

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klemen Krenker

Dirkalna igra z uporabo pogona Unity

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Preučite možnosti sistema Unity, predstavite podprte platforme, njegov obstoj v razvoju iger in primerjavo z drugimi sistemi. Razvijte 3D dirkalno igro in opišite glavne koncepte razvoja in probleme ter rešitve, ki ste jih uporabili.

Rad bi se zahvalil svojemu mentorju doc. dr. Luki Šajn za strokovno svetovanje in potrpežljivost pri nastajanju diplomskega dela.

Hvala tudi mami, očetu in sestri za vso podporo pri študiju.

Svoji dragi družini.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Dirkalne igre	3
3	Unity	5
3.1	Unity in podprte platforme	6
3.2	Igre izdelane s pogonom Unity	9
3.3	Unity in drugi pogoni	12
4	Implementacija 3D dirkalne igre v Unity	13
4.1	Predmeti (angl. Objects)	14
4.2	Uporabnikški vmesnik	16
4.3	Okolje	21
4.4	Igralec	25
4.5	Nasprotnik	32
4.6	Nadgradnje igre	34
5	Sklepne ugotovitve	35
	Literatura	37

Seznam uporabljenih kratic

kratica	angleško	slovensko
UE	unreal engine	unreal engine
UV	user interface	uporabniški vmesnik
UAS	unity asset store	trgovina sredstev unity

Povzetek

Naslov: Dirkalna igra z uporabo pogona Unity

Avtor: Klemen Krenker

Z izdelavo diplomske naloge sem se poizkušal postaviti v čevlje oblikovalcev računalniških iger. V tem času sem se srečal s številnimi problemi, ki oblikovalcu stojijo na poti pri izdelovanju dobro dodelane igre. Kot vsi začetniki sem probleme reševal na najbolj preprost in verjetno tudi ne najbolj optimalen način. Eden izmed težjih problemov je bilo ustvarjanje nasprotnika, ki poleg igralca dirka po progi. V večini računalniških iger se uporablja koncept težavnosti, saj si vsi igralci iger niso enaki in se močno razlikujejo v sposobnostih. Težavo sem odpravil tako, da sem jo omejil na tri glavne težavnosti, težja, lažja in vmesna. Glede na igralčevo izbiro težavnosti sem nato ustrezno prilagodil nasprotnika in s tem omogočil, da je nivo nasprotnika ustrezen igralcu.

Ključne besede: Unity, 3D igra, dirkalna igra.

Abstract

Title: Racing game using Unity

Author: Klemen Krenker

By making the Diploma thesis I tried to put myself into the shoes of game developers. During this time I became acquainted with many problems that stand in the way while making a well-made game. Like all beginners, I have solved the problems in the simplest and probably not the most optimal way. One of the hardest problems was creating an opponent who is racing the track besides the player. In most computer games the concept of difficulty is used, since all players are not the same and differ greatly in their abilities. I solved the problem by limiting myself to three main difficulties, hard, easy and medium. Depending on the player's choice of difficulty, I then adjusted the opponent accordingly, thus allowing the opponent's level to be appropriate for the player.

Keywords: Unity, 3D game, racing game.

Poglavje 1

Uvod

Video igre so že nekaj desetletij velik del naših življenj. Popularnost le teh ves čas narašča, kar pomeni, da se vlaga vanje vse več časa in denarja. Od majhnih razvijalskih skupin ali posameznika, do velikih organizacij, ki se že veliko let trudijo igralcem iger nuditi nekaj novega. V letu 2017 so igre ustvarile 108 milijard dolarjev, od tega mobilne igre kar 59 milijard, računalniške igre pa zavzemajo drugo mesto z 33 milijardami dolarjev [28]. Razvijanje iger je definitivno zelo dobičkonosno, je pa temu primerno tudi zahtevno delo. To velja najbolj za manjše skupine razvijalcev, saj je potrebno postoriti veliko dela, na primer grafično oblikovanje modelov, zvočni efekti, glasba, oblikovanje terena in uporabniškega vmesnika (UV) in pa seveda programiranje logike igre. 2. poglavje bo predstavilo nekaj ključnih informacij tega žanra igre. V 3. poglavju bo predstavljen pogon Unity, igre, ki so bile izdelane v pogonu in pa primerjava z drugimi podobnimi pogoni. 4. poglavje pa predstavlja mojo implementacijo igre. Predstavljeni bodo vsi elementi igre, težave, ki so mi bile na poti in pa rešitve s katerimi sem uspel rešiti problem.

Poglavje 2

Dirkalne igre

Dirkalne igre spadajo med enega od žanrov video igrar. Gre za igre, kjer igralec dirka po terenu, bodisi v avtomobilu, letalu, ali pa v katerem koli drugem prevoznem sredstvu [31].

Dirkalne igre se delijo na tri večje podzvrsti, kjer pa je vsaka izmed le-teh unikatna in igralcu omogoča nekaj specifičnih elementov dirkanja. Največje razlike se opazijo v resničnosti vožnje [32] [30].

Poglavje 3

Unity

Unity je pogon za izdelavo 2D in 3D iger [9], ki je bil izdan leta 2005 s strani podjetja Unity Technologies [20].



Slika 3.1: Unity, Unity Technologies 2005.

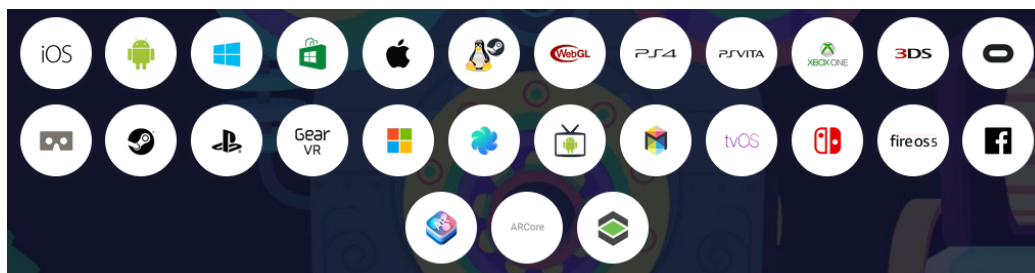
Programska oprema je tako zastopna, kot plačljiva, odvisno od tega kakšno podporo želimo in kakšen je namen našega razvoja igre. Trenutno obstajajo trije paketi, osebni, plus in pro paket [26].

1. Osebni paket [23] (angl. Personal)
 - Podpira razvoj na vseh Unity platformah,
 - omogoča dostop do trgovine z modeli (angl. Asset store),
 - podpora do 20 igralcev v več-igralskem načinu (angl. multiplayer),
 - omogoča brezplačno uporabo za dobičkonosne razvijalce, z omejitvijo do \$100.000 v letnem bruto prihodku

- in mnoge druge možnosti.
2. Plus paket [24] - \$25 mesečno
- Tečaji oblikovanja iger,
 - 20% popust pri nakupu sredstev v Unity trgovini,
 - možnost spremembe ali odstranitve začetnega zaslona (angl. Unity splash screen),
 - podpora do 50 igralcev v več-igralskem načinu (angl. multiplayer),
 - oglaševanje in prodaja produkta v sami aplikaciji Unity.
 - omogoča uporabo z omejitvijo do \$200.000 v letnem bruto prihodku
 - in mnoge druge možnosti.
3. Pro paket [25] - \$125 mesečno
- Dostop do izvorne kode,
 - analitika in izvoz podatkov analize,
 - podpora do 200 igralcev v več-igralskem načinu, možna je tudi nadgradnja,
 - poročanje o uspešnosti (angl. Performance reporting).
 - Zbiranje napak (angl. error),
 - reagiranje v resničnem času (angl. real-time),
 - hitro naslavljanje največjih problemov.

3.1 Unity in podprte platforme

Unity omogoča izdelavo 2D in 3D iger za tako mobilne, namizne, kot tudi konzolne platforme. Trenutno je v Unity podprtih kar 27 različnih platform [22]. V nadaljevanju bom na kratko predstavil večje podskupine podprtih platform in njihove največje predstavnike.



Slika 3.2: Podprte platforme v Unity

Ta trenutek so podprte že vse večje platforme namiznih oziroma prenosnih računalnikov:

- Windows,
- Linux,
- Mac.

Med zgoraj navedenimi sistemi ni opaziti velikih razlik, kar se tiče razvoja iger, vsi primarno uporabljajo miško in tipkovnico in enako tehnologijo.

Pri mobilnih napravah igralec ne uporablja miške, temveč zaslon na dotik in pa nagibanje naprave. Bolj pomembne in težje spremembe v primerjavi z računalniškimi platformami so pa seveda viri, ki so nam na voljo.

Mobilna naprava ima občutno manj spomina, prostora in pa manj zmogljivo centralno procesno enoto (CPE, angl. CPU). S tega razloga, je potrebno igro prilagoditi in ustrezno varčevati z njihovo porabo [4].

Podprte mobilne platforme:

- Android,
- iOS,
- Windows 10 mobile,

- Tizen.

Naslednja skupina podprtih platform so konzolne platforme (angl. consoles). Kot pri mobilnih napravah, se ne uporablja miške ali tipkovnice, uporabljajo se namreč igralne palice in pa kontrolerji (angl. game controller).



Slika 3.3: Xbox One brezžični kontroler, Microsoft 2016.

Konzole najlažje primerjamo z računalniškimi igrami, seveda ima vsaka tako dobre, kot tudi slabe lastnosti.

Konzolni sistemi so cenejši in lažji za uporabo. Eden izmed večjih problemov pa je kompatibilnost. Velikokrat se zgodi, da ob izdaji novega sistema, vse igre za star sistem niso prenosljive. Igre pa so po večini dražje, in slabše grafično dodelane.

Najbolj uporabljene konzolne platforme:

- PlayStation 4 ,
- Xbox One,
- Wii U ,

- 3DS.

Z razsvetom navidezne resničnosti, so omembe vredne tudi različne platforme, ki uporabniku omogočajo novo izkušnjo [27]. Med njimi so:

- Google Carboard,
- PlayStation VR,
- SteamVR,
- Oculus Rift.



Slika 3.4: Oculus Rift, Microsoft 2016.

3.2 Igre izdelane s pogonom Unity

Do sedaj je bilo izdanih že kar nekaj večjih iger, izdelanih s pogonu Unity.

3.2.1 Escape from Tarkov

Razvoj igre se je začel leta 2012 s strani razvijalca Battlestate Games [7]. V letu 2016 je bila igra na voljo vsem, ki so jo prednaročili. 28.7.2017 pa se je začelo obdobje, kjer je igra na voljo samo z nakupom [14].



Slika 3.5: Escape from Tarkov, Battlestate Games 2016.

3.2.2 Cuphead

Indie StudioMDHR (Studio Moldenhauer) je leta 2017 izdal 2D igro žanra 'run and gun'. Že v letu 2017 je igra dosegla več kot 2 milijona prodanih kopij in s tem dosegla velik uspeh. Zelo zanimivo je dejstvo, da so za izdelavo animacije uporabili metodo iz leta 1930 [29]. Vse slike so bile narisane na papir, in nato obarvane v programu Adobe Photoshop [11] [13].



Slika 3.6: Na list narisana skica, StudioMDHR 2017.

Čeprav je igra prikazana kot 60 sličic na sekundo (angl. frames per second), je animacija narejena samo na 24 sličic na sekundo.



Slika 3.7: Igra Cuphead, StudioMDHR 2017.

3.2.3 Slender: The Eight Pages

V letu 2012 je bila igra Slender: The Eight Pages [33] ena izmed najbolj popularnih prosto dostopnih grozljivih iger.

Čeprav v primerjavi z drugimi igrami, ki sem jih omenil, ni grafično usmerjena, je igralce verjetno privlačila legenda visokega brezobraznega moškega. Gre za zlobno izmišljeno osebo, ki naj bi bila povezana z ugrabitvami otrok v gozdovih ali zapuščenih stavbah.



Slika 3.8: Slender: The Eight Pages, Parsec Productions 2012.

3.3 Unity in drugi pogoni

Igralni pogon (angl. Game engine) [5] je okolje, ki omogoča razvijalcem iger lažje in hitrejšo razvijanje iger. Vsebujejo zaznavanje trkov, fiziko objektov, animiranje, zvočno podporo, itd. Načrtovanje in razvoj novega igralnega pogona bi zahtevalo veliko časa in pa znanja, zato veliko razvijalcev iger uporabi že obstoječe, dobro razvite pogone, kot so:

- Unreal Engine (UE),
- Unity,
- GameMaker,
- Godot,
- CryEngine.

V nadaljevanju bom izpostavil Unity, saj sem v njem tudi sam izdeloval igro, in pa UE.

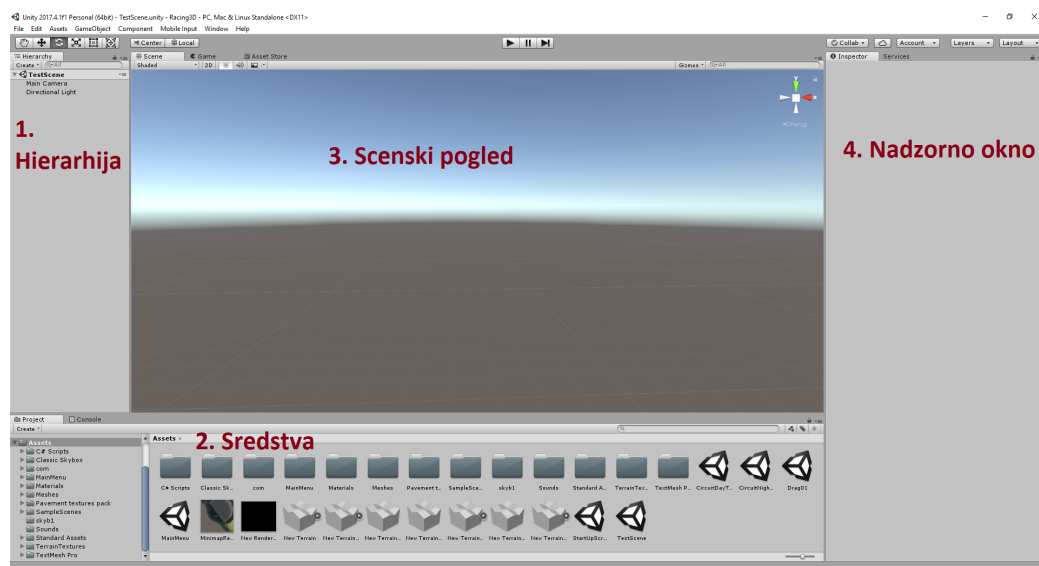
Razlike med Unity in UE [10]:

- Programski jezik, Unity uporablja C# in Javascript, medtem ko UE uporablja C++.
- Uporaba 'blueprint' načina v UE, kjer uporabnik za razvoj igre ne potrebuje pisati kode v jeziku C++. Seveda pa ima ta način tudi svoje omejitve.
- Cena uporabe, kjer je UE prostodostopen pogon. V kolikor igra postane dobičkonosna, Epic Games, razvijalec pogona vzame 5% dobička.

Poglavje 4

Implementacija 3D dirkalne igre v Unity

Z vsako nadgradnjo in spremembo UV je lažje razvijanje projektov v Unity, tako za izkušene razvijalce, kot tudi začetnike, ki sploh ne poznajo programske opreme.



Slika 4.1: Unity vmesnik, Unity Technologies, 2017

Slika (glej sliko 4.1) prikazuje štiri večje komponente vmesnika [15], ki vsaka po svoje pomaga pri razvoju.

1. *Hierarhija* - pomaga pri ureditvi objektov v posameznem prizoru.
2. *Sredstva* - vsebuje datoteke, ki so vključene v celoten Unity projekt.
3. *Scenski pogled* - služi kot predogled celotnega okolja igre.
4. *Nadzorno okno* - ob označitvi elementa pomaga pri prikazu njegovih nastavitev in komponent.

4.1 Predmeti (angl. Objects)

Unity nam omogoča vpeljavo modelov v različnih formatih. Izvožene 3D datoteke, ki so podprte [16], si lahko ogledamo na sliki 4.2.

Exported 3D files

Unity can read [.fbx](#), [.dae \(Collada\)](#), [.3ds](#), [.dxf](#), [.obj](#), and [.skp](#) files.

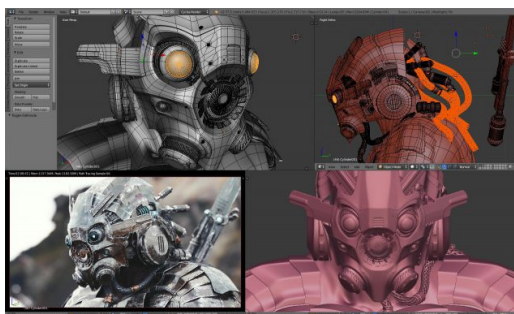
Slika 4.2: Unity documentation, 2018

Poleg teh 3D datotek, je omogočen tudi uvoz zaščiteneh aplikacijskih datotek, ustvarjenih v naslednjih programih [16]:

- 3D Studio Max,
- Maya,
- Blender,
- Cinema4D,
- Modo,
- LightWave,
- Cheetah3D.

4.1.1 Blender

Blender [6] je odprtokoden program za modeliranje, animiranje, simulacijo 3D objektov. Z velikim številom uporabnih orodij, je orodje popularno pri izdelavi modelov v igrah, filmih, oglaševanju, izobraževanju in na drugih področjih.



Slika 4.3: Blender, Blender Foundation, 2018

4.1.2 Maya

Maya [2] je plačljiva programska oprema, ki omogoča 3D animiranje, modeliranje, simulacijo in upodabljanje (ang. rendering).

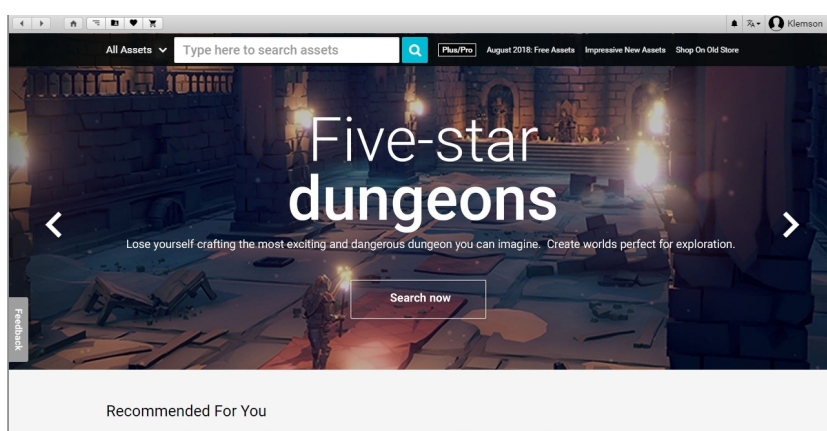


Slika 4.4: Maya, Autodesk, Inc., 2018

4.1.3 Unity Asset Store

Unity Asset Store (UAS) [21] omogoča razvijalcem nakup objektov, in drugih elementov igre, v kolikor elementov sami ne želijo izdelati. Možen je prenos in nakup tako neplačljivih, kot tudi plačljivih elementov.

Dostop je mogoč tako preko brskalnika, kot tudi direktno iz pogona Unity (glej sliko 4.5),



Slika 4.5: Unity Asset Store dostopno iz pogona Unity, 2018

Poleg 2D in 3D elementov, trgovina vsebuje tudi zvočne efekte in glasbo, različna orodja, predloge, vizualne učinke (angl. VFX) in pa dodatke. Glede na zahtevnost modeliranja, animiranja in vseh drugih dejavnosti vključene v izdelavo objektov je nakup le teh včasih zelo dobra rešitev. Tudi sam sem poiskal kvalitetne objekte v UAS in jih vključil v svoji igri.

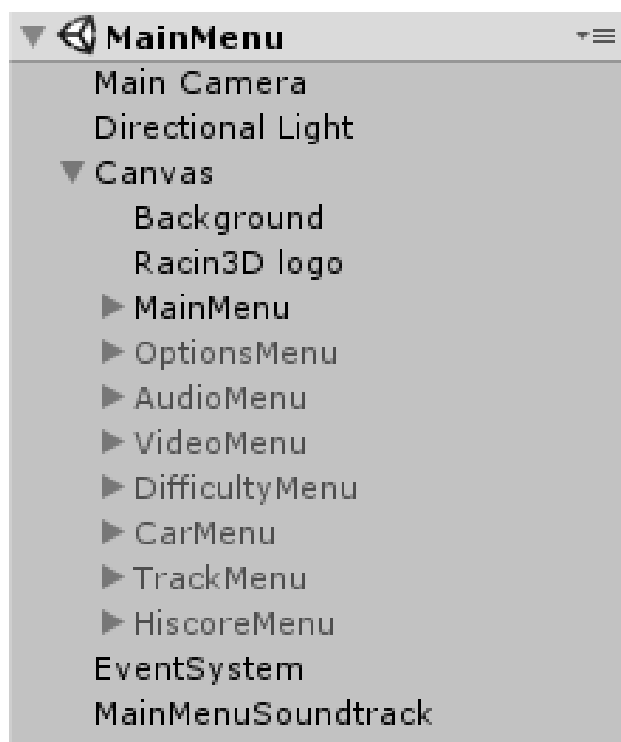
4.2 Uporabnikški vmesnik

Glavna cilja implementacije vmesnika sta bila preprostost in lahka dostopnost do ključnih informacij. Vsi elementi UV so bili izdelani z uporabo platna (angl. Canvas) [12]. Platno je pravokotnik, ki predstavlja vidno polje igralca. Vsi elementi UV morajo biti vsebovani v objektu 'Canvas', njihov položaj v

hierarhiji pa označuje vrstni red izrisa. Vmesnik med uporabnikom in igro sem razdelil na tri večje sklope, ki jih bom v nadaljevanju natančneje opisal.

4.2.1 Glavni meni

Glavni meni sem izdelal s pomočjo novega prizora. Prizor (angl. Scene) si lahko predstavljamo kot samostojno okolje znotraj celotnega projekta. Uporabil sem jih pri glavnih menijih, začetnih oknih in nivojih igre. Izgled hierarhije glavnega menija je prikazan na sliki 4.6.



Slika 4.6: Glavni meni

Nekateri elementi so obarvani s sivo barvo, kar pomeni, da element trenutno ni aktiven v prizoru. Z vklopom in izklopom posameznih elementov UV lahko tako preprosto prikažemo ali skrijemo določene elemente. Vsak podmeni oziroma sklop je združen v svoj igralni objekt(angl. Game

Object), ki omogoča lažji pregled in druge priročne možnosti, npr. 'Main-Menu', 'OptionsMenu'.

Glavni meni obsega gumbe, ki vodijo do željenega rezultata(glej sliko 4.7):

- *PLAY*, ki odpre podmeni z izbiro avtomobila.
- *OPTIONS* z nastavitvijo glasnosti in načina igre.
- *HISCORES*, ki prikazuje rekordne čase in ime igralca vsake dirkalne proge.
- *QUIT* za končanje igre.

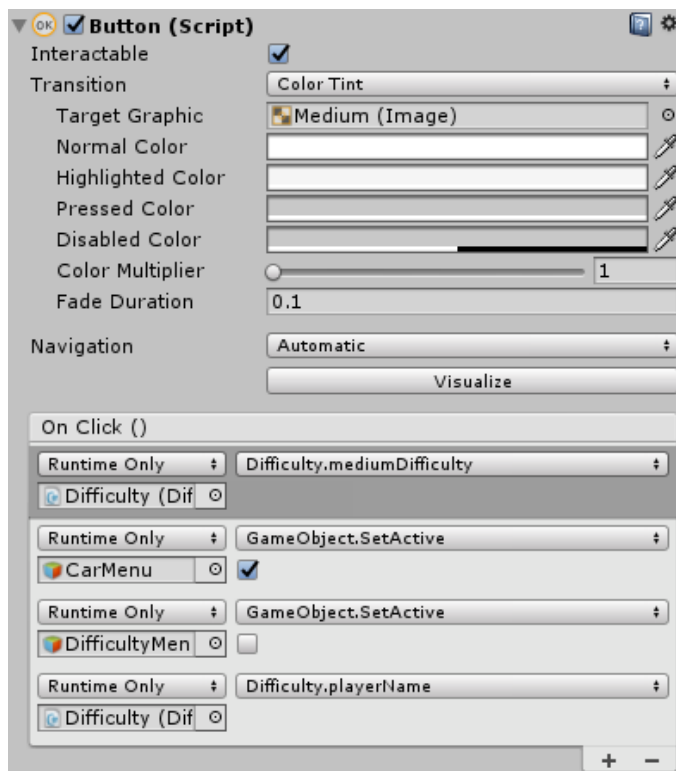


Slika 4.7: Glavni meni

V Unity imajo gumbi 'on click' metode, ki se prožijo ob kliku na gumb. Ob kliku na gumb se lahko proži več kot en dogodek. V primeru na sliki 4.8 se ob kliku na gumb zgodijo štirje dogodki:

- Klic metode 'mediumDifficulty',
- klic metode 'playerName',
- aktivacija ig. objekta 'CarMenu',
- deaktivacija ig. objekta 'DifficultyMenu'.

Težava proženja dogodkov ob kliku na gumb je rešljiva s pomočjo programskega jezika C#, a Unity omogoča lažjo in hitrejšo rešitev z uporabo grafičnega načina.

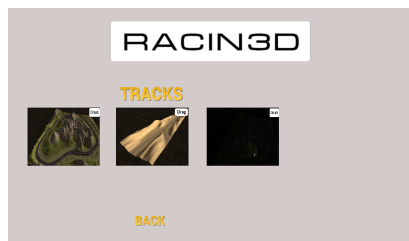


Slika 4.8: on click metode gumba

Pritisk na *PLAY* odpre podmeni z izbiro avtomobila, imena, težavnostjo in progo, na kateri se bo odvila dirka (glej sliki 4.9, 4.10).



Slika 4.9: Izbira avtomobila



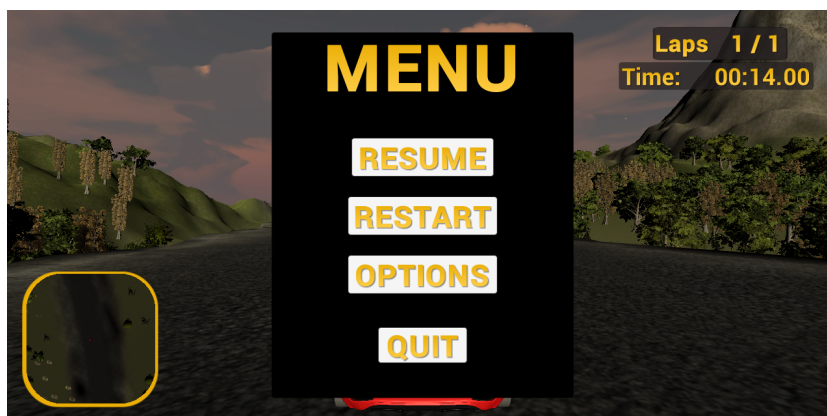
Slika 4.10: Izbira dirkalne proge

4.2.2 Meni v igri

V primerjavi z glavnim menijem ima meni v igri manj možnosti, drugače pa je njun izgled enak. Največja razlika je v izvedbi, namreč nisem uporabil prizora, temveč sem igro začasno ustavil in na ekranu prikazal meni.

```
Time.timeScale = 0f;  
gameMenu.SetActive(true);
```

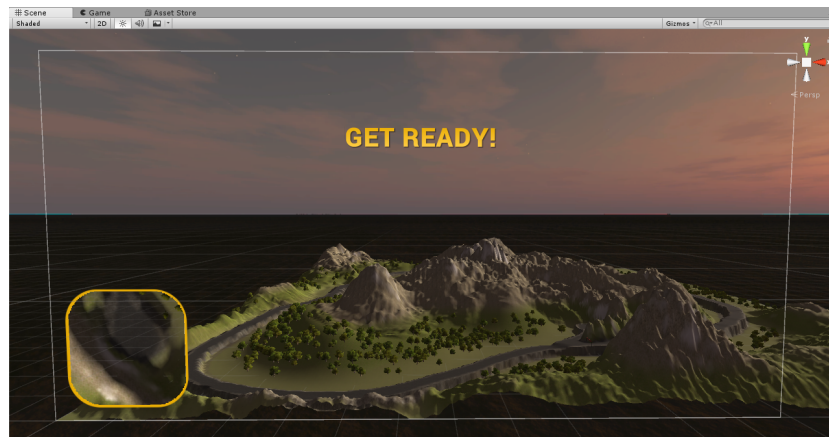
Prva vrstica omogoča zaustavitev igre pri čemer se čas popolnoma ustavi. Druga vrstica pa vklopi meni. V kolikor igralec nadaljuje z igro, vrednosti preprosto obrnemo in dobimo želen rezultat.



Slika 4.11: Meni v igri

4.2.3 UV v igri

Od zemljevida do izpisa preteklega časa so vsi napisi del vmesnika. Nekateri so vidni na sliki 4.12.

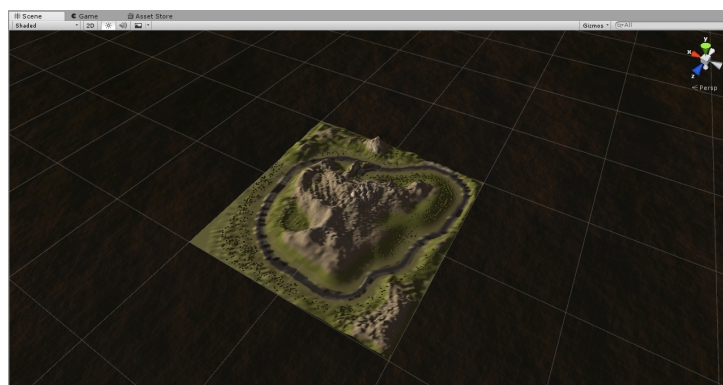


Slika 4.12: Elementi UV v igri

4.3 Okolje

4.3.1 Teren

Za izdelavo kompleksnejših terenov so na voljo generatorji, dobra primera sta Gaia [34] in World Creator [3]. Primera neplačljivih generatorjev pa sta World Builder [1] in Terrain Toolkit [8].

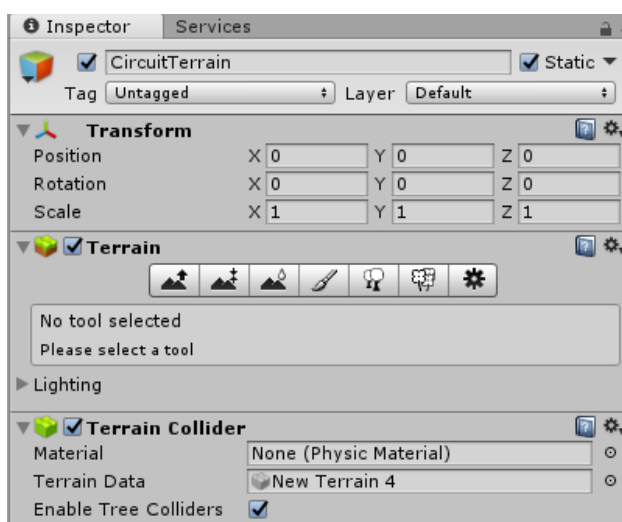


Slika 4.13: Teren dirkalne proge

Oblikovanje

Pri oblikovanju sem uporabil Unity oblikovalec terena [18]. Orodje je zelo priročno in izpopolnjeno za izdelavo preprostejših terenov. Z uporabo čopičev različnih oblik in velikosti sem lahko preprosto dvigal oziroma spuščal dele terena, da sem dobil dokaj razgiban teren. Orodje omogoča tudi dvigovanje površine do določene točke. Tako sem izdelal dvignjeno cesto, po kateri dirka igralec. Na koncu sem uporabil še orodje za glajenje, s katerim sem zgladil prevelike višinske razlike na terenu.

Oblike in pa nastavitve velikosti ter moči so prikazane na sliki 4.14.

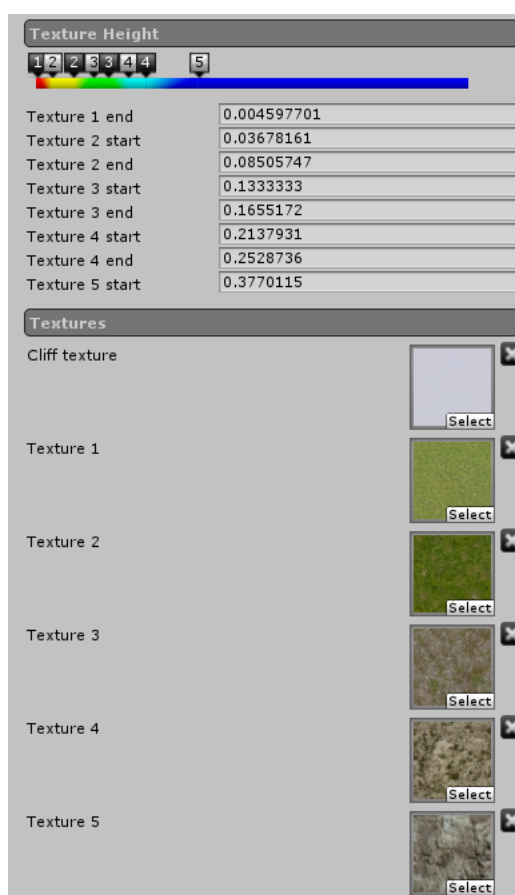


Slika 4.14: Oblikovanje terena v Unity

Teksture

Vse teksture uporabljene pri izdelavi terena so bile pridobljene v UAS. Nanos tekstur bi lahko opravil na enak način kot oblikovanje terena, a sem raje uporabil orodje Terrain Toolkit [8]. S pomočjo tega orodja sem teksture nanesel na površino samodejno. Pri izdelavi terena na sliki 4.13 sem uporabil šest različnih tekstur, vsaka je namenjena svoji vrsti podlage. Vsako od le-teh sem vnesel v orodje in nastavljal meje med njimi. Tekstura je nanešena na

območjih med začetkom in koncem, ki sem ga določil. Začetna in končna točka sta določeni z višino terena, na kateri je izveden nanos določene texture.

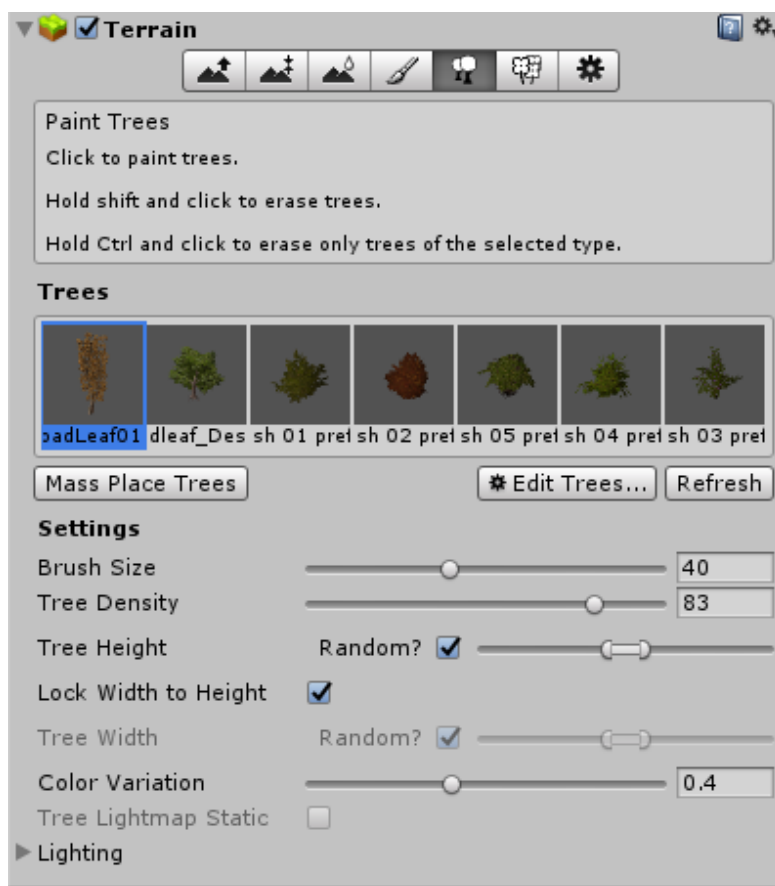


Slika 4.15: Terrain Toolkit, 2018

Dodajanje dreves in grmovja

Orodje za oblikovanje terena v Unity omogoča tudi dodajanje elementov na teren. Ta del orodja sem uporabil za dodajanje dreves in grmovja. Modele dreves, pridobljene v UAS [21] sem nato nanesel na igralno površino. Postopek dodajanja je preprost, po izbiri drevesa, sem nastavil velikost čopiča

in pogostost dreves. Prav tako sem določil barvno variacijo drevesa in pa interval višine ter širine drevesa (slika 4.16).



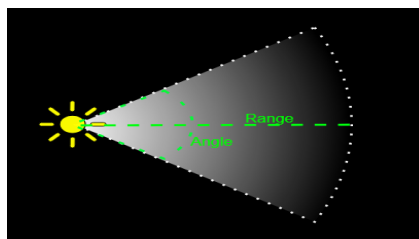
Slika 4.16: Dodajanje elementov v Unity, 2018

4.3.2 Osvetlitev

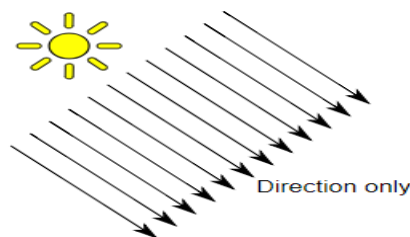
V igri sem uporabil dve vrsti osvetlitve.

- *Žaromet (angl. Spot light)* [19], ki je uporabljen kot luč avtomobila. Luč sveti v kotu do določene dolžine, kjer svetloba upade (slika 4.17). Ta vrsta svetlobe je primerna za imitacijo luči avtomobila.

- *Usmerjena svetloba* (angl. *Directional light*) [19], ki služi kot primarna svetloba oz. sonce ali luna. Pri usmerjeni svetlobi je pomembna usmerjenost svetlobe, saj so osvetljeni vsi elementi v prizoru (slika 4.18).



Slika 4.17: Žaromet



Slika 4.18: Usmerjena svetloba

4.4 Igralec

Igralec je v naši igri avtomobil, ki dirka po progi. Primarno je uporabljena tretje-osebna kamera.

Nadzor avtomobila in drugih elementov:

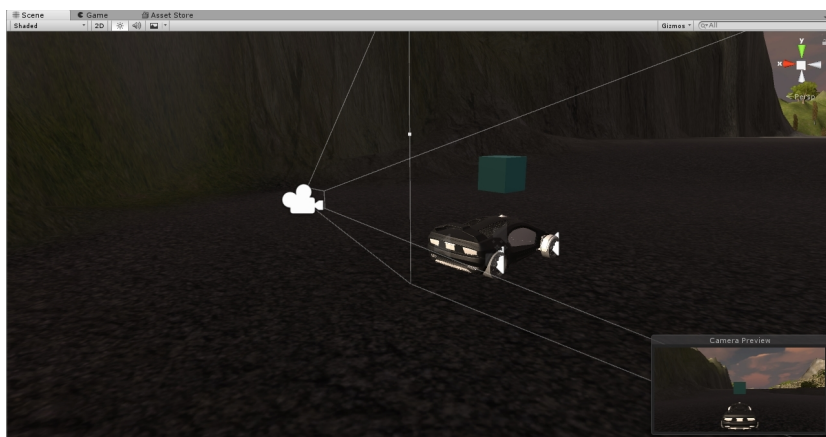
- *W,A,S,D/Puščice*: Premikanje vozila.
- *V*: Sprememba pogleda, vsebuje tri različne poglede.
- *T*: Vzratno ogledalo.
- *F*: Postavitev vozila na vozišče.
- *L*: Žarometi avtomobila.

4.4.1 Kamere

Za izvedbo celotne igre sem uporabil tri kamere:

- Glavna kamera, ki je usmerjena v kocko nad avtomobilom. Z uporabo kocke in preproste skripte sem omogočil izgled stabilnosti kamere.

- Vzratna kamera na vozilu.
- Kamera za realizacijo zemljevida.



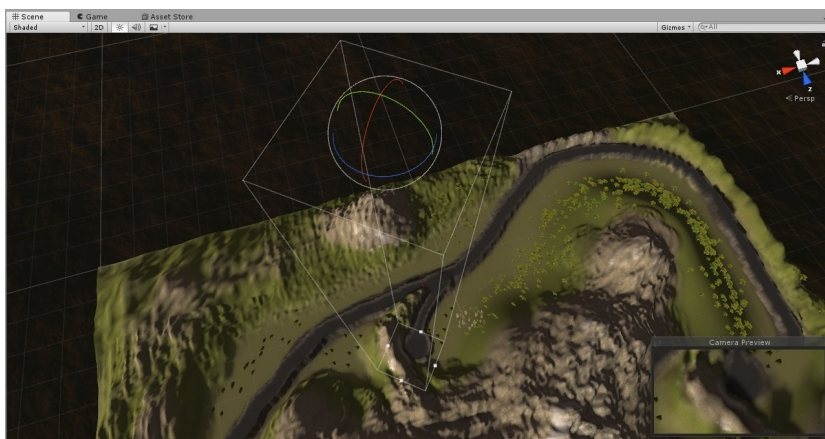
Slika 4.19: Glavna kamera

Za realizacijo zemljevida sem uporabil kamero, ki se nahaja nad igralcem in mu sledi. Pogled kamere sem nato za vsako sličico (angl. frame) posebej posodobil s pomočjo posebne vrste strukture. Tekstura za upodabljanje (angl. render texture) omogoča samodejno posodabljanje. Za izris zemljevida na ekran sem nato uporabili 'raw image' objekt, ki v primerjavi z drugimi načini prikaza, lahko prikazuje tudi to teksturo.

Kamera, ki je uporabljena za zemljevid je ortografska kamera (slika 4.20), le-ta odstrani perspektivnost in je predvsem uporabljena za 2D poglede. Obe drugi kameri pa sta perspektivni in uporabljeni za prikaz 3D okolja, poleg tega so vse nastavitve enake.

Sprememba pogleda

Stabilizacija glavne kamere je izvedena s pomočjo 3D kocke v katero je kamera usmerjena, zato je potreben le premik kocke in s tem tudi kamere.



Slika 4.20: Kamera za izvedbo zemljevida

Kocko sem premikal na mesta označena s pomočjo nevidne kocke, ki v prizoru ni fizično prisotna. To pomeni, da ni izrisana in dotik le-te s katerikoli objektom v igri nemogoč. Prvi trije stavki predstavljajo shranjevanje x, y in z koordinate nevidne kocke v spremenljivke. Zadnji stavek pa spremeni položaj kocke in s tem kamere.

```
float xpos = cameraPositions[cameraPositionNum].
    GetComponent<Transform>().position.x;
float ypos = cameraPositions[cameraPositionNum].
    GetComponent<Transform>().position.y;
float zpos = cameraPositions[cameraPositionNum].
    GetComponent<Transform>().position.z;

transform.position = new Vector3(xpos, ypos, zpos );
```

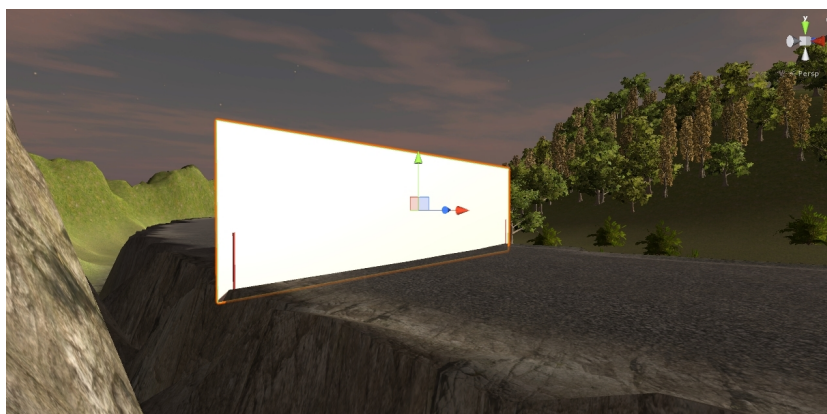
4.4.2 Nadzorna točka (angl. Checkpoint)

Z uporabo naslednjega sistema poizkušamo minimalizirati goljufanje igralca. Po progi so postavljene nadzorne točke, ki jih mora igralec prečkati. V koli-



Slika 4.21: Položaji kock, ki držijo glavno kamero

kor igralec zgreši eno ali več točk, prečkanje cilja ne deluje in igralec ne zmaga.



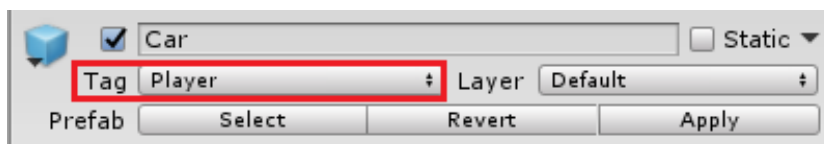
Slika 4.22: Glavna kamera

Za kontrolno točko sem uporabil 3D kocko, in jo preoblikovali v tanek, dolg in širok pravokotnik, ki pokriva celotno širino dirkalne proge na tisti točki.

Pravokotnik vsebuje orodje zaznavanje trka. Uporabil sem metodo 'OnTriggerEnter', ki se proži ko nek objekt vstopi v pravokotnik.

Ker sem se odločil, da bom v nadaljevanju dodal tudi nasprotnika sem

prožilca dogodka moral označiti in prepoznati. Za označevanje objektov sem uporabili oznake (angl. tag) [17], ki so dostopne v pogonu Unity.



Slika 4.23: Oznaka v Unity

Preko oznake je razpoznavno, ali je nadzorno točko prečkal igralec ali nasprotnik. V primeru, da nasprotnik prečka igralčevo nadzorno točko, moramo ta dogodek ignorirati in počakati na igralca. Če pa dogodek sproži igralec, ki prečka nadzorno točko, moramo aktivirati naslednjo točko, trenutno pa deaktivirati.

```
if (other.gameObject.CompareTag("Player"))
{
    waypoints[1].SetActive(true);
    waypoints[0].SetActive(false);
}
```

Igralec je označen z oznako 'Player'. V kolikor je prožilec dogodka označen z oznako 'Player' se aktivira naslednja točka, trenutna pa se deaktivira.

4.4.3 Ponovna postavitvev igralca na progo (angl. Respawn)

Igralcu med igranjem igre je ponujena tako možnost ponovnega začetka, kot tudi ponovna postavitvev na igralno površino. Možne uporabe so seveda, če igralec obtiči na igralni površini, pade izven površine itd. Čeprav se razvijalci trudijo, da igralec ne more obtičati nikjer v igri ali pasti iz terena, se

to včasih zgodi. Igralec ima kadarkoli priložnost pritisniti na tipko 'F' in se ponovno vrniti na dirkalno površino. Kraj, kjer se igralec ponovno pojavi, je v igri odvisen od tega, kako daleč je igralec prišel. Ob prečkanju nadzorne točke iz prejšnjega poglavja, se zamenja tudi 'respawn' točka, ki je vedno locirana za nadzornimi točkami in tako je goljufanje nemogoče.

Ob pritisku na tipko 'F' je torej igralec postavljen na trenutno 'respawn' točko, vozilo je tudi ustavljeno in pravilno obrnjeno za nadaljevanje dirkanja.

```
if ( Input.GetKeyDown(KeyCode.F))
{
    Vector3 respawnCoord = getRespawnCoords();
    Vector3 respawnRotation = getRespawnRotation();

    car.SetActive(false);
    car.GetComponent<CarController>().m_Topspeed = 0f;
    car.SetActive(true);

    car.GetComponent<CarController>().m_Topspeed =
        settings.GetComponent<Settings>().playerTopSpeed;

    car.transform.position = respawnCoord;
    car.transform.rotation = Quaternion.Euler(respawnRotation.x,
        respawnRotation.y, respawnRotation.z);
}
```

4.4.4 Matematični problemi

V igro sem vključil tudi reševanje matematičnih problemov med dirkanjem. Igralec mora tako med dirkanjem ne samo paziti na svojo vožnjo, ampak tudi nekoliko napeti možgane in pravilno rešiti dani problem.

Težavnost problema je odvisna od težavnosti dirke. S spremembo težavnosti



Slika 4.24: Matematični problem

se spremeni kompleksnost problema in pa čas, ki ga ima igralec, da reši problem.

[0-20] predstavlja zaprti interval števil, izmed katerih je izbrana naključna številka. Ta številka označuje čez koliko sekund se bo pojavil nov matematični problem. Težavnosti:

- *Težja (angl. Hard):* $[0-20] * [0-20] + [0-20]$
 - Čas reševanja: 12 sekund
- *Vmesna (angl. Medium):* $[0-10] * [0-10] + [0-10]$
 - Čas reševanja: 8 sekund

- *Lahka (angl. Easy):* $[0-10] * [0-10]$

– Čas reševanja: 5 sekund

Pojavitev matematičnega problema je prav tako odvisna od naključne številke izbrane iz intervala [28-38]. Vsaka napačna rešitev poveča končen del intervala za 1 sekundo, vsaka pravilna rešitev pa interval zmanjša za eno sekundo.

Če igralec pravilno reši problem, se ne zgodi nič, v kolikor pa je odgovor napačen ali ga ni, pa se pred igralcem pojavi olje na cesti, ki igralca močno zavrti, občutno zmanjša njegov oprijem s cesto in oteži zavijanje.



Slika 4.25: Olje na progi

4.5 Nasprotnik

Nasprotnika v igri predstavlja avtomobil, ki tekmuje z igralcem.

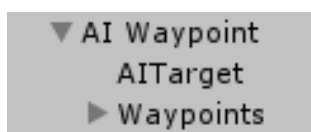
Za premikanje nasprotnika so uporabljene smerne točke (angl. waypoint), na sliki 4.26 si lahko ogledamo izgled med testiranjem, v končni različici pa so seveda skrite.

Ob nasprotnikovem prečkanju točke, je naloga pravilna določitev naslednje točke. Za izvedbo sem uporabil 3D kocke. Primarno kocko, ki se bo pre-



Slika 4.26: Smerne točke

mikala po progi in bo veljala za tarčo nasprotnika ('AITarget' na sliki 4.27). In pa večje število kock, ki označujejo položaj naše tarče ('Waypoints' na sliki 4.27).



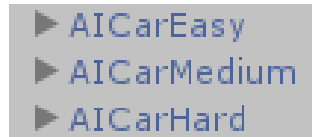
Slika 4.27: Hierarhija smernih točk

Uporabljene so tri različne težavnosti:

- *Težja (angl. Hard)*: 220 km/h (kilometrov na uro).
- *Vmesna (angl. Medium)*: 180 km/h.
- *Lahka (angl. Easy)*: 150 km/h.

Vsaka od težavnosti ima svojo maksimalno hitrost avtomobila. Lahka in vmesna težavnost uporabljata iste smerne točke, medtem ko težja težavnost uporablja druge smerne točke, ki so bolj optimalne in tako omogočajo hitrejšo in dokaj optimalno vožnjo.

Za vsako od težavnosti je uporabljen svoj avtomobil, pravilen je aktiviran pred začetkom igre, odvisno od izbire igralca. Slika 4.28 prikazuje neaktivne nasprotnikove avtomobile.



Slika 4.28: Nasprotniki različne težavnosti

4.6 Nadgradnje igre

Možnosti za nadgradnjo je res veliko, saj je v podžanru dirkalnih iger veliko uspešnih iger, od katerih bi lahko pridobil ideje o izboljšavah.

Kot velika nadgradnja, bi lahko upoštevali dodajanje dodatnih vozil, ki bi bili na voljo igralcu. Igralcu bi lahko tudi omogočili, da sam določi težavnost vožnje avtomobila in pa ročno menjavo prestav.

Izboljšali bi lahko tudi nasprotnika in povečali dinamičnost s tem, da bi se težavnost nasprotnika spreminjala med samo dirko in ne samo pred začetkom. Drastično bi lahko izboljšali tudi okolje igre, kjer bi z uporabo plačniških generatorjev terena [34] izdelali bolj kompleksen in realističen teren.

Poglavje 5

Sklepne ugotovitve

Z izdelavo diplomske naloge sem se nekoliko poglobil v izdelavo 3D igre v pogonu Unity in tako spoznal njegovo okolje in pa slabe ter dobre lastnosti. Izvedel sem nekaj več o podprtih platformah in o uspešnih igrah, ki so bile izdelane v programu. Poleg tega sem se seznanil z izdelavo preprostih a učinkovitih menijev znotraj igre in z metodami pridobitve ali ustvarjanja objektov, ki so nato dodani v igro. V teku izdelave 3D igre sem se naučil izdelovati igralno površino, osvetlitev le-te in pa uporabo nasprotnika (angl. AI).

Izdelovanje 3D igre seveda ni enostavno, kar sem kar hitro ugotovil med izdelovanjem. Potrebna je velika količina znanja, dela in predanosti pri izdelavi dobro dodelane, konkurenčne igre.

Literatura

- [1] AJOpinnaytetyo. World builder. Dosegljivo: <https://assetstore.unity.com/packages/tools/terrain/world-builder-11333>. [Dostopano: 19. 8. 2018].
- [2] Autodesk. Maya. Dosegljivo: <https://www.autodesk.com/products/maya/overview>. [Dostopano: 19. 8. 2018].
- [3] BiteTheBytes. World creator. Dosegljivo: <https://www.worldcreator.com/>. [Dostopano: 19. 8. 2018].
- [4] Unity Technologies Documentation. Unity cross-platform considerations. Dosegljivo: <https://docs.unity3d.com/Manual/CrossPlatformConsiderations.html>. [Dostopano: 19. 8. 2018].
- [5] David H Eberly. *3D game engine design: a practical approach to real-time computer graphics*. CRC Press, 2006.
- [6] Blender Foundation. Blender. Dosegljivo: <https://www.blender.org/>. [Dostopano: 19. 8. 2018].
- [7] Battlestate Games. Escape from tarkov. Dosegljivo: <https://www.escapefromtarkov.com/>. [Dostopano: 19. 8. 2018].
- [8] HEPARO. Terrain toolkit 2017. Dosegljivo: <https://assetstore.unity.com/packages/tools/terrain/terrain-toolkit-2017-83490>. [Dostopano: 19. 8. 2018].
- [9] Joseph Hocking et al. *Unity in action*. Manning, 2017.

-
- [10] Pluralsight. Unreal engine 4 vs. unity: Which game engine is best for you? Dosegljivo: <https://www.pluralsight.com/blog/film-games/unreal-engine-4-vs-unity-game-engine-best>. [Dostopano: 19. 8. 2018].
- [11] StudioMDHR. Cuphead. Dosegljivo: <http://www.cupheadgame.com/>. [Dostopano: 19. 8. 2018].
- [12] Unity Technologies. Canvas. Dosegljivo: <https://docs.unity3d.com/Manual/UICanvas.html>. [Dostopano: 19. 8. 2018].
- [13] Unity Technologies. Cuphead - made with unity. Dosegljivo: <https://unity.com/madewith/cuphead>. [Dostopano: 19. 8. 2018].
- [14] Unity Technologies. Escape from tarkov - made with unity. Dosegljivo: <https://unity.com/madewith/escape-from-tarkov>. [Dostopano: 19. 8. 2018].
- [15] Unity Technologies. Interface overview. Dosegljivo: <https://unity3d.com/learn/tutorials/topics/interface-essentials/interface-overview?playlist=17090>. [Dostopano: 19. 8. 2018].
- [16] Unity Technologies. Model file formats. Dosegljivo: <https://docs.unity3d.com/Manual/3D-formats.html>. [Dostopano: 19. 8. 2018].
- [17] Unity Technologies. Tags. Dosegljivo: <https://docs.unity3d.com/Manual/Tags.html>. [Dostopano: 19. 8. 2018].
- [18] Unity Technologies. Terrain engine. Dosegljivo: <https://docs.unity3d.com/Manual/script-Terrain.html>. [Dostopano: 19. 8. 2018].
- [19] Unity Technologies. Types of lights. Dosegljivo: <https://docs.unity3d.com/Manual/Lighting.html>. [Dostopano: 19. 8. 2018].
- [20] Unity Technologies. Unity. Dosegljivo: <https://unity3d.com/>. [Dostopano: 19. 8. 2018].

-
- [21] Unity Technologies. Unity asset store. Dosegljivo: <https://assetstore.unity.com/>. [Dostopano: 19. 8. 2018].
- [22] Unity Technologies. Unity multiplatform. Dosegljivo: <https://unity3d.com/unity/features/multiplatform>. [Dostopano: 19. 8. 2018].
- [23] Unity Technologies. Unity personal. Dosegljivo: https://store.unity.com/products/unity-personal?_ga=2.208783890.1869666610.1534676824-528401881.1524392168. [Dostopano: 19. 8. 2018].
- [24] Unity Technologies. Unity plus. Dosegljivo: https://store.unity.com/products/unity-plus?_ga=2.208783890.1869666610.1534676824-528401881.1524392168. [Dostopano: 19. 8. 2018].
- [25] Unity Technologies. Unity pro. Dosegljivo: https://store.unity.com/products/unity-pro?_ga=2.208783890.1869666610.1534676824-528401881.1524392168. [Dostopano: 19. 8. 2018].
- [26] Unity Technologies. Unity products. Dosegljivo: <https://unity3d.com/unity>. [Dostopano: 19. 8. 2018].
- [27] Unity Technologies Tutorials. Vr overview - unity. Dosegljivo: <https://unity3d.com/learn/tutorials/topics/virtual-reality/vr-overview>. [Dostopano: 19. 8. 2018].
- [28] James Batchelor (ured.). gI games industry generated \$108.4bn in revenues in 2017. Dosegljivo: <https://www.gamesindustry.biz/articles/2018-01-31-games-industry-generated-usd108-4bn-in-revenues-in-2017>. [Dostopano: 23. 7. 2018].
- [29] Andrew Webster. Cuphead: creating a game that looks like a 1930s cartoon. Dosegljivo: <https://www.theverge.com/2017/9/28/16378364/cuphead-art-design-1930s-animation>. [Dostopano: 19. 8. 2018].

- [30] Wikipedia. Kart racing game. Dosegljivo: https://en.wikipedia.org/wiki/Kart_racing_game. [Dostopano: 19. 8. 2018].
- [31] Wikipedia. Racing video game. Dosegljivo: https://en.wikipedia.org/wiki/Racing_video_game. [Dostopano: 19. 8. 2018].
- [32] Wikipedia. Sim racing. Dosegljivo: https://en.wikipedia.org/wiki/Sim_racing. [Dostopano: 19. 8. 2018].
- [33] Wikipedia. Slender: The eight pages. Dosegljivo: https://en.wikipedia.org/wiki/Slender:_The_Eight_Pages. [Dostopano: 19. 8. 2018].
- [34] Procedural Worlds. Gaia. Dosegljivo: <http://www.procedural-worlds.com/gaia/>. [Dostopano: 19. 8. 2018].