

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

MOODLE JUMP

Završni projekt iz kolegija Multimedijски sustavi

Članovi tima: Mislav Kocijan, Vlado Bilogrević, Margarita Tolja, Tihana Oremuš

Voditelj kolegija: Goran Igaly

Akadska godina: 2019./2020.

UVOD

Za projekt iz kolegija Multimedijски sustavi odlučili smo razviti platformsku igru u Processing programskom okruženju. Željeli smo izraditi vlastitu verziju već poznate igre Doodle jump, u kojoj je cilj dovesti igrača što je moguće više na beskonačnom nizu platformi. Na putu se pojavljuju razni objekti, neki mu daju određeni poticaj, dok neke treba izbjegavati ili eliminirati. S našom verzijom ćemo se više upoznati u sljedećem odlomku.

FUNKCIONALNOST IGRE

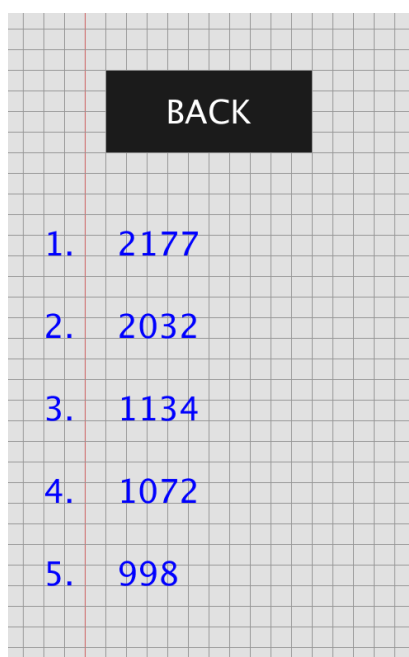
Pri otvaranju igre, pojavljuje se početni meni na kojem su navedene upute za igru, te se bira jedan od dva moguća izbora: pregledaj listu najboljih rezultata ili započni novu igru.

Odabirom na *highscores* prikazuje se popis prvih 5 najboljih rezultata te se nudi opcija povratka na početni meni klikom na gumb *back*.

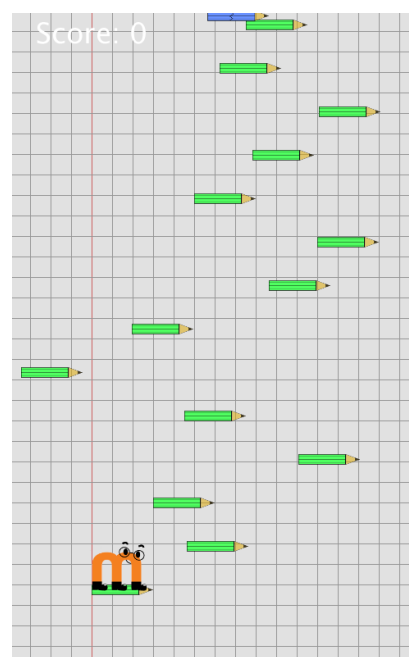
Odabirom na *start* počinje nova igra.



Početni meni



Highscores



Početak igre

Igrač je skakajuće stvorenje zvano *The Moodler*,  kojem je cilj, odskakujući od platformi, doseći što veću visinu. S visinom se platforme prorjeđuju te se više pojavljuju one s kojima se teže popeti. Postoje četiri vrste platformi, od kojih svaka ima drugačiji izgled i svojstva:

1. Obične platforme

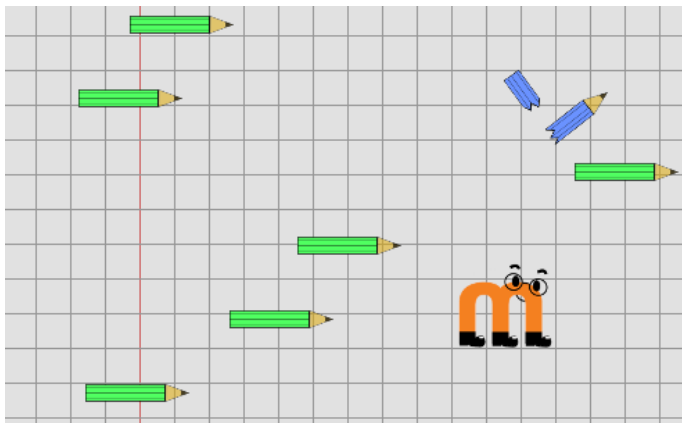


Igrač odskakuje od zelene platforme bez ikakvih posljedica.

2. Slomljene platforme



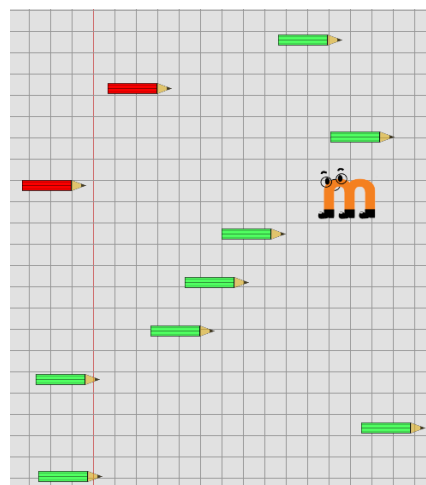
Slijetanjem na plavu platformu, ona se slama i igrač propada kroz nju.



3. Pomične platforme



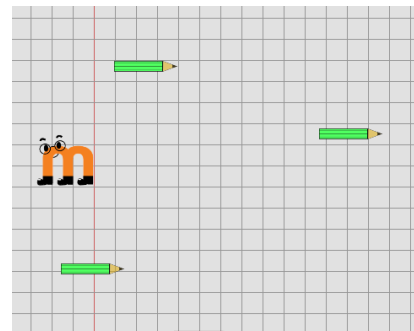
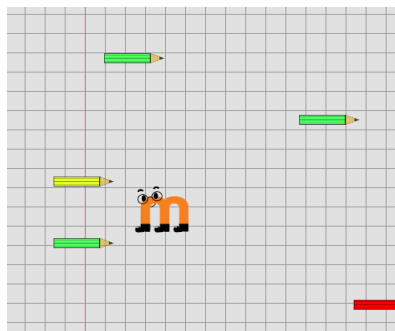
Crvene platforme se kreću po horizontalnoj liniji udesno te nakon što izađu iz prozora s desne strane, ponovo se pojavljuju na lijevoj strani.



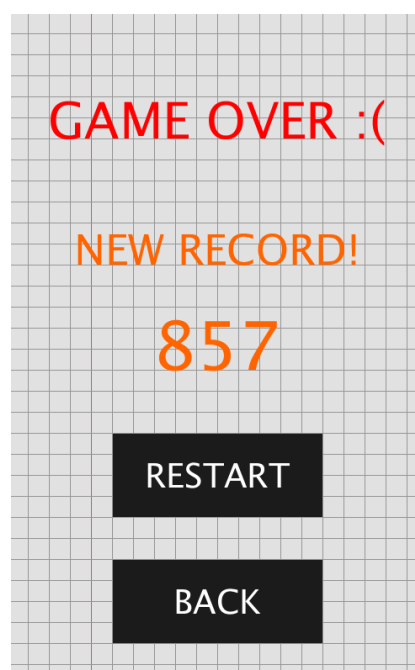
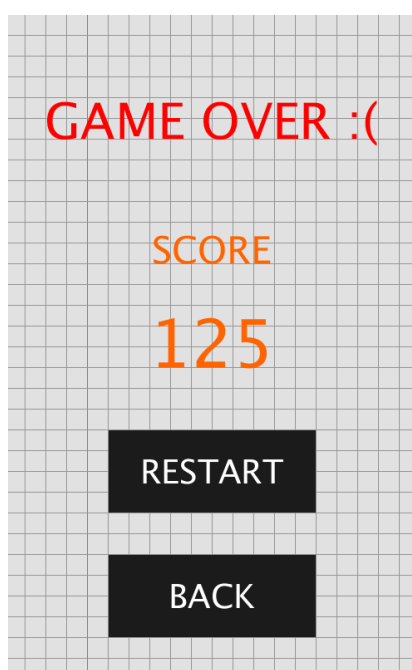
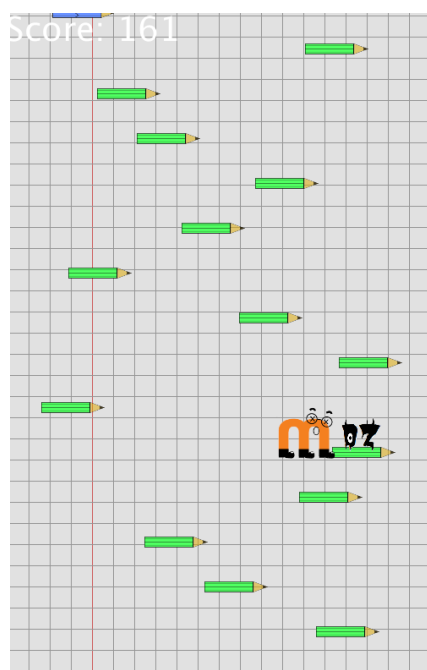
4. Nestajuće platforme



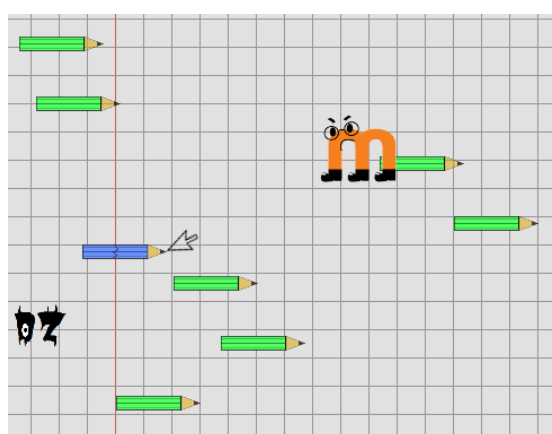
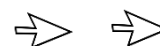
Od žute platforme igrač može jednom odskočiti prije nego što ona nestane.



Igra završava ako Moodler ne uspije sletjeti ni na jednu platformu prije nego što padne preko donje granice prozora. Također, Moodler može umrijeti ukoliko se sudari s jednim od „čudovišta“ koji mu otežavaju put. Postoje dvije vrste čudovišta, to su **0Z** (zadaca) i **SEMINAR** (seminar). Pri završetku igre, prikazuje se odgovarajući meni gdje se nudi opcija povratka na početni meni ili pokretanje nove igre. Također se ispisuje ostvareni rezultat te javlja je li on najbolji dosad.



Moodler se može riješiti čudovišta ispaljivanjem metaka u njihovom smjeru. Zadaca nestaje nakon jednog pogotka metkom, a seminar tek nakon dva.



Moodler odskakuje u fiksnim vremenskim razmacima, te se može kretati ulijevo i udesno. Ako prijeđe granicu prozora, ne sudara se, nego prelazi na drugu stranu. Osim tih osnovnih sposobnosti kretanja, Moodleru pomažu i moći koje skuplja na putu. Kada se neka moć pojavi na platformi, Moodler ju može pokupiti tako da skoči na tu platformu.



Federi pomažu Moodleru da skače više.



Dok Moodler ima *štit*, *seminar* i *zadaca* ga ne mogu ubiti.



Nakon što pokupi *propeler*, Moodler preleti određeni put, bez skakanja, te ga *seminar* i *zadaca* također ne mogu ubiti.

Svaka moć ima vrijeme trajanja, u slučaju federa i štita, to vrijeme iznosi 6 sekundi, a propeler traje 3 sekunde. Također treba napomenuti da dok jedna moć traje, Moodler nije u mogućnosti pokupiti drugu moć.

Sada je opisano kako se svi dijelovi igre ponašaju, pa prelazimo na opis njihove implementacije.

IMPLEMENTACIJA

GRAFIKA I ZVUK

Svi spriteovi Moodlera, platformi i supermoći korišteni u igri napravljeni su pomoću Gimp-a i Inkscapea. Čudovišta seminar i zadaca su napisani u Google fontu Butcherman-Regular.

Glazba u početnom meniju uzeta je s youtubea, dok su ona koja traje tijekom same igre, te ona koja reprezentira kraj igre napravljene pomoću programa MuseScore 3.

PROGRAM

Program je podijeljen na glavni program i dijelove koje implementiraju funkcionalnosti igrača, platformi, čudovišta i metaka. Ukratko ćemo po dijelovima opisati svaku klasu i njezine funkcije.

platform_class

Implementira platforme, sadrži klasu Platform i podklase Regular_Platform, Moving_Platform, Disappearing_Platform i Broken_Platform.

Nadklasa Platform sadrži:

- *float x_pos, y_pos* - varijable koordinata
- *float p_length* - duljina platforme
- *PImage slikapl, power* - sprite platforme i moći na njoj
- *String superpower* - "", "federi", "stit" ili "propela"
- *boolean visited* - True ako je igrač skočio na nju
- *public Platform(float x, float y, String spower)* - konstruktor koji postavlja početne rijednosti varijabli i u slučaju da se neka supermoć pojavljuje na platformi, učitava sprite te supermoći
- *public void display()* - koja postavlja sprite platforme i supermoći na platformu
- *public void destroy()* - funkcija koja uništava platformu
- *public float get_x()* - funkcija koja vraća varijablu x_pos
- *public float get_y()* - funkcija koja vraća varijablu y_pos
- *public String get_power()* - funkcija koja vraća varijablu superpower
- *abstract void update()*

Klasa Moving_Platform:

- *private float x_velocity* - varijabla koja predstavlja brzinu kretanja platforme
- *public Moving_Platform(float x, float y, String spower)* - konstruktor koji učitava sprite pomicne platforme
- *public void update()* - ažurira x koordinatu platformu

Klasa Disappearing_Platform:

- *public Disappearing_Platform(float x, float y, String spower)* - konstruktor koji učitava sprite nestajuće platforme
- *public void update()*

Klasa Broken_Platform:

- *private PImage slikapl0, slikapl1, slikapl2, slikapl3, slikapl4, slikapl5* - spriteovi koji predstavljaju razine slamanja platforme
- *private boolean broken* - True ako je platforma slomljena
- *public Broken_Platform(float x, float y)* - konstruktor koji učitava spriteove slomljene platforme
- *public void display()* - u slučaju da je proces slamanje počeo, postavlja iduću sliku slomljene platforme
- *public void update()*
- *public void break_pl()* - postavlja varijablu broken na True

player_class

Implementira igrača i njegove funkcionalnosti.

- *enum Orientation* - smjer igrača, LEFT ili RIGHT
- *enum State* - predstavlja stanje u kojem se Moodler nalazi, ovisno o tome ima li supermoć ili ispaljuje metke

Klasa Player:

- *private float x, y, x_velocity, y_velocity, gravity, size* - koordinate, brzine, gravitacija, veličina igrača
- *private float climb* - koliko se platformi popeo igrač
- *public int score* - postignuti rezultat
- *private int sp_duration, propeller_duration, t_start* - varijable koje se koriste pri upravljanju trajanja neke supermoći
- *private Orientation orientation* - orijentacija igrača
- *private State state* - stanje igrača
- *public Player(int x, int y, int x_velocity, int y_velocity, int size)* - konstruktor koji postavlja početne vrijednosti varijabli
- *public float get_x()* - vraća koordinatu x
- *public float get_y()* - vraća koordinatu y
- *public void update(ArrayList<Platform> platforms, ArrayList<Broken_Platform> broken_platforms, MyFloat first_horiz_line)* - ažurira stanje igrača, provjerava kolizije, pomiče igrača, pomiče okruženje oko njega
- *public void display()* - ovisno o stanju, postavlja sprite igrača
- *public void move()* - ovisno o kliku tipkovnice, postavlja orijentaciju i položaj, ažurira gravitaciju i brzinu
- *public void move_world(ArrayList<Platform> platforms, ArrayList<Broken_Platform> broken_platforms, MyFloat first_horiz_line)* - ovisno o visini na koju se igrač popeo, ažurira score, ovisno o položaju igrača, pomiče okruženje, ažurira položaj čudovišta
- *public void check_collision(ArrayList<Platform> platforms, ArrayList<Broken_Platform> broken_platforms)* - upravlja sudarima između igrača i platformi/čudovišta/supermoći ovisno o tome je li igrač mrtav ili ima li već neku moć
- *public void update_state()* - ažurira stanje igrača ovisno o tome ima li supermoć i je li istekla

evil_stuff

Implementira čudovišta i putanje po kojima se kreću. Sardži klasu path2 i podklasu path2ellipse, te klasu homework.

Klasa homework

- *int type* - 0 za zadaću, 1 za seminar
- *float x_pos, y_pos* - koordinate
- *float x_size[]={0,0}, y_size[]={0,0}* - veličina
- *PImage HWImage[]={null, null}* - sprite čudovišta
- *int health* - 1 za zadaću, 2 za seminar, koliko metaka je potrebno da ga uništi
- *int i = 0* - pomoćna varijabla
- *float x_velocity, y_velocity* - brzina
- *path2 HWpath[]* - putanja po kojoj se kreće
- *private Direction direction=Direction.LEFT* - smjer kretanja
- *public homework(float x, float y)* - konstruktor koji postavlja početne vrijednosti varijabli

- *public void display()* - postavlja sprite čudovišta ako nije uništeno
- *public void update()* - provjerava je li ga pogodio metak i je li izašao iz prozora
- *private void makeHW(int t)* - na slučajan način bira hoće li iduće čudovište biti zadaća ili seminar
- *float rect_intersect(float x11, float y11, float x12, float y12, float x21, float y21, float x22, float y22)* - provjerava kolizije

bullet_class

- *enum Direction* - smjer ispaljivanja metka, LEFT ili RIGHT

Klasa Bullet

- *private float x_pos, y_pos* - koordinate
- *private float angle* - kut ispaljivanja metka
- *private Direction direction* - smjer
- *private boolean bingo* - true ako je metak pogodio čudovište
- *public Bullet(float x_click, float y_click)* - konstruktor koji postavlja početne vrijednosti varijabli
- *public float get_x()* - vraća x koordinatu
- *public float get_y()* - vraća y koordinatu
- *public void display()* - ažurira položaj metaka
- *public void update()* - metak se ne prikazuje ako je pogodio metu

moodle_jump

Glavni program za Moodle jump, u globalnim varijablama se učitavaju potrebne datoteke poput biblioteka, glazbe, deklariraju se liste platformi, spritova grača, metaka i inicijaliziraju se pomoćne varijable.

Funkcije koje program sadrži:

- *void setup()* - konstrukcija svih potrebnih elemenata
- *void draw_background()*
- *void draw()* - u ovisnosti o globalnoj varijabli state (trenutno stanje igre), prikazuje se lista najboljih rezultata, glavni meni ili meni završetka igre
- *void mousePressed()* - klikom miša se ispaljuje metak
- *void remove_bullets()* - uništava metke ako izađu iz prozora
- *void add_remove_platforms()* - određuje vjerojatnost da se pojavi nova platforma ili supermoć, uništava one platforme koje su izašle iz prozora
- *void reset()* - u slučaju da igrač umre, resetira sve varijable

MOGUĆA POBOLJŠANJA

Igra bi se mogla poboljšati na razne načine, neki od primjera su:

- prilagođavanje za Android sustave
- poboljšanje vjerojatnosti pojavljivanja novih platformi i čudovišta
- dodavanje novih čudovišta
- određivanje razina igre, te prikazivanje trenutne razine na ekranu
- dodavanje mogućnosti da se prije početka igre bira težina igre