

UNIVERZITET U SARAJEVU



ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA RAČUNARSTVO I INFORMATIKU

Projekat

Space race

Asim Ćeman, Adin Heco, Baždar Faris

Predmet: Ugradbeni sistemi

Akadska godina: 2019/2020

Sarajevo, Juli 2020.

Za naš projektni zadatak, igricu Space Race, razvit ćemo određeni broj klasa, kako bi olakšali implementaciju. Projekat sadrži sljedeće klase:

1. Brod

2. Debris

[Baždar Faris]

Brod(int x); Inicijaliziramo atribut broda(x,y,score,r,vel), te ga spawnamo. Atributi x i y predstavljaju centar broda, atribut score predstavlja postignuti broj poena tog igrača, r je poluprečnik koji obuhvata brod, vel je pomoćna varijabla za iscrtavanje broda.

respawn(); Postavlja y koordinatu broda na dno ekrana.

gore(); Pomjera brod po y - koordinati ka vrhu ekrana.

dole(); Pomjera brod po y - koordinati ka dnu ekrana.

addingPoint(); Povećava postignuti broj poena.

display(); Crta brod na ekranu sa njegovim pripadajućim x,y koordinatama.

Dajscore(); Vraća vrijednost poena.

resetujScore(); Postavlja postignuti broj poena na 0.

Dajx(); Vraća vrijednost x-a.

Dajy(); Vraća vrijednost y-a.

DajR(); Vraća vrijednost R-a.

[Heco Adin]

Debris(); Konstruktor bez parametara, postavlja vrijednost poluprečnika na 9, i resetuje debris.

resetDebris(); Resetuje debris i određuje sa koje strane se pojavljuju neprijateljski brodovi.

update(); Ažurira položaj neprijateljskom brodu.

isOffScreen(); Provjerava da li je brod van ekrana.

Display(); Crta brod na ekranu sa njegovim pripadajućim x,y koordinatama.

udariloBrod(Brod b); Provjera da li su neprijateljski i prijateljski brod u sudaru.

upute(); Prikazuje početni izgled ekrana zajedno sa uputama za igranje igrice.

Pored ovih klasa imali smo pomoćne funkcije.

[Ćeman Asim]

distance(int x1, int y1, int x2, int y2); Funkcija prima kao argument dvije tačke, i vraća udaljenost između njih.

randomBool(); Vraća random vrijednost.

void updateDebrisAndCheckCollisions(std::vector<Debris*> allDebris, Brod &leftShip, Brod &rightShip); Funkcija koja poziva metodu iz debris klase update() i provjerava da li je došlo do sudara između igrača i neprijateljskih brodova.

crtanjeVatre(int x, int y, bool prvak, int a=9); Crta vatru ako je došlo do sudara.

level(); Funkcija koja određuje koliko će se nalaziti neprijateljskih brodova na osnovu

pritisnutog tastera.

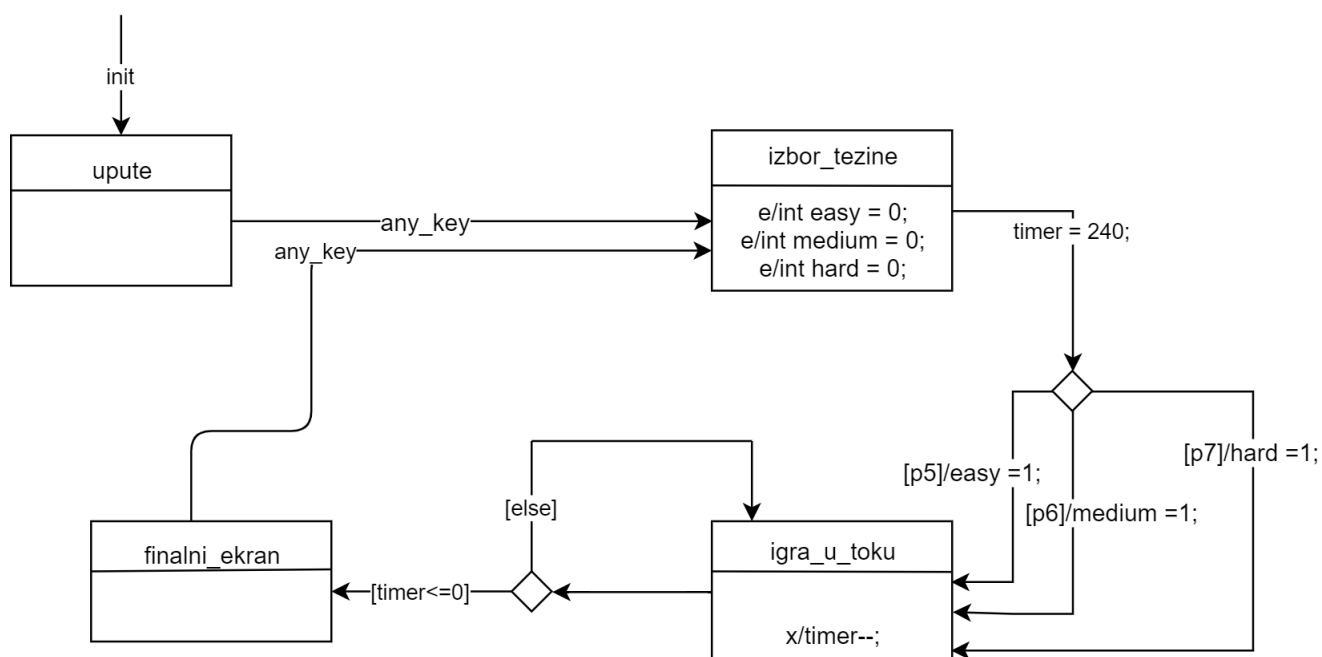
formirajDebris(int NUM_DEBRIS); Formira vektor debrisa, čiji broj elemenata zavisi od izabrane težine.

zvijezde(std::vector<std::pair<int,int> > v); Iscrtava zvijezde na ekranu na osnovu proslijeđenog vektora parova.

Pehar(int p); Crta pehar za pobjednika koji se šalje u argumentu i ispisuje poruku na ekran.

[Baždar Faris i Ćeman Asim]

main(); U main funkciji se izvršava igra, ona poziva sve navedene metode i klase.



Slika 0.1: Dijagram stanja i prelaza.

Implementacija:

U početku se nalazimo u stanju *upute* koje na ekranu ispisuju način igranja igrice, nakon što pritisnemo bilo koji od četiri ponuđena tastera prelazimo u stanje *izbor_tezine* gdje pritiskom odgovarajućeg tastera biramo težinu (broj neprijateljskih brodova). Nakon što izaberemo težinu shodno uputama na ekranu igra se započinje sa početnim tajmerom od 240s. Sve dok ne istekne vrijeme nalazimo se u istom stanju (*igra_u_toku*). Pri isteku vremena prelazimo u stanje *finalni_ekran* koje nas obavještava ko je pobjednik, te daje mogućnost da igramo opet što nas vraća u stanje *izbor_tezine*.