

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

RAČUNALNA GRAFIKA

3. laboratorijska vježba

Računalna igra Terminal V

Nikola Šunjo

0036488018

Zagreb, siječanj, 2021.

Sadržaj

1. Uvod.....	1
2. Implementacija igre	2
2.1 Dizajn scene.....	2
2.2 Kretanje igrača	3
2.3 Interakcija s objektima.....	3
2.4 Dizajn zagonetki.....	6
2.5 Izbornici, grafičko sučelje i završetak igre	7
2.6 Upute za korištenje	8
3. Zaključak	9
4. Literatura	10

1. Uvod

Terminal V je kratka računalna igra stvorena u razvojnem okruženju *Unity Engine* [1]. Igra je dizajnirana kao *room escape* igra u kojoj je cilj riješiti nekoliko zagonetaka te skupiti pet ključeva unutar dvadeset minuta. Svaka zagonetka se rješava na različit način od drugih, a rješenje može otkriti lokaciju ključa ili lozinku za otključavanje vrata. Ključevi se nalaze na ili ispod mnogih interaktabilnih objekata dostupnih u igri. Za uspješan prelazak igre potrebno je riješiti sve zagonetke, otključati sva vrata te skupiti sve ključeve.

Motivacija za odabir ove teme je veliko zanimanje za dizajniranje i izradu video igara, kao i prilika za dodatno učenje i usavršavanje rada u *Unityu*, mogućnosti izrade cjelovite igre, od početka do kraja, bez obzira na to što se radi o jednostavnoj i kratkoj igri te potencijalnom učenju novih stvari u procesu izrade video igara.

2. Implementacija igre

2.1 Dizajn scene

Igra se sastoji od dvije scene, glavnog izbornika te scene u kojoj se odvija igra. Scena je dizajnirana kao manji, moderno uređen stan koji se sastoji od četiri sobe. U svakoj sobi nalaze se razni objekti s pripadajućim teksturama i materijalima [2]. Objekti su podijeljeni na dvije vrste, statične i one objekte s kojima je moguće rukovati, odnosno primiti ih. Velika većina objekata su besplatni *asseti* preuzeti s Unity Asset Storea [3]. Osvjetljenje scene ostvareno je primjenom svjetlosnih sonda (eng. *light probe*) zajedno s reflektivnim sondama (eng. *reflection probe*) s namjenom stvaranja što je bliže moguće prirodnog, realnog osvjetljenja stana. Radi boljeg izvođenja i optimizacije osvjetljenje je ugrađeno (eng. *baked*) u scenu tako da prilikom pokretanja i izvođenja nije potrebno konstantno izvoditi matematičke operacije računanja za stvaranje osvjetljenja i sjena u sceni [4]. U scenu je dodan i *skybox* koji prikazuje lokaciju u Düsseldorfu, a služi za poboljšanje ugođaja tijekom igranja, tj. radi stvaranja dojma da se igrač nalazi na stvarnoj lokaciji.



Slika 1. Dizajn scene

2.2 Kretanje igrača

Kretanje igrača ostvareno je pomoću dvije skripte zajedno s *Character Controller* komponentom koja dodana na objekt igrača. *Character Controller* komponenta definira dimenzije igrača, tj. njegovu visinu i radijus (širinu). Od već spomenute dvije skripte, jedna je zadužena kretanje igrača u kontekstu scene po x, y i z koordinatama, a druga definira rotaciju i pomicanje kamere pokretom miša. Skripta za rotaciju prima koordinatu pokreta miša pomnoženu s vremenom te na temelju toga korištenjem kvaterniona izračuna i izvrši rotaciju igrača. U skripti postoji ograničenje od 90 stupnjeva na y koordinati radi simulacije stvarnog pokreta glave. Druga skripta radi na sličan način, u njoj je definira brzina kretanja igrača te na klik tipke na tipkovnicu korištenjem te informacije i proteklog vremena transformira lokaciju igrača po odgovarajućoj koordinati. Za kretanje se koriste uobičajene tipke „WASD” koje su tako definirane prilikom stvaranja svakog projekta u Unityu, a moguće ih je promijeniti u postavkama projekta. Također, dodana je mogućnost čučnja pritiskom tipke na tipkovnici što je ostvareno smanjenjem visine igrača.

```
float x = Input.GetAxis("Horizontal");
float z = Input.GetAxis("Vertical");

Vector3 move = transform.right * x + transform.forward * z;

controller.Move(move * speed * Time.deltaTime);

velocity.y += gravity * Time.deltaTime;

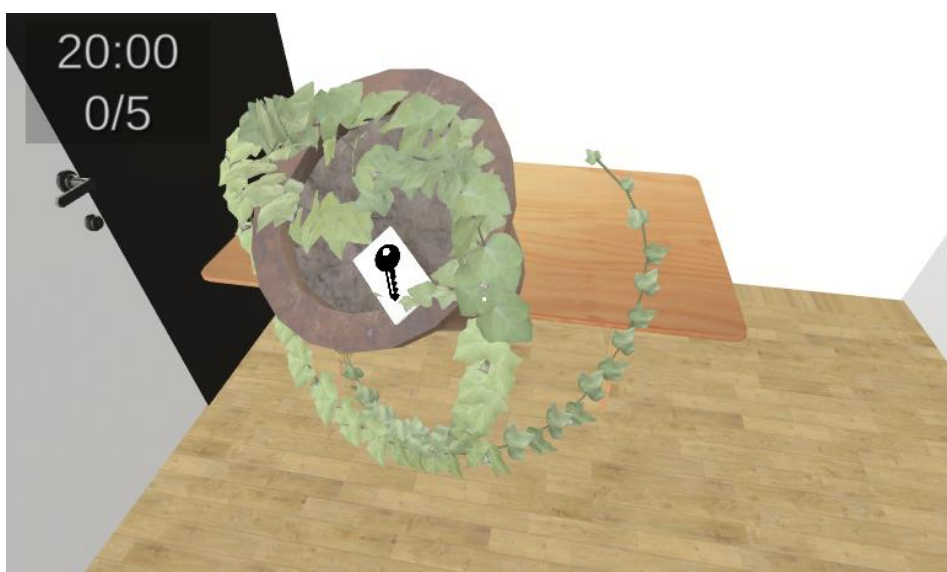
controller.Move(velocity * Time.deltaTime);
```

Slika 2. Isječak skripte za transformiranje lokacije igrača

2.3 Interakcija s objektima

Kao što je već spomenuto u prvom potpoglavlju, objekti u sceni podijeljeni su na statične i interaktabilne. Razlikuju se u tome što statične objekte igrač ne može primiti, rotirati i nositi po sceni. Statični objekti su većinom veći objekti u sceni poput stola, frižidera, kuhinjskih elemenata i kreveta, odnosno to su objekti za koje nije

realno očekivati da ih čovjek može lako primiti i nositi. Interaktabilni objekti se unutar *Unity inspector*a razlikuju u dvije komponente koje statični nemaju, *Rigidbody* i prazna skripta „Pickupable“. *Rigidbody* definira masu objekta i omogućuje korištenje gravitacije prilikom interakcije s objektom što je potrebno u slučaju kada igrač treba ispustiti objekt koje trenutno nosi. Prazna skripta „Pickupable“ služi kao znak da je s tim objektom moguće rukovati, a to se prepoznaje unutar skripte zadužene primanje, nošenje, rotaciju i ispuštanje objekta te skupljanje ključa.



Slika 3. Interakcija s objektom

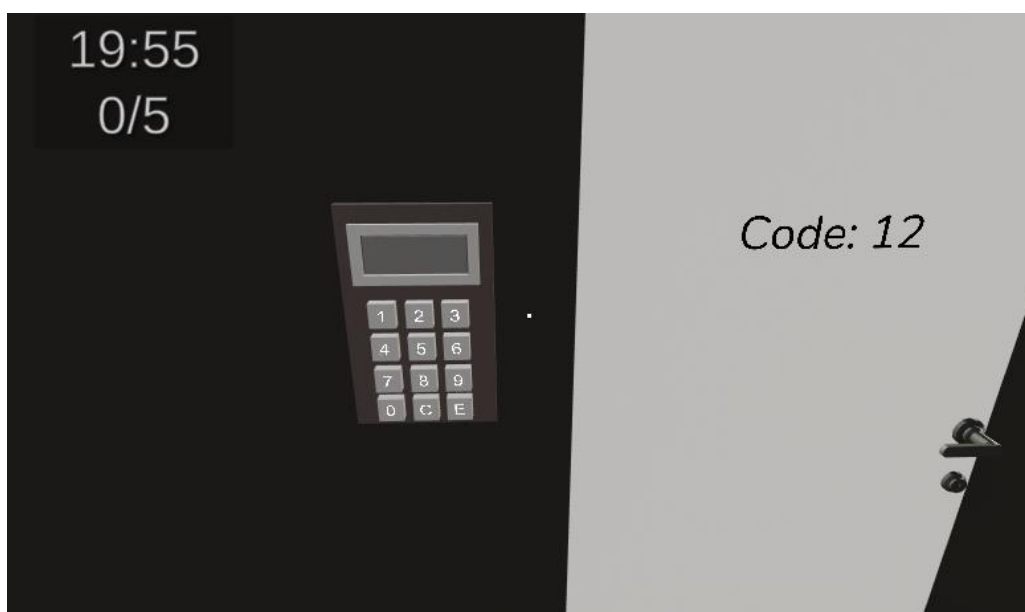
Provjera je li objekt moguće primiti nakon pritiska tipke „E“, pod uvjetom da je to objekt sa skriptom „Pickupable“, se odvija projekcijom zrake [5] (eng. *raycast*) iz središta definiranog s *crosshair*om (mala bijela točka) na objekt i tada, ako su svi uvjeti izvršeni objekt se prima, tj. transformira se na poziciju prethodno definiranog praznog objekta. Skupljanje ključa odvija se na identičan način, s tim da objekt prvo mora biti primljen, različita je jedino prazna skripta, „Key“ u ovom slučaju te tipka „F“ umjesto „E“ na tipkovnici. Nakon što je ključ pokupljen on nestaje sa scene.

Uz to, na objekte koje je moguće primiti dodane su još dvije mogućnosti, rotacija objekta oko dvije koordinatne osi pritiskom jedne ili dviju strelica na tipkovnici

te približavanje i udaljavanje objekta od igrača s definiranim granicama, tj. najvećom i najmanjom udaljenosti od igrača.

Osim objekata koje je moguće primiti u igri se može interaktivirati s još dvije vrste interaktivnih objekata, vratima i tipkovnicama za unos lozinki. Vrata se otvaraju pritiskom lijeve tipke na mišu, a prepoznavanje nalazi li se *crosshair* na vratima također se odvija pomoću *raycasta*. Otvaranje i zatvaranje vrata odvija se animacijom napravljenom u *Unity Animatoru*. Vrata mogu biti zaključana ili otključana o čemu odlučuje varijabla povezana s tipkovnicom za unos lozinke, tj.

Svaka zaključana vrata imaju pripadajuću tipkovnicu i lozinku. Tipkovnicom se također upravlja lijevim klikom miša pritiskom na odgovarajući gumb. Za rad tipkovnice zadužene su tri skripte, jedna za prepoznavanje lokacije *crosshaira* (također se koristi *raycast*), druga jednostavna skripta za slanje broja u treću koja upravlja radom tipkovnice. U njoj je moguće namjestiti lozinku, provjerava je li unesena točna lozinka te briše unos ako je unesena kriva lozinka ili ako je igrač kliknuo gumb za brisanje. Nakon unosa lozinke, klikom na gumb za potvrdu unosa igrač dobiva dvije povratne informacije, promijeni se boja unosa (zelena ili crvena) i pušta se odgovarajući zvuk, ovisno o tome je li unesena lozinka točna.



Slika 4. Tipkovnica za unos lozinke

2.4 Dizajn zagonetki

U igri postoji nekoliko zagonetki različitih težina, od najlakše u početnoj sobi koja služi za učenje korištenja tipkovnice za unos lozinke pa do vjerojatno najteže zagonetke koja koristi Morseovu abecedu. Za završetak igre potrebno je riješiti svih pet zagonetki (šest ako se uključi i zagonetka, odnosno lozinka u početnoj čije se rješenje nalazi odmah do tipkovnice) jer do nekih ključeva nije moguće doći bez otključavanja vrata.

Razina težine i način rješavanja zagonetki varira od jedne do druge, primjerice kod jedne je potrebno samo naći lako vidljiv trag koji vodi do objekta u sceni, dok neke druge zahtijevaju više vremena i razmišljanja. Također, rješenja nekih zagonetki je potrebno iščitati iz objekata na sceni, a za jednu je preporučeno i nošenje objekta s tragom. Vjerojatno najzanimljivija, ali i najteža je prethodno spomenuta zagonetka koja koristi Morseove abecedu kod koje svjetlo treperi u plavoj i crvenoj boji sukladno točkama i crticama za definiranu lozinku. Za treperenje je zadužena skripta s definiranom sekvencom, tj. brojevima i slovima, u ovom slučaju su to brojevi, ali skripta se može koristiti i u drugim situacijama.



Slika 5. Zagonetka koja koristi Morseovu abecedu

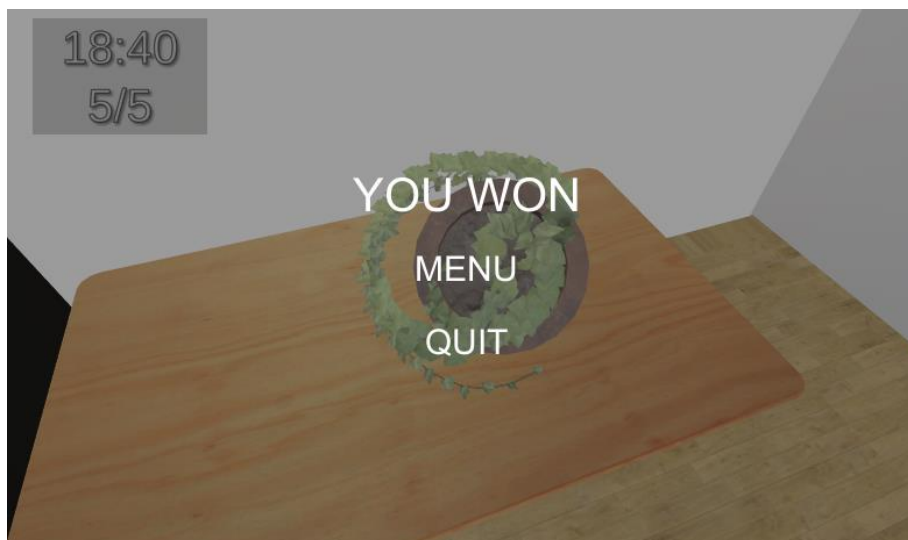
2.5 Izbornici, grafičko sučelje i završetak igre

Nakon pokretanja igre prvo što igrač vidi je početni izbornik koji je napravljen kao posebna scena za razliku od ostala tri izbornika prisutna u igri. Scena se sastoji od kamere te ugrađenih elemenata Unitya za izradu grafičkog sučelja. Igrač može pritisnuti na jedan od dva gumba, pritiskom na prvi ulazi u scenu u kojoj se odvija igra, a drugi služi za izlazak iz igre. Funkcionalnosti ovog, ali i ostalih izbornika su ostvarene u pripadajućoj skripti za svaki izbornik.

Pause izborniku igrač se može pristupiti u bilo kojem vremenu tijekom izvođenja igre pritiskom na tipku „Esc”, osim ako se nalazi u nekom drugom izborniku. Kada je ovaj izbornik aktivan vrijeme u igri je zaustavljeno što uključuje odbrojavanje preostalog vremena za rješavanje igre, kao i kretanje igrača. U ovom izborniku igrač može odabrati jednu od tri opcije, nastaviti s igrom klikom na gumb „Resume” ili tipkom „Esc” na tipkovnici, vratiti se u početni izbornik ili izaći iz igre.

Grafičko sučelje tijekom odvijanja igre je vrlo jednostavno, sastoji se od tri elementa, brojača ključeva, vremena koje odbrojava te crne pozadine u slučaju da neki od spomenuta dva brojača bude na bijelom zidu ili nekom drugom objektu jer se u suprotnom tekstu ne bi vidio s obzirom da su oba brojača bijele boje. Za oba brojača zadužene su jednostavne skripte koje stalno ispisuje trenutno stanje. Skripta za odbrojavanje ima definirano trajanje u sekundama te ispis ovisi o preostalom vremenu, oblik ispisa je 00:00:00 tako da se dodaju nule ako je to potrebno.

Igra može završiti na jedan od dva načina, ako je igrač skupio sve ključeve unutar određenog vremena igra se zaustavlja, prikazuje se izbornik za pobjedu. U suprotnom, ako je vrijeme isteklo prikazuje se izbornik za poraz. Oba izborniku su vrlo slična *pause* izborniku, razlika je što nema „Resume” gumba umjesto kojeg je prikazan odgovarajući tekst ovisno o izborniku te nema mogućnosti ulaska i izlaska iz izbornika, tj. ako je igra završila igrač se može vratiti u glavni izbornik ili izaći iz igre.



Slika 5. Izbornik za pobjedu

2.6 Upute za korištenje

Prije pokretanja igru je potrebno preuzeti s Git repozitorija što je moguće ostvariti na par načina. Ako korisnik ima instaliran Unity Engine i Unity Hub repozitorij je moguće klonirati na lokalni disk ili preuzeti, dodati projekt u Unity Hubu te ga pokrenuti u Unityu. Drugi način je preuzimanje instalacijske datoteke slijedeći poveznicu u Git repozitoriju na Google Drive lokaciju te instaliranje igre na računalo.

Nakon pokretanja igre, igrač se nalazi u početnoj sobi koja služi kao uvod u igru te za učenje kontrola i tehnika igre. Na zidu lijevo od početne pozicije igrača nalazi se pregled kontrola potrebnih za dovršetak igre, odnosno navedeno je kako primiti objekt i pokupiti ključ te otvoriti vrata.

Igrač se može kretati s „WASD” tipkama na tipkovnici, s lijevom tipkom miša upravlja s tipkovnicom za unos lozinke za otvaranje vrata, a s istom tipkom se otvaraju i vrata. Nakon unosa lozinke potrebno je pritisnuti gumb u donjem desnom uglu tipkovnice za potvrdu lozinke. Također, na tipkovnici postoji i tipka za brisanje unosa, to je tipka „C”. Objekt se prima pritiskom na tipku „E”, a ključ pokupiti tipkom „F” ako se *crosshair*, odnosno malo bijeli kružić na sredini kamere nalazi na ključu. *Pause* izborniku pristupa se s tipkom „Esc”. Igra, odnosno odbrojavanje započinje unosom prve lozinke.

3. Zaključak

Iako je igra ostvarena u sklopu laboratorijske vježbe kratka i jednostavna u njoj je dan pregled osnovnih tehnika u razvoju video igara. U velikoj većini igara se moguće kretati i interagirati s objektima, a svaka igra mora imati izbornike, ideju te dizajn scene, ali i same igre.

Kao i kod svake igre, i ova ima mnogo prostora za poboljšavanje i nadogradnju. Najočitija nadogradnja je proširenje igre, odnosno dodavanje novih razina (eng. *levela*) s novim zagonetkama, ali i mogućnost dodavanja novih zagonetki te produženja već napravljene razine. Također, moguće je i dodatno raditi na dizajnu i tijeku igre, tj. poboljšati trenutne zagonetke. Igra se može nadograditi i s grafičke strane, postoje objekti koji se mogu zamijeniti s grafički boljim, detaljnijim modelima s boljim teksturama.

4. Literatura

- [1] Unity Engine, <https://unity.com/>, siječanj 2021.
- [2] Meshes, Materials, Shaders and Textures, <https://docs.unity3d.com/Manual/Shaders.html>, siječanj 2021.
- [3] Unity Asset Store, <https://assetstore.unity.com/>, siječanj 2021.
- [4] Baked Lighting, <https://docs.unity3d.com/Manual/LightMode-Baked.html>, siječanj 2021.
- [5] Raycast, <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>, siječanj 2021.