Sveučilište u Zagrebu

Prirodoslovno – matematički fakultet

Diplomski sveučilišni studij Računarstvo i matematika

Akademska godina 2023./2024.

Sandwich Stacker

Završni projekt iz kolegija Multimedijski sustavi

Ivana Sablić, Christian Sandri, Dorotea Žilec

Zagreb, veljača 2024.

Sadržaj

[1.Uvod 2](#_Toc159264555)

[2. Implementacija 3](#_Toc159264556)

[SandwichStacker 3](#_Toc159264557)

[Home 3](#_Toc159264558)

[FirstGame 3](#_Toc159264559)

[GameOver 4](#_Toc159264560)

[Settings 5](#_Toc159264561)

[Highscore 5](#_Toc159264562)

[Highscores 6](#_Toc159264563)

[FallingObject 6](#_Toc159264564)

[3. Dnevnik rada 7](#_Toc159264565)

# 1.Uvod

Za završni projekt iz kolegija Multimedijski sustavi odlučili smo implementirati igricu Sandwich Stacker. Pri dnu ekrana nalazi se donji dio peciva koji se neprekidno kreće lijevo-desno. Na gornjem dijelu ekrana nalazi se dispenzer sastojaka koji se pritiskom na tipke strelice lijevo i strelice desno pomiče lijevo-desno. Pritiskom na tipku space iz njega padaju različiti sastojci.

U svakom trenu je na desnom dijelu ekrana dostupan "info panel" na kojem je prikaz sendviča kakav je potrebno složiti. Također je prikazan broj ostvarenih bodova, broj života te je prikazan sljedeći sastojak koji će pasti iz dispenzera. Cilj igre je složiti sendvič prikazan na info panelu, a jednom kada se to dogodi igrač dobiva bodove. Na početku igre igrač ima 3 života. Ako uhvati sastojak koji nije sljedeći na redu gubi jedan život. Ako ne uhvati sastojak koji sljedeći treba staviti na sendvič također gubi jedan život.

Povremeno se na ekranu može pojaviti power-up koji utječe na razne aspekte igre (brzina pada sastojaka, dodatni bodovi, poništenje jednog promašenog sastojka...). Dodatan kriterij igre je stabilnost sendviča, odnosno postoji mogućnost da se sendvič uruši ako.

Omogućena su dva načina igre. U prvom načinu cijelo vrijeme slažemo istu vrstu sendviča, dok u drugom načinu svaki puta kada složimo sendvič jedne vrste, na info panelu se prikazuje sendvič druge vrste koji je sljedeći potrebno složiti. Omogućen je i odabir težine igre koja će utjecati na brzinu pada sastojaka.

Na početnom izborniku omogućen je i prikaz najboljih 5 rezultata ostvarenih igrajući igru.

2. Implementacija

Igra je implementirana u Processing-u, a sam kod podijeljen je u nekoliko klasa.

Potrebne klase:

* SandwichStacker
* Home
* FirstGame
* GameOver
* Settings
* Highscores
* Highscore
* FallingObject

Opišimo detaljnije svaku od navedenih klasa.

SandwichStacker

Osnovna klasa u projektu je SandwichStacker. Funkcije setup() i init() učitavaju potrebne slike i postavljaju vrijednosti varijabli koje će omogućavati kretanje kroz različite ekrane igrice. draw() ovisno o vrijednostima varijabli prikazuje različite ekrane igrice. Funkcije overCircle() i overRect() provjeravaju nalazi li se pokazivač miša na nekom od gumba koji omogućavaju kretanje kroz igricu, a funkcije mousePressed(), keyPressed(), keyReleased() provjeravaju je li pritisnuta tipka miša ili neka od tipki na tipkovnici te pozivaju potrebne funkcije.

Home

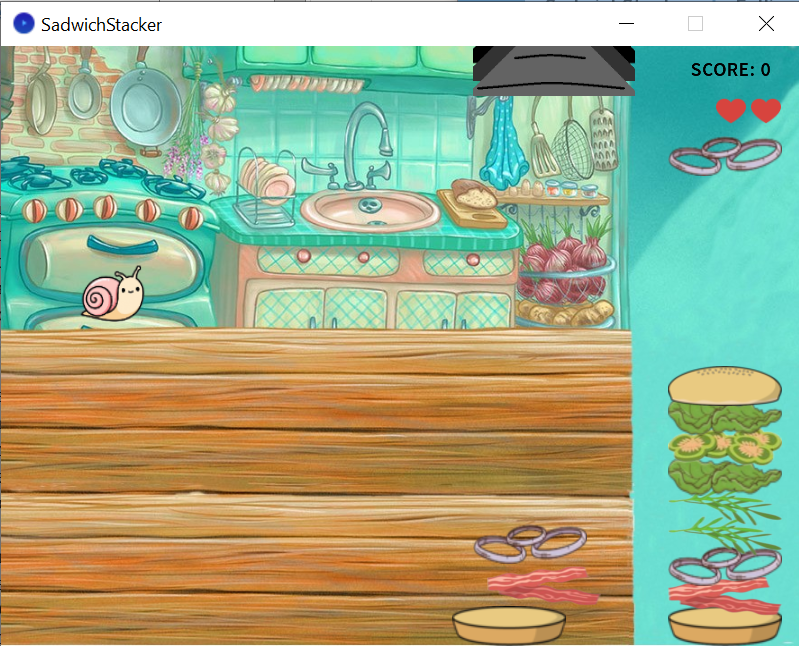
Implementira glavni izbornik igre koji se sastoji od četiri gumba:

* Play
* Settings
* Highscores
* Exit

Klikom na pojedini gumb mijenjaju se vrijednosti boolean varijabli koje se provjeravaju u funkcijama klase SandwichStacker te ovisno o njima prikazuje se odabrani ekran igrice. To je implementirano u funkciji myMousePressed(). Funkcija myDraw() zadužena je prikaz početnog izbornika.

FirstGame

U ovoj je klasi (preciznije u funkciji myDraw()) većina logike same igre. Pomoću varijabli *Tower*, *TowerTopX*, *TowerTopY* pratimo „toranj“ sastojaka na sendviču – na početku je „toranj“ zapravo samo bottomBun. Ovdje se nalazi i logika za neprekidno kretanje donjeg dijela sendviča (*bottomBun*) lijevo-desno implementirana pomoću *orientation* (smjer kretanja – lijevo ili desno) i *speed* (brzina *bottomBun*-a). Varijable *dropIngredient*, *ingredientDropped* i *lastSpaceKeyPressTime* koristimo za onemogućavanje neprestanog pritiskanja tipke space (ne želimo dozvoliti „spammanje“). Težina igre podešava se preko varijabli *difficulty* i *speedIng* (veća težina = veća brzina pada sastojaka). ... Preostali dio funkcije myDraw() bavi se provjerom kolizije vrha *Tower*-a i padajućeg sastojka koja rezultira „lijepljenjem“ tog padajućeg sastojka na vrh *Tower*-a, odnosno kolizije *PowerUp*-a i padajućeg sastojka koja rezultira usporavanjem kretanja *bottomBun*-a (puž), dobivanjem dodatnog života (srce; u svakom trenutku igrač ima maksimalno 3 života) ili dobivanjem dodatnih bodova (zvijezda). U ovom se dijelu provjerava i uvjet urušavanja sendviča preko varijabli *collaps* i *collapsed*.

Osim funkcije ovdje imamo i funkcije setGameOver() (ažurira Highscore ako je potrebno i prikazuje GameOver prozor), myKeyPressed()/myKeyReleased() (kretanje dispensera lijevo-desno pomoću strelica i ispuštanje sastojka pomoću tipke space), generateRandomSandwich() (stvaranje random sendviča čija je visina najmanje 2, a najviše 10 te koji uvijek na dnu ima *bottomBun*, a na vrhu *topBun*), drawGoalSandwich() (crtanje info-panela, odnosno sendviča koji trebamo složiti), generateNewIngredient() (generiranje random sastojka uz veću vjerojatnost generiranja sastojka koji nam je zapravo potreban), nextIngredientInfo() (prikaz sastojka koji će sljedeći pasti na info-panelu), isHighScore() (provjera je li igračev rezultat bolji od najgoreg rezultata na listi najboljih rezultata).

GameOver

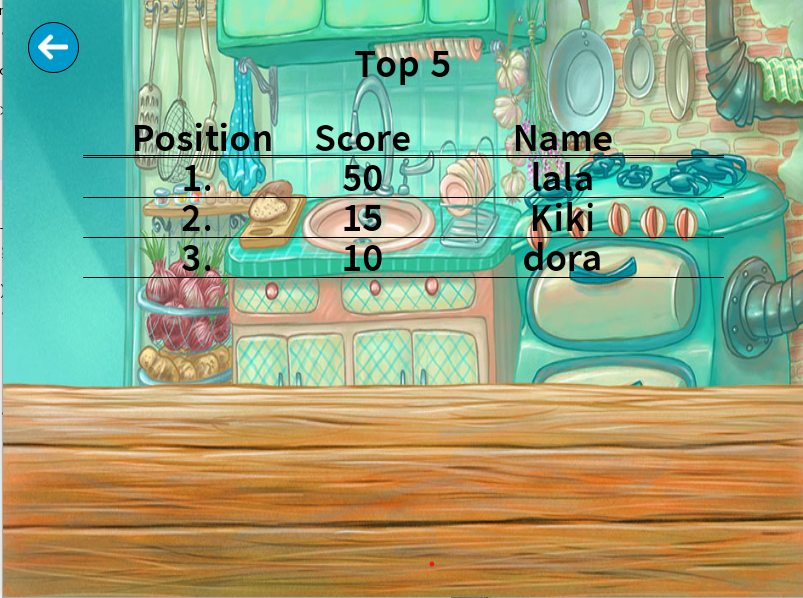
Klasa čiji je primarni zadatak ispis poruke igraču o završetku igre kada igrač ostane bez zadnjeg života. U konstruktoru se inicijalizira varijabla playerName u koju pohranjujemo unos imena igrača te je inicijaliziran textfield u koji će se to ime unositi. Funkcija myDraw() prikazuje poruku o završetku igre te ovisno o tome je li igrač ostvario rezultat koji ulazi u pet najboljih prikazuje se čestitka na ostvarenom rezultatu te textfield i gumb Confirm koji omogućavaju unos imena. Funkcija myMousePressed() provjerava je li igrač kliknuo na neki od ponuđenih gumbova, a to su: gumb za povratak na početni izbornik te gumb za potvrdu unosa imena.

Settings

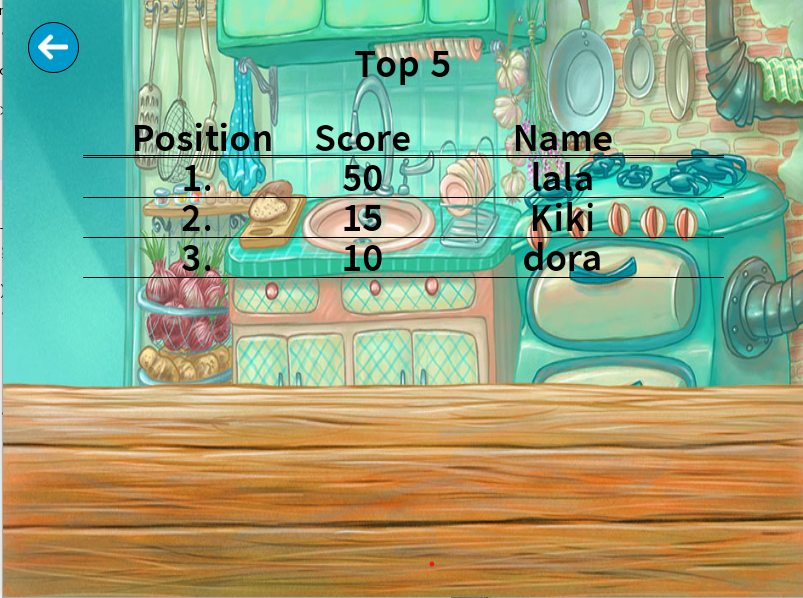
Klasa Settings implementira izbornik načina i težine igranja igrice. Sadrži funkciju myDraw() koja prikazuje gumbe koji omogućuju jednog od dva opisana načina igre, odabir jedne od 4 težine igre te gumb Apply koji odabrano postavlja i primjenjuje u sljedećoj pokrenutoj igri. Zadani način igre je stalno slaganje sendviča iste vrste te je brzina pada sastojka postavljena na Normal. Osim navedenoga, moguće je odabrati način u kojemu svaki puta kada složimo sendvič jedne vrste, dobivamo sendvič druge vrste koji moramo složiti. Za težinu igre možemo birati između Easy, Normal, Hard i Pro, gdje odabirom načina Hard i Pro povećavamo brzinu igre, dok odabirom načina Easy smanjujemo brzinu u odnosu na zadanu brzinu tj. brzinu Normal. Navedena funkcija također prikazuje i gumb za povratak na početni izbornik tj. Home. Funkcija myMousePressed() provjerava je li došlo do klika miša ne neki od navedenih gumbova te pohranjuje napravljene promijene.

Highscore

Highscores je klasa čija je primarna zadaća prikaz najboljih ostvarenih rezultata. Rezultate dohvaća funkcijom show() te ih sprema u dvije odvojene liste scores i names. Funkcija myDraw() zadužena je za prikaz učitanih informacija igraču, dok funkcija myMousePressed() provjerava je li igrač kliknuo na gumb za povratak na početni izbornik.



Highscores

Highscores je klasa čija je primarna zadaća prikaz najboljih ostvarenih rezultata. Rezultate dohvaća funkcijom show() te ih sprema u dvije odvojene liste scores i names. Funkcija myDraw() zadužena je za prikaz učitanih informacija igraču, dok funkcija myMousePressed() provjerava je li igrač kliknuo na gumb za povratak na početni izbornik.

FallingObject

Reprezentira padajući objekt tako da pamti njegovu širinu, visinu, koordinate, indeks (označava naziv same slike u folder-u data) i redniBroj (varijabla potrebna za urušavanje sendviča tj. za prepoznavanje koji od trenutno prisutnih objekata na ekranu pripadaju sendviču, a koji ne).

3. Dnevnik rada



4. Literatura

* <https://github.com/MatejaT4/Catch>
* <https://github.com/sojamo/controlp5>