Catch



Završni projekt iz kolegija Multimedijski sustavi

Martina Čuklić i Mateja Terzanović

Ožujak, 2022.

Sadržaj

[1. Opis projekta](#_heading=h.gjdgxs)

[2. Implementacija](#_heading=h.jbms22pwzkbb)

[2.1. Game.pde](#_heading=h.30j0zll)

[2.2. Home.pde](#_heading=h.2s8eyo1)

[myDraw()](#_heading=h.17dp8vu)

[myMousePressed()](#_heading=h.v9mxo6phunaj)

[2.3. FirstGame.pde](#_heading=h.2xcytpi)

[Varijable:](#_heading=h.dhxjdeubyfo7)

[myDraw()](#_heading=h.1ci93xb)

[myMousePressed()](#_heading=h.3whwml4)

[setGameOver()](#_heading=h.2bn6wsx)

[2.4. FallingObject.pde](#_heading=h.qsh70q)

[2.5. Settings.pde](#_heading=h.3j2qqm3)

[2.6. GameOver.pde](#_heading=h.b5mb2gjzvyuw)

[2.7. Highscores.pde](#_heading=h.1ksv4uv)

[calculate()](#_heading=h.z337ya)

[3. Poboljšanja u odnosu na prvu verziju](#_heading=h.tqgufl5tez74)

[4. Moguća daljnja poboljšanja](#_heading=h.3as4poj)

# 

# 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 detaljnjije proći kroz te i funkcije koje su korištene prilikom implementacije. Klase koje su korištene su Home, FirstGame, GameOver, HighScores i Settings.

## 2.1. Game.pde

Osnovna datoteka u projektu je Game.pde. U njoj se nalaze standardne funkcije setup() i draw(*)* i neke pomoćne funkcije*.* 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() pozivamo funkciju init() u koju je prebačen dio inicijalizacije iz setup() funkcije koji ovisi o veličini prozora kako bi kasnije mogli mijenjati veličinu prozora pozivajući tu funkciju.

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*)*. 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.

overRect(float x, float y, float width, float height) funkcija prima koordinate gornjeg lijevog kuta pravokutnika te širinu i visinu pravokutnika te vraća true ukoliko se miš nalazi iznad zadanog pravokutnika, a false inače. Slično tome, funkcija

overCircle(float x, float y, float diameter) prima koordinate centra kruga te njegov promjer i vraća true ukoliko se miš nalazi iznad zadanog kruga, a false inače.

appendTextToFile(String filename, String text) je funkcija koja prima ime datoteke i tekst te dodaje tekst na kraj datoteke.

## 2.2. Home.pde



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

### myDraw()

Iscrtava se pozadina, logo te gumbi Play, Settings, Highscores i Exit.

Prelaskom miša preko navednih gumba dobivamo tzv. hover efekt, tj. boja gumba iz svijetlo plave prelazi u tamno plavu.

### myMousePressed()

Vrši se 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 mijenjju 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.pde

A picture containing graphical user interface

Description automatically generated

### Varijable:

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

**lives** – broj preostalih života igrača

**obj** – instanca klase *FallingObject*

**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

### 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.

Svakim pozivom funkcije prolazi se kroz listu padajućih objekata te se provjerava njegovo stanje.

Ukoliko je objekt uhvaćen te je riječ o hrani, objekt se miče iz liste te se ažuriraju bodovi.

Ukoliko je objekt uhvaćen te je riječ o srcu, objekt se miče iz liste te ako broj života igrača manji od tri dodaje mu se život.

Ukoliko je objekt uhvaćen te je riječ o pužu, objekt se miče iz liste te se igra usporava na način da se varijable **low**, **up** i **secondsToFall** postave na početne vrijednosti..

Ukoliko je objekt uhvaćen te je riječ o bombi, objekt se miče iz liste te se broj života igrača smanjuje za jedan. Ukoliko je objekt došao do dna ekrana, odnosno ako nije uhvaćen, miče se iz liste objekata te se smanjuje broj života igrača za jedan.

Inače, objekt se pomiče prema dolje prema 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*. Također se pozicija objekta na x-osi određuje 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.

### myMousePressed()

Provjerava je li pritisnut gumb za nazad, mijenja boolean varijable homeB i firstGameB te postavlja score na 0.

### setGameOver()

Rezultat igre se sprema u datotetku *scores.txt*, te firstGameB se postavlja na false, gameOverB na true kako bi se prikazao *Game over* ekran.

## 2.4. FallingObject.pde

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

width – širina slike objekta

height – visina slike objekta

posX – pozicija na x-osi

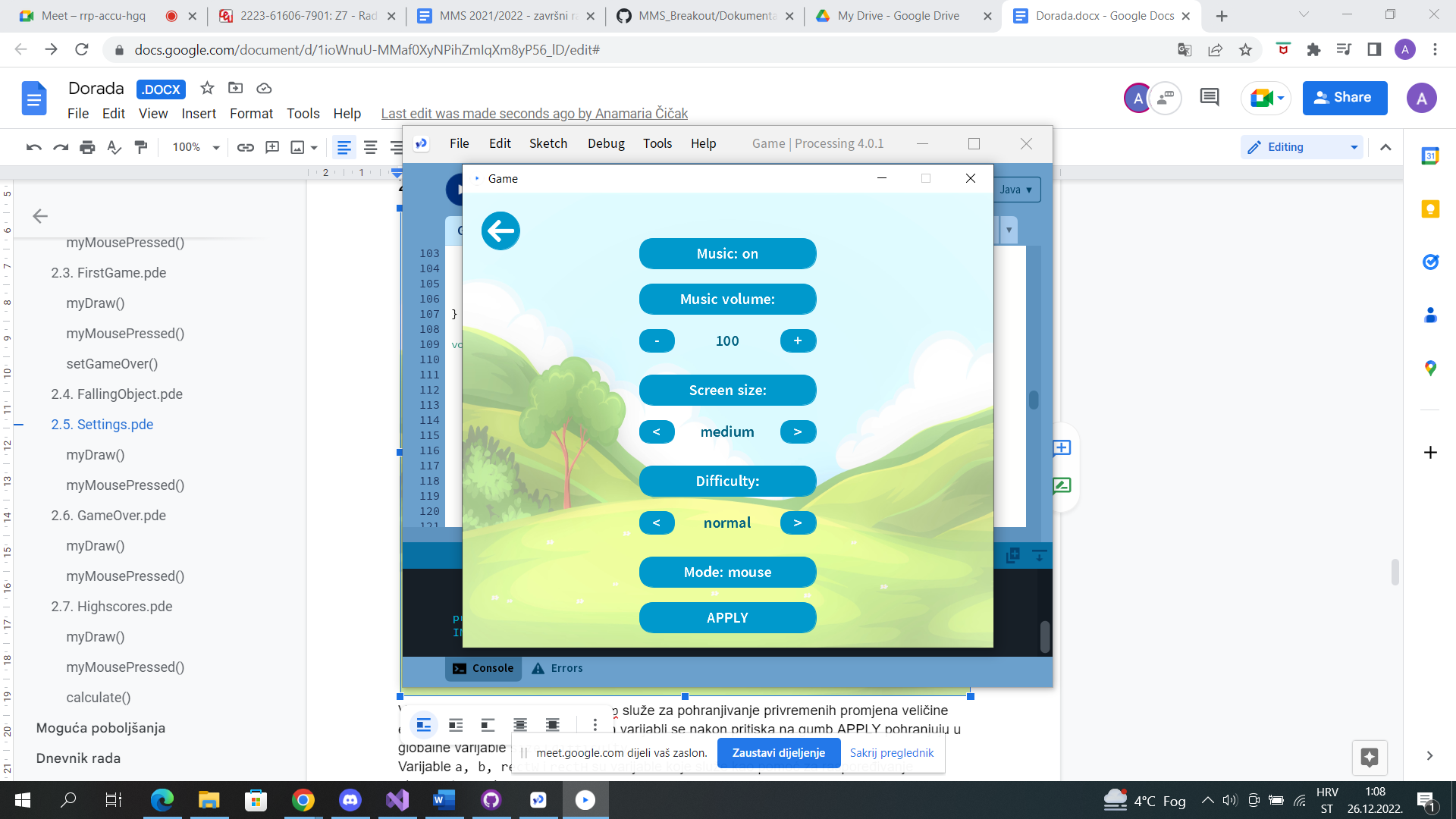
posY – pozicija na y-osi

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.pde



U postavkama igrice imamo razne mogućnosti, možemo manipulirati veličinom ekrana, pozadinskom glazbom, možemo birati težinu 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 veličina ekrana, odnosno težina igrice te konačno, pritiskom na gumb APPLY, sve promjene se pohranjuju te se mijenja veličina ekrana (ukoliko je bila promjenjena).

## 

## 2.6. GameOver.pde

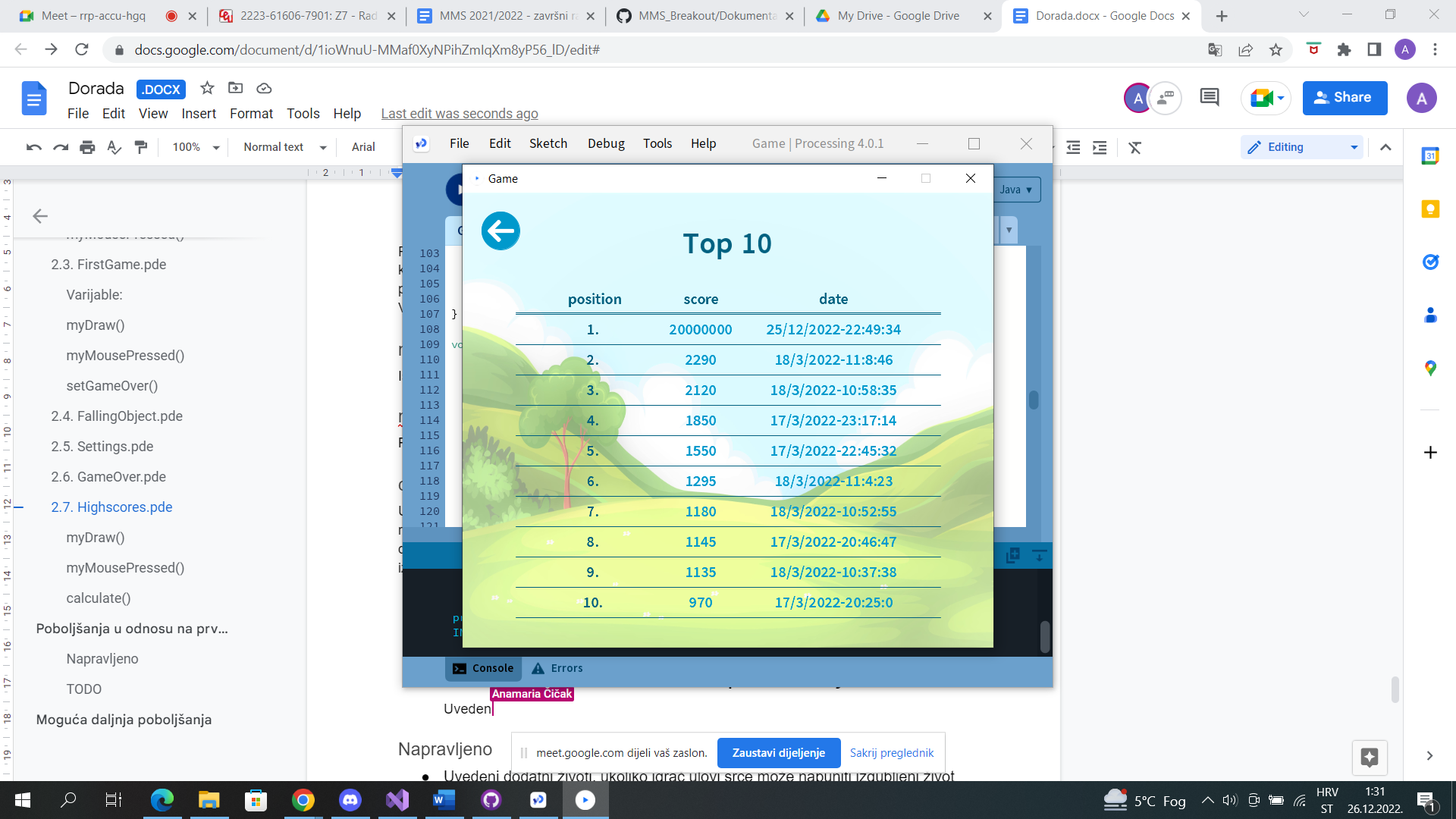
Graphical user interface, website

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.pde



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 varijale homeB i highscoresB.

### calculate()

U polje lines spremaja se 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. Zatim koristeći *bubble sort* sortiramo polja scores i dates silazno po vrijednostima iz polja scores.

# Poboljšanja u odnosu na prvu verziju

Uvedeni su dodatni životi, odnosu 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.

# Moguća daljnja poboljšanja

Kako bi igra bila zanimljivija mogu se implementirati razne podigre, recimo hvatanje samo određenih objekata (npr. iste boje), pomak košare po y osi a objekata po x osi (dakle košara se pomiče gore - dolje, a objekti lijevo - desno). Pri hvatanju objekta možemo dodati poseban zvuk koji označava da je objekt uhvaćen. Korisno bi bilo uvesti mogućnost pauziranja igrice. Uz postojeće objekte mogli bi dodati objekt čije hvatanje povećava košaru ili objekt koji bombe čini “neutralnim” na nekoliko trenutaka, što bi značilo da bi u tim trenucima hvatanje bombe donosilo bodove.