi objekt klase FirstGame te se varijabla gameOverB postavlja na false, a varijabla firstGameB se postavlja na true. Također se globalna varijabla score postavlja na 0. Slična stvar se dešava i ukoliko je pritisnut gumb Home.

## 

## 2.7. Highscores



Rezultati se spremaju u datoteku *scores.txt*. Svaki rezultat predstavljen je jednom linijom u kojoj je zapisan rezultat i datum. Polje lines služi za pohranu tih linija, polje scores za pohranu samo rezultata kao integer-a te polje dates za pohranu datuma.

Varijable a i b su varijable koje služe kao pomoć za raspoređivanje elemenata na ekranu.

Ovdje funkciju myDraw() koristimo kako bi scrtali tablicu s Top 10 najboljih rezultata i gumb za nazad, dok funkciju myMousePressed() koristimo za provjeru je li igrač kliknuo na gumb za nazad, ovisno o tome mijenjaju se varijable homeB i highscoresB.

Funkcija calculate() u polje lines sprema tekst iz datoteke *scores.txt* liniju po liniju. Zatim iz polja lines rezanjem stringova dobivamo polja scores i dates u koja spremamo redom rezultate i datume. Nakon toga koristeći *bubble sort* sortiramo polja scores i dates silazno po vrijednostima iz polja scores.

# Poboljšanja u odnosu na prvu verziju

Uvedeni su dodatni životi, odnosno ukoliko igrač u košaru ulovi srce može nadoknaditi izgubljeni život. Uz srce je dodan i novi objekt puž čijim hvatanjem se brzina igre resetira. Omogućeno je igranje pomoću tipkovnice, koristeći tipke Left i Right za pomak košare lijevo, odnosno desno. Odabir kontrola vidljiv je u opcijama Settings.

# Poboljšanja u odnosu na prethodnu verziju

Dodan padajući objekt (*ruka*) koji poveća košaru na 5 sekundi, a zatim ju vrati na početnu veličinu. Ako se uhvati više padajućih objekata za povećanje košare u vremenskom razmaku manjem ili jednakom 5 sekundi, košara se neće dodatno povećati nego će ostati povećana duže.

Dodan padajući objekt *zvijezda* koji donosi više bodova, *štit* koji čini bombe neutralnima na 5 sekundi

Dodana podigra “Catch only cupcakes!” gdje se dobiju bodovi samo ako se uhvati cupcake (i dalje se mogu hvatati posebni objekti kao što su srce, zvijezda itd.), a ako se uhvati neka druga hrana gubi se život (dakle, ako ne uhvatimo nešto što nije cupcake, ne gubimo život).

Dodani posebni zvukovi za svaki “posebni” padajući objekt (onaj koji nije hrana)

dodan gumb za pauziranje/pokretanje igre.

# Moguća daljnja poboljšanja

Mogla bi se dodati mogućnost odabira pozadinske slike, pozadinske glazbe, izgleda košare i slično. Moglo bi se dodati još tipova padajućih objekata (objekt koji uzrokuje pad 5 bombi odjednom ili 5 srca odjednom, objekt koji povećava brzinu pada objekata, objekt koji uzrokuje instantno gubljenje svih života, objekt koji uzrokuje gubitak određene količine bodova). Mogao bi se izmijeniti način pada nekih objekata (umjesto da padaju ravno odozgo prema dolje neki bi mogli vibrirati pri padu ili padati krivudavo odozgo prema dolje).