Sveučilište u Rijeci – Odjel za informatiku

Preddiplomski jednopredmetni studij informatike

SNAKE GAME

Objektno programiranje

Student: Antonio Suljić

Mentor: doc. dr. sc. Miran Pobar

U Rijeci, 17. ožujka 2022.

SADRŽAJ

1. UVOD1
2. STRUKTURA PROGRAMA I DIJAGRAMI2
   1. Dijagram klasa3
   2. Dijagram aktivnosti4
3. UPUTE ZA *BUILDANJE* PROGRAMA5
4. UPUTE ZA KORIŠTENJE6
5. ZAKLJUČAK7
6. LITERATURA8
7. UVOD

Ideja “*Snake Game*” projekta je napraviti igricu u kojoj je cilj igrača doseći što veću dužinu zmije jedenjem voćki te u isto vrijeme paziti da zmija ne ugrize samu sebe. U konzoli zmija je predstavljena slovom ‘O’ dok je voćka predstavljena sa slovom ‘Q’. Rub je definiran na način da zmija može ‘izaći’ kroz jedan rub te ‘ući’ u konzolu kroz nasuprotni rub. Igrica ima neke osnovne mogućnosti kao što je mogućnost da zmija ugrize samu sebe te time završi igra - “*Game Over*”, izbornik prije pokretanja glavne igrice. Neke od dodatnih implementacija uključuju praćenje rezultata ‘*Score*’ u gornjem lijevom kutu, blinkanje voćke te mogućnost promjene težine, odnosno, brzine tokom igranja igrice.

1. STRUKTURA PROGRAMA I DIJAGRAMI

Program se sastoji od klase Element(“*element.cpp*”), klase Snake(“*snake.cpp*”), globalnih funkcija gotoxy i setcursor(“*define.cpp*”) te main funkcije(“*game.cpp*”). Struktura je organizirana na način da su klase i globalne funkcije podijeljene u source i header datoteke koje se koriste u glavnoj “*game.cpp*” source datoteci.

Slika na kojoj se prikazuje tekst, snimka zaslona, telefon, mobitel

Opis je automatski generiran

Figure Struktura programa

Klasa Element predstavlja dijelove zmije, dakle to je tip podatka kojim je definiran vector zmije – klasa Element definirana je pomoću koordinatnih točaka zato što je svrha te klase postavljanje točaka koje predstavljaju mjesto u konzoli na kojem će program crtati dio zmije ili voćku. Dok klasa Snake predstavlja zmiju kao cjelinu, što znači da tu definiramo koji su atributi zmije – to može biti njena veličina, stanje i sl. Globalna funkcija gotoxy pomoću “<*windows.h>*” biblioteke omogućava programu postavljanje kursora na novu poziciju dok setcursor globalna funkcija postavlja velicinu i vidljivost kursora u konzoli. U main funkciji iskorištavamo povezane funkcionalnosti klasa i globalnih funkcija te definiramo igru, odnosno, primjenjujemo njeno izvođenje.

Biblioteke korištene u programu su - conio.h, windows.h, dos.h, time.h.

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Figure Biblioteke

Conio biblioteka je korištena zbog kbhit i getch funkcija koje omogućavaju čekanje *user input-a* i registriranje tipke koju je korisnik pritisnuo. Windows biblioteka poslužila je za definiranje prethodno opisanih globalnih funkcija pomoću *HANDLE i COORD* tipova podataka. Dos biblioteka omogućila nam je funkcionalnost Sleep metode unutar Snake klase dok je Time biblioteka pružila manipulaciju time i date što znatno pomaže u kreiranju *SEED-a* te zahvaljujući njemu voćke nemaju isti *random pattern* već je svaki put drugačiji.

1. . Dijagram klasa

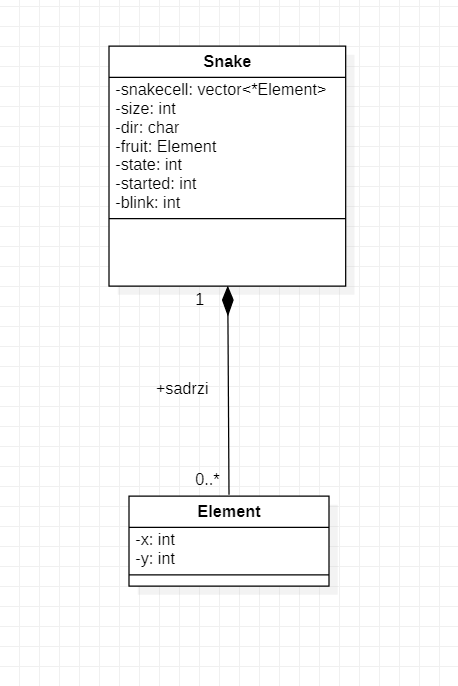
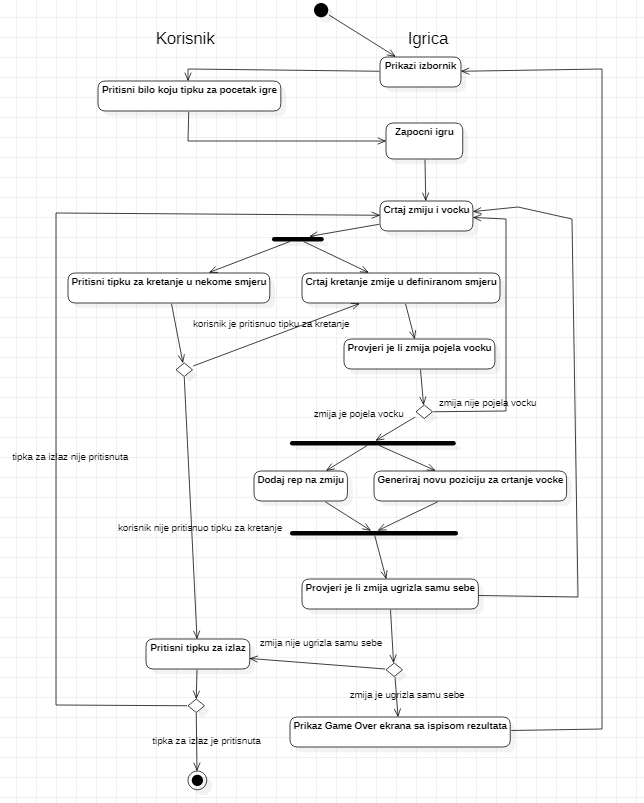


Figure Dijagram klasa

U strukturi imamo dvije prethodno opisane klase Snake i Element. Definirani su atributi koje posjeduju kao i u kodu te njihova brojnost koja se može opisati kao “Jedan Snake može sadržavati više Elemenata” pa prema tome je definirana kao što je prikazano na slici.

1. . Dijagram aktivnosti



Dijagram aktivnosti opisan je pomoću ‘plivaćih traka’ gdje lijeva predstavlja korisnika a desna igricu kao program. Aktivnost korištenja igrice započela bi u izborniku koji korisnika traži da pritisne bilo koju tipku za početak igranja, zatim započnemo igru te program korisniku ispisuje poziciju zmije i voćke pri čemu se očekuje od korisnika da pritisne tipku kako bi zmija znala u kojem smjeru krenuti. Nakon toga, program crta kretanje zmije u zadanom smjeru te provjerava je li zmija pojela voćku, ako je to istina onda program zmiji dodaje rep i generira novu poziciju voćke, a ukoliko je to laž onda program nastavlja crtati zmiju i voćku bez promjena. Zatim, provjeravamo je li zmija ugrizla samu sebe gdje u slučaju istine dobivamo prikaz ‘*Game Over*’ekrana, a u slučaju laži ništa se ne događa, već očekujemo hoće li korisnik pritisnuti tipku za izlaz iz programa – ako neće onda se vraćamo na crtanje zmije i voćke.

1. UPUTE ZA *BUILDANJE* PROGRAMA

Za development projekta korišten je Visual Studio Code te g++ compiler. Dakle, projekt se *builda* na način da se prvo navigiramo u direktorij preko terminala te komandom



Figure komanda za build u game.exe

*buildamo* project sa outputom u game.exe datoteci. Nakon toga možemo pokrenuti game.exe datoteku te bi se trebao pojaviti početni ekran kao na slici.

Slika na kojoj se prikazuje tekst

Opis je automatski generiran

Figure Pocetni ekran

U slučaju bilo kakvih problema link na repozitorij: https://github.com/sukyca/snake\_game.git

1. UPUTE ZA KORIŠTENJE

Nakon pokretanja game.exe datoteke te prikaza početnog ekrana – možete pogledati popis kontrola u konzoli. Zatim, pritisnite bilo koju tipku na tipkovnici za početak igre. Nakon što se igra pokrene krećete se sa ‘w’, ‘s’, ‘a’, ‘d’ tipkama koje nije potrebno držati već je dovoljno stisnuti jednom da zmija promijeni smjer. Pritiskom na tipke ‘h’, ‘j’, ‘k’ mijenjate težinu igre gdje je h za ‘*easy mode*’, j za ‘*normal mode*’, k za ‘*hard mode*’. Težinu možete mijenjati usred igre te pri promijeni težine očekivano ponašanje je da zmija ubrza.   
Napomena – preporuka je da se konzola ne povećava već da se igra u default veličini prozora zbog definicije rubova inače može doći do neočekivanog ponašanja zmije.

1. ZAKLJUČAK

Projekt Snake Game izrazito mi je pomogao u nadograđivanju logike objektno-orijentiranog programiranja te sam se prvi put susreo sa realnim razmišljanjem o objektu u igrici – u ovom slučaju – zmija. Također, naučio sam puno o definiciji same konzole i proučavanju meni novih biblioteka kao što je windows.h. Zadovoljan sam izazovima na koje sam naišao stvarajući programsko riješenje za moju prvu igricu.

1. LITERATURA