Dvodimenzionalna akcijska igra s horizontalnim pomicanjem

Varnica, Marin

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:944449

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-26**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

DVODIMENZIONALNA AKCIJSKA IGRA S HORIZONTALNIM POMICANJEM

Završni rad

Marin Varnica

Osijek, 2018.

SADRŽAJ

1. UVOD 1
   1.1. Zadatak završnog rada 1
2. OSNOVE UNITYJA I OSTALI KORIŠTENI PROGRAMI 2
3. Nastanak Unityja 2
   3.2. Licence Unityja 2
   3.3. Višeplatformska podrška 3
   3.4. Sučelje Unityja 4
   3.5. Ostali korišteni programi 7
      3.5.1. Microsoft Visual Studio 7
      3.5.2. Tiled i Tiled2Unity 8
      3.5.3. Piskel 8
4. DVODIMENZIONALNI POGLED NA UNITY 9
5. Vektori u Unityju 9
   5.2. GameObject 9
   5.3. Scene 10
   5.4. Kamera 10
   5.5. Animator i upravljanje animacijama 11
   5.6. Grafičke sličice i listići 12
   5.7. Skriptiranje 13
   5.8. Dvodimenzionalne fizičke veličine 13
      5.8.1. Rigidbody 2D 13
      5.8.2. Collider 2D 14
   5.9. Zvukovi 15
6. PRIMJENA UNITYJA U RAZVOJU IGRE 16
7. Početni izbornik 16
   7.2. Lik ..... **Pogreška! Knjižna oznaka nije definirana.**
      7.2.1. Skupljanje bodova i najbolji rezultat 20
   7.2.2. Gubljenje i skupljanje životnih bodova 22
      7.2.3. Uklanjanje lika 26
   7.3. Protivnik 27
   7.4. Glavni zaslon 28
      7.4.1. Prikaz skupljenih bodova i najboljeg rezultata 29
      7.4.2. Prikaz životnih bodova 29
      7.4.3. Prikaz minijaturne mape 30

7.5. Izbornik stanke 31
7.6. Mape 32
7.7. Zvukovi 34
8. ZAKLJUČAK 36
LITERATURA 37
SAŽETAK 40
ŽIVOTOPIS 41

1. UVOD

S početkom 21. stoljeća, industrija i tržište računalnih igara su se strahovito poboljšale. Postoji veliki broj tvrtki koje razvijaju razne žanrove i tipove računalnih igara, kao i igara za konzole poput PlayStationa i Xboxa. U današnje vrijeme, povećava se razvoj mobilnih igara, te zbog velike količine mobilnih uređaja i besplatnih igara, ljudi ih koriste.

Ima puno programa za razvoj igara (engl. Game Engine) pomoću kojih bilo tko može naučiti razvijati ili besplatno razviti vlastitu igru za bilo koju platformu. Unity podržava korištenje više programskih jezika, te omogućuje razvoj igara za više platformi poput računala, mobilnih uređaja ili konzola.

Završni rad sadrži pet poglavlja, a to su uvod, osnove Unityja i ostali korišteni programi, dvodimenzionalni pogled na Unity, primjena Unityja u razvoju igre i zaključak.

Poglavlje osnove Unityja i ostali korišteni programi će sadržavati osnovne značajke i elemente Unity programskog okruženja i ostale programe koje će se koristiti u završnom radu zbog toga što Unity neće biti jedini program koji će se koristiti. U poglavlju dvodimenzionalni pogled na Unity će biti opisane bitnije značajke, alati i dijelovi Unityja koje će biti potrebne za stvaranje dvodimenzionalne igre s horizontalnim pomicanjem. U poglavlju primjena Unityja u razvoju igre će se primijeniti ti opisani alati, te će se prikazati stvaranje dvodimenzionalne igre s horizontalnim pomicanjem.

1.1. Zadatak završnog rada

Zadatak završnog rada je razvijanje dvodimenzionalne igre s horizontalnim pomicanjem koja će biti pisana u programskom jeziku C#. Igra neće biti rađena samo u Unityju, pa će se i objasniti povezanosti i s drugim korištenim programima.

2. OSNOVE UNITYJA I OSTALI KORIŠTENI PROGRAMI

Unutar ovog poglavlja opisuju se osnove Unityja, odnosno kada je razvijen, tko ga je razvio, te se opisuju bitne značajke Unityja kao programa za izradu računalnih, mobilnih i igara za igraće konzole. Još se opisuju ostali programi i internetske stranice koje su korištene u izradi završnog rada.

3. NASTANAK UNITYJA

Prvu verziju Unityja je 8. lipnja 2005. razvio Unity Technologies koji se prvenstveno koristi za razvijanje trodimenzionalnih i dvodimenzionalnih video igara, te simuliranja za računala, igraće konzole i mobilne uređaje. Unity je višeplatformski program za razvoj video igara. [1]

Njegovo izvršenje (engl. runtime) je napisan u programskom jeziku C++ koji služi za stvaranje igre preko uređivača, a njegovo aplikacijsko programsko sučelje (engl. application programming interface) je napisano u C#. [2]

Unity je također program koji može obrađivati dvodimenzionalnu i trodimenzionalnu grafiku, te osim C#, još podržava Boo i JavaScript programske jezike. [1]

3.2. Licence Unityja

Kao što većina programa ima svoje vlastite besplatne i plaćene licence, odnosno dodatke koje omogućavaju svojim korisnicima, tako kod Unityja postoje četiri licence:

1. Unity Personal
2. Unity Plus
3. Unity Pro
4. Unity Enterprise

Unity Personal je potpuno besplatna i opremljena verzija, ali je u nekim slučajevima ograničen. [3]

Unity Plus je verzija dizajnirana za pojedince i timove koji žele što više korisnih i efektivnih. [3]

Unity Pro je novija i poboljšana verzija za profesionalne timove koji trebaju potpunu fleksibilnost. [3]

Unity Enterprise koriste veće organizacije i poduzeća želeći izvorne kodove i podršku prilikom izrade igara. [3]

3.3. Višeplatformska podrška

Na svijetu postoji velika količina različitih platformi preko kojih je korisnicima omogućeno igranje različitih tipova i žanrova igara. Korištenje Unityja omogućava razvoj igara na nešto više od dvadeset pet platformi, a taj broj se sve više povećava. [4]

Unity se može podijeliti prema pet područja razvoja višeplatformskih igara, a to su:

1. Unity za mobilne uređaje
2. Unity za virtualnu stvarnost
3. Unity za računala
4. Unity za konzole
5. Unity za Web. [4]

Pošto postoji velika količina mobilnih uređaja, postoji i veliki broj osoba koji koriste mobilne uređaje u svrhu igranja mobilnih igara. Unity je jedan od najpoznatijih programa za razvoj takvih igara.

Virtualna stvarnost (engl. virtual reality) je prikazivanje virtualnog okruženja u očima korisnika, odnosno simulacija virtualnog okruženja. Takva ideja tehnologije se počela razvijati 2013. kada je Palmer Luckey spojio nekoliko tehnologija u jednu. [5] Postoji nekoliko uređaja koji koriste

ovakvu tehnologiju, primjerice Steam VR, PlayStation VR, Oculus VR. Danas se Unity često upotrebljava za razvoj igara namijenjene takvim uređajima.

U Unity programskom okruženju najlakše je razviti igru za računala i Web zbog toga što se danas računalo i internet najviše koristi. Zbog snažnije razvijenosti industrije i tržišta video igara, velike tvrtke razvijaju igre za više platformi, posebno za računala, PlayStation i Xbox.

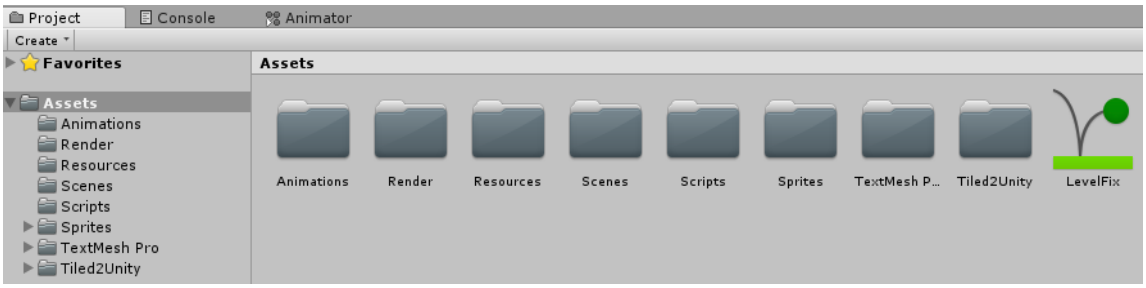
Igre za Web se mogu razviti pomoću WebGL-a, aplikacijskog programskog sučelja JavaScript-a koji omogućava korištenje dvodimenzionalnih i trodimenzionalnih grafika preko bilo kojeg internetskog preglednika. [6]

3.4. Sučelje Unityja

Sučelje Unityja (engl. Unity Editor), u sebi sadržava brojne prozore i potprozore koji omogućuju jednostavnije korištenje alata prilikom razvoja različitih tipova igara. Osim toga, još nudi drag-and-drop funkcionalnost, te skriptiranje kodova koristeći C#. [1] Najčešće korišteni prozori sučelja Unityja su:

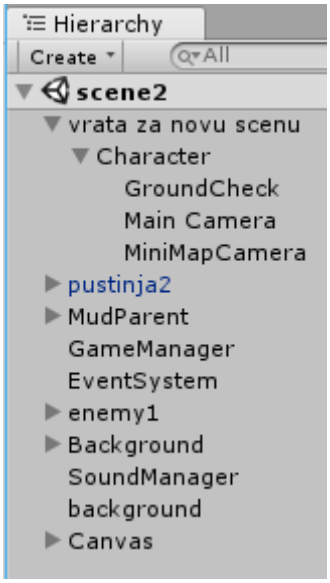
1. Projektni prozor (Slika 2.1.)
2. Prozor hijerarhije (Slika 2.2.)
3. Prozor inspektora (Slika 2.3.)
4. Pogled scene (Slika 2.4.)
5. Prozor igre (Slika 2.5.) [7]

Pomoću projektnog prozora (engl. project browser) korisnik može uvezati imovinu (engl. assets) u svoj Unity projekt i može joj pristupiti i izmjenjivati ju u bilo kojem trenutku. Imovina može biti bilo što vezano za projekt, poput slike, zvuka, scene ili čak skripte koja je pisana u C#. [8]



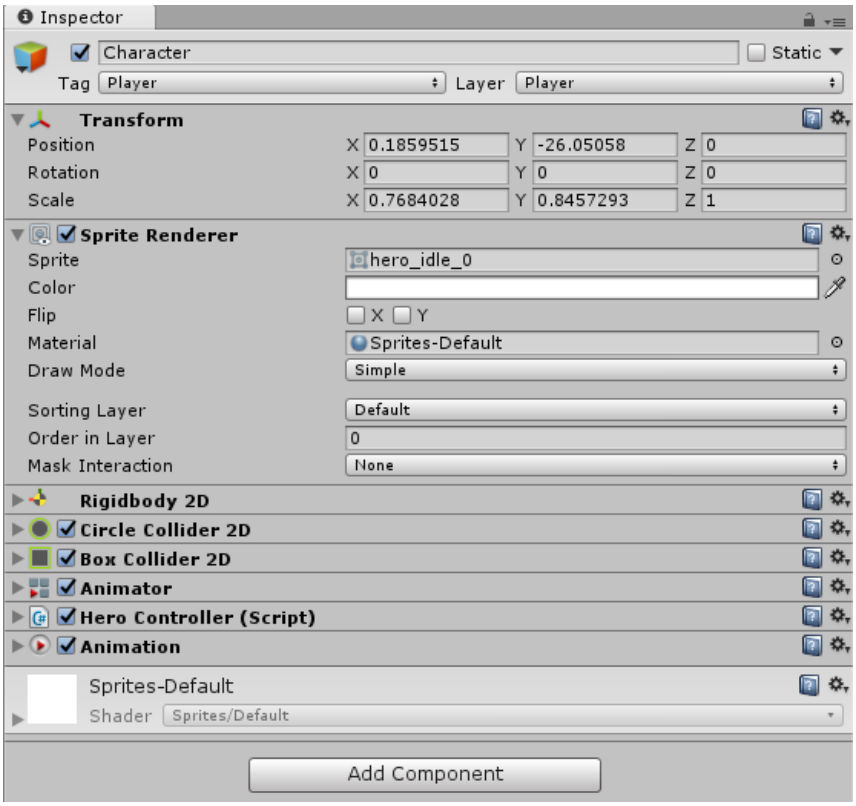
Slika 2.1. Prikaz projektnog prozora sučelja Unityja

U prozoru hijerarhije (engl. hierarchy) se nalaze svi predmeti (engl. GameObject) koji se nalaze u sceni. Povezan je s prozorom „Project“ na način se može pretvoriti imovina u predmet drag-and-drop funkcijom. Također, prozor hijerarhije koristi koncept roditeljstva (engl. parenting) koji omogućava da neki predmet bude dijete nekom drugom predmetu. [9]



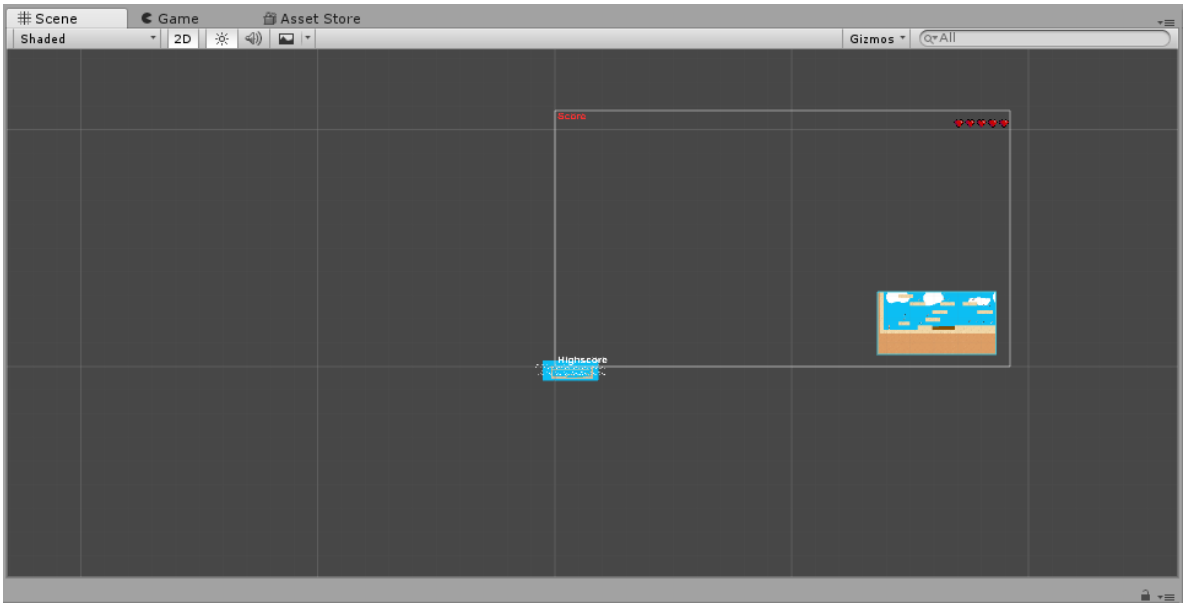
Slika 2.2. Prikaz prozora hijerarhije i koncepta roditeljstva

U prozoru inspektora (engl. inspector) se mogu izmijeniti detalji o predmetima, te prikazuje detaljnije informacije i pregled tih predmeta. Preko tog prozora se može izmijeniti bilo što objekt sadrži i mogu se dodati određeni sastavni dijelovi (engl. components) koji su potrebni prilikom razvoja igre. [10]



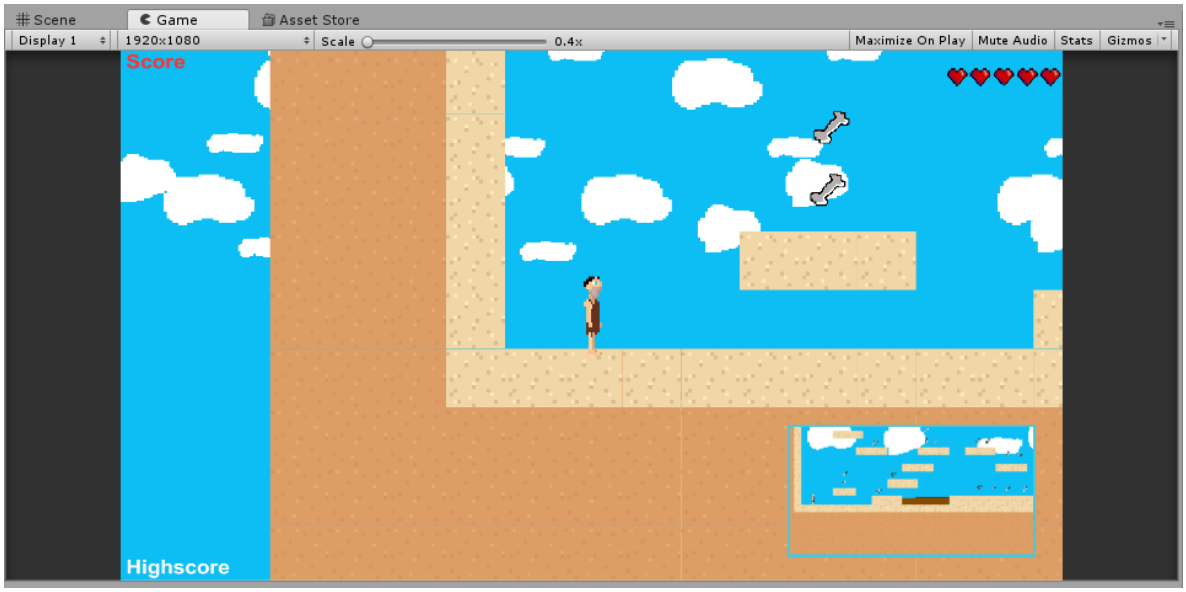
Slika 2.3. Prikaz prozora inspektora

Pogled scene (engl. scene view) je pogled u kojem se na brži i efikasniji način razni predmeti, poput igrača, protivnika, kamere, mogu dodavati, mijenjati ili nešto slično. Još se može koristiti za premještanje predmeta, bez da se određuje položaj u prozoru inspektora. [11]



Slika 2.4. Prikaz pogleda scene

Pogled igre je ono što prikazuje kamera prilikom razvoja igre. Donosi pregled kako će igra izgledati nakon završetka (engl. build).



Slika 2.5. Prikaz pogleda igre

3.5. Ostali korišteni programi

Pošto Unity nije bio jedini program prilikom razvoja akcijske igre s horizontalnim pomicanjem, bitno je spomenuti i ostale programe koji su se koristili prilikom izrade igre.

3.5.1. Microsoft Visual Studio

Da bi se neka igra mogla razviti, potrebno je znati programski jezik pomoću kojeg se može omogućiti kretanje predmeta, premještanje teksta ili nešto slično. Unity u sebi ima ugrađeno integrirano razvojno okruženje (engl. integrated development environment) zvano MonoDevelop. Omogućava pisanje izvornih kodova, odnosno skripti. Ali Unity ima mogućnost korištenja vanjskih alata, te se zbog toga koristio Microsoft Visual Studio.

Microsoft Visual Studio je integrirano razvojno okruženje koje je razvio Microsoft i koristi se za pisanje računalnih programa, kao i internetskih stranica, aplikacija i slično. [12] Za Unity

predstavlja bolje razvojno okruženje od MonoDevelop-a koje ima razne alate i mogućnosti za lakše i efikasnije pisanje izvornih kodova, odnosno skripti za razvoj igara.

3.5.2. Tiled i Tiled2Unity

U nekim slučajevima, prilikom razvoja dvodimenzionalnih igara, dolazi do problema kada se žele spojiti razni predmeti, odnosno teksture kako bi se stvorio određeni nivo s raznim preprekama. Kako bi se mogla lakše napraviti takva mapa, koristi se program „Tiled“, a za prebacivanje te mape u Unity, koristi se program „Tiled2Unity“.

Program „Tiled“ je uređivač dvodimenzionalnih nivoa, odnosno mapa, koji omogućuje puno lakši razvoj igre. Osim što služi za stvaranje mapa, sadrži puno mogućnosti poput zamjene slike ili čak dodavanje sudarivača (engl. collision) pločicama teksture. [13]

„Tiled2Unity“ je besplatni program koji omogućuje lakši izvoz mape s „.tmx“ nastavkom u Unity projekt i pomoću njega se brže i efikasnije može razviti igru. [14]

3.5.3. Piskel

„Piskel“ je internetska stranica koja omogućuje uređivanje i izradu animiranih sličica. Potpuno je besplatna usluga za sve korisnike koji ju žele koristiti. [15]

4. DVODIMENZIONALNI POGLED NA UNITY

Budući da je zadatak završnog rada bio razvoj dvodimenzionalne akcijske igre s horizontalnim pomicanjem, u ovom poglavlju su objašnjeni bitniji dijelovi Unityja u vezi s izradom te dvodimenzionalne igre.

5. VEKTORI U UNITYJU

U elementarnoj matematici i fizici, vektor najčešće označava veličinu koja ima iznos, smjer i orijentaciju i zadovoljava pravila vektorskog računa. Takav se opis odnosi na veličine u trodimenzionalnom prostoru, koji u matematici najbolje opisuje tzv. Euklidski prostor. [16]

Dvije najčešće korištene strukture u Unityju su „Vector2“ i „Vector3“. Vector2 je struktura koja opisuje, odnosno daje položaj dvodimenzionalnih vektora ili predmeta. [17] Vector3 opisuje trodimenzionalne predmete, odnosno daje njihov položaj, a čak i sadrži funkcije za rad nad vektorskim operacijama. [18]

5.2. GameObject

Predmet u Unityju (GameObject) je objekt koji predstavljaju sve što se nalazi u projektu. To mogu biti likovi, tekstovi ili nešto slično. Svu imovinu, koja se nalazi u projektnom prozoru Unity razvojnog okruženja, premještanjem u prozor hijerarhije dobije se takav predmet.

Predmetima se može mijenjati razna svojstva u prozoru inspektora poput položaja, veličine, a i mogu im se dodati brojni sastavni dijelovi.

5.3. Scene

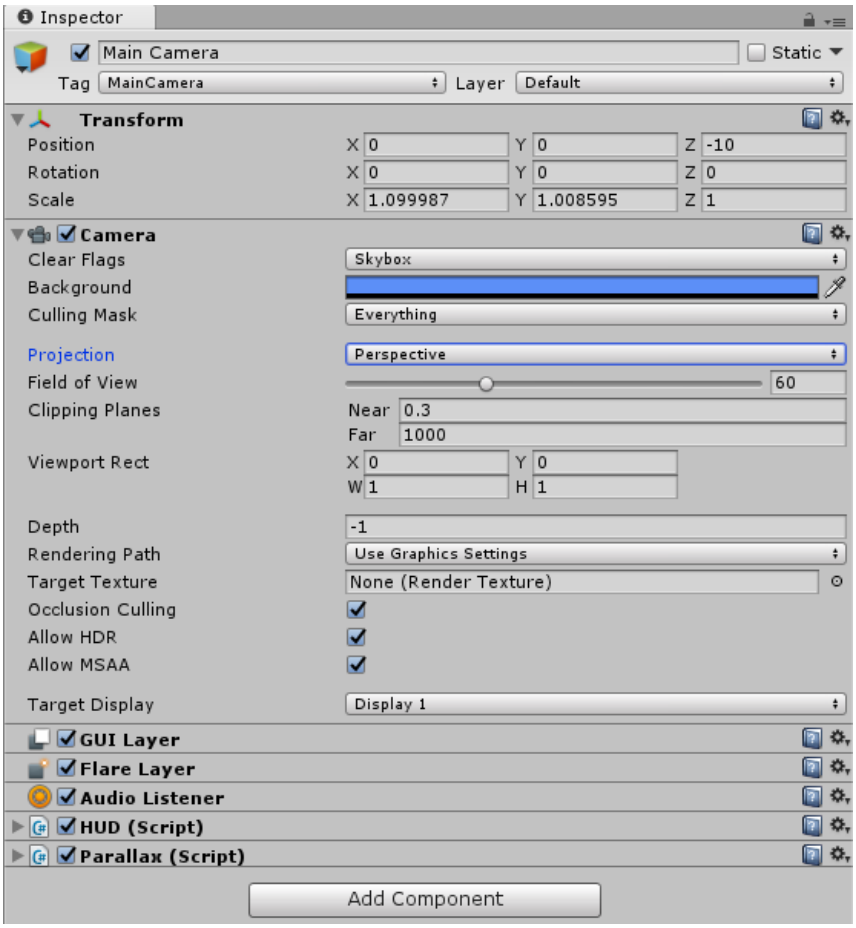
Scene u Unityju predstavljaju okruženja koja sadržavaju svu imovinu, odnosno predmete u određenom projektu. U svakoj sceni se prikupljaju, uređuju, te spajaju podaci kako bi se igra mogla razviti.

Svaka scena u projektu može imati svoje jedinstveno ime, a i može predstavljati svoj vlastiti nivo koji se pomoću programskih jezika mogu spojiti i predstaviti igru.

5.4. Kamera

Kamera je uređaj, odnosno alat u Unityju preko kojeg igrač može vidjeti perspektivu virtualnog svijeta. Prostor točke gledanja ovisi o kameri. Donji lijevi dio kamere je (0, 0), a gornji desni je (1, 1). Preko „z“ pozicije se može odrediti udaljenost kamere. [19]

U završnom radu korištene su dvije vrste projektiranje. Kada je uključen ortografski način, kamera je definirane preko ortografske veličine (engl. orthographic size). Takva je veličina je polovična veličina vertikalnog volumena kamere. [20] A kada je uključen perspektivni način, kamera je definirana preko vidnog polja (engl. field of view), a takvo se polje mjeri u stupnjevima. [21]



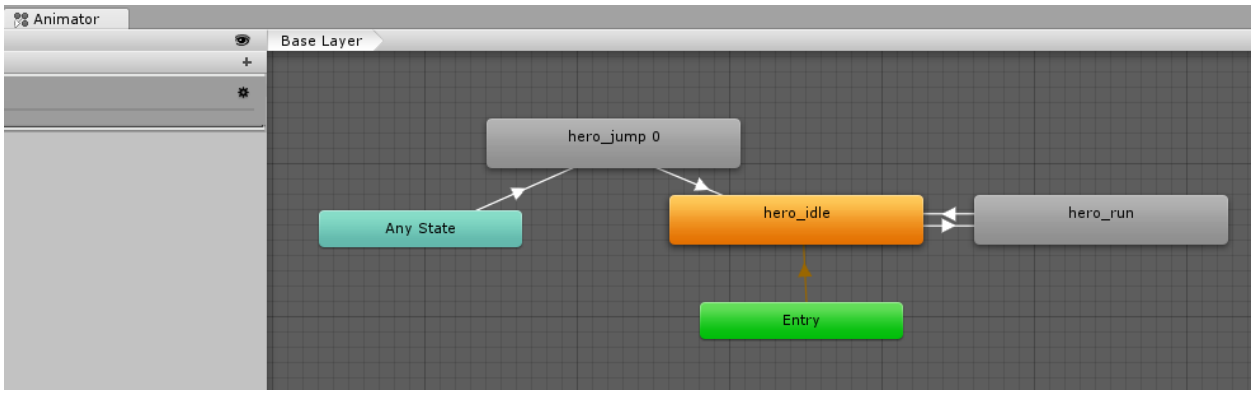
Slika 3.1. Inspektor glavne kamere

5.5. Animator i upravljanje animacijama

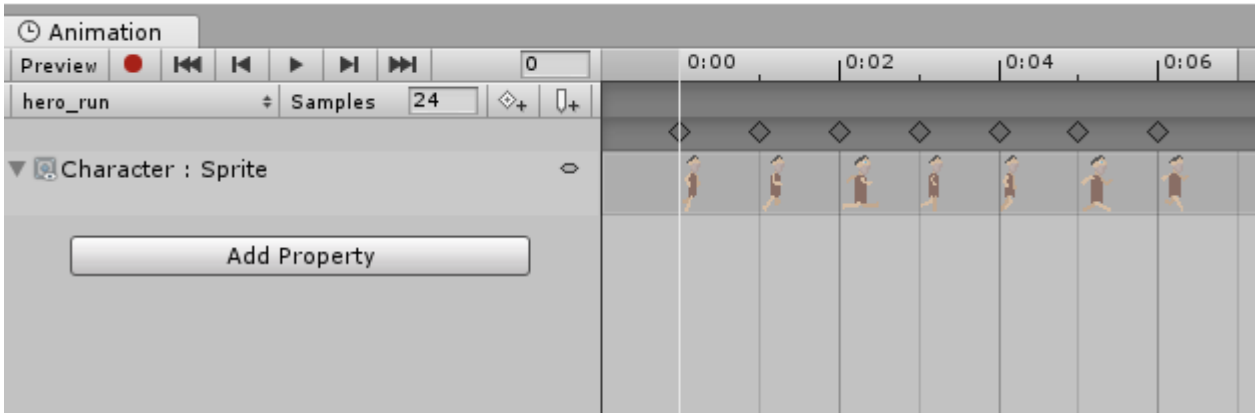
Razvijanjem igre, razni likovi ili predmeti se moraju kretati, točnije hodati, trčati ili skakati. Da bi im bilo omogućeno takvo što, moraju biti animirani.

U Unityju, takve animirane sličice se pokreću preko animacijskih sastavnih dijelova, što zapravo daje olakšanje prilikom razvoja igre.

U animacijskom prozoru (engl. animation window) programa Unity mogu se stvoriti kratki animirani isječci tako što se spoje više grafičkih sličica u jednu, a prozorom animatora (engl. animator window) se omogućuje kretanje lika uz pomoć određenih izvornih kodova.



Slika 3.2. Prikaz prozora animatora



Slika 3.3. Prikaz animacijskog prozora

5.6. Grafičke sličice i listići

Grafičke sličice su se oduvijek koristile prilikom razvijanja dvodimenzionalnih igara. U klasičnim igrama, pa čak i u modernim. [22]

Grafička sličica (engl. sprite) je samo jedna sličica koja se koristi razvojem igre, dok je Grafički listić (engl. sprite sheet) skup različitih sličica pomoću kojih se može dobiti kratki animirani isječak.

Unity razvojno okruženje sadrži i uređivač grafičkih listića (engl. sprite editor) koji može te listiće odvojiti i stvoriti više sličica

5.7. Skriptiranje

Kako bi se stvorio određeni računalni program ili čak razviti igra u Unityju, potrebno je znati određene programske jezike poput C#.

Pisanje programa (engl. scripting) je izrazito važno u Unityju zbog toga što se na taj način može napisati ono što se želi. Potrebno je u svim tipovima igara, pa čak i u najjednostavnijim. S time se može postići kontrola nad fizičkim ponašanjima predmeta. [23]

5.8. Dvodimenzionalne fizičke veličine

Najjednostavnije rečeno, dvodimenzionalne fizičke reference, odnosno sastavni dijelovi, daju predmetima fizičke mogućnosti u određenim Unity projektima. Dva sastavna dijela korištena u radu su dvodimenzionalno kruto tijelo (rigidbody 2D) i sudarivač (collider 2D).

5.8.1. Rigidbody 2D

Sastavni dio dvodimenzionalnog krutog tijela postavlja predmete pod kontrolu preko fizičke jezgre. Ono može pomicati objekte XY ravninom, te ih okretati po osi okomitoj na tu ravninu. [24]

Jezgra dvodimenzionalne fizike omogućuje kretanje dvodimenzionalnih sudarivača i omogućuje interakciju između njih, odnosno fizička jezgra komunicira s pokretima sudarivača. [24] Dvodimenzionalno kruto tijelo upravo služi za takvu komunikaciju i povezanost između sudarivača. [24]

Dodavanjem dvodimenzionalnog krutog tijela na neki predmet, omogućuje mu kretanje na uvjerljiv fizički način, te se upravo zbog takvog sastavnog dijela, igrač prilikom igranja osjeća zadovoljnije.

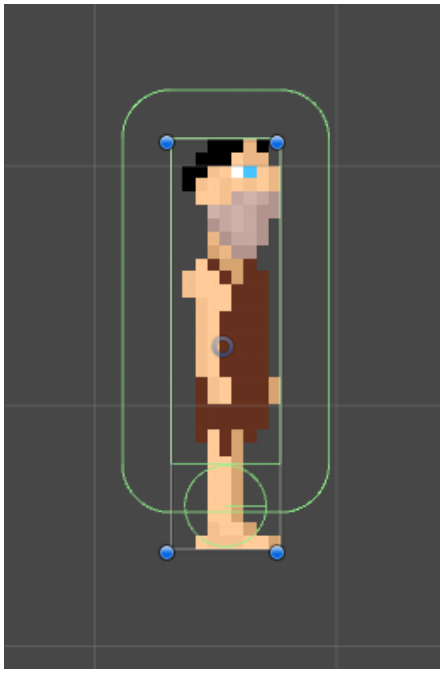
5.8.2. Collider 2D

Dvodimenzionalni sudarivač je nevidljivi sastavni dio koji opisuje oblik predmeta s ciljem fizičkog sudaranja. [25] U završnom radu su korištene dvije vrste ovakvih komponenata, a to su:

1. Kružni dvodimenzionalni sudarivač
2. Pravokutni dvodimenzionalni sudarivač

Kružni dvodimenzionalni sudarivač (Circle Collider 2D) kružnog je oblika, te je definiran po određenom radiusu.

Pravokutni dvodimenzionalni sudarivač (Box Collider 2D) pravokutnog je oblika, te je definiran po određenoj visini i dužini.



Slika 3.1. Prikaz dvodimenzionalnih sudarivača lika

5.9. Zvukovi

U današnje vrijeme niti jedna igra ne funkcionira bez zvuka. Ako se za primjer uzmu igre za više igrača, gotovo su nemoguće za igrati bez zvuka. Razvojem dvodimenzionalnih igara potrebno je umetnuti zvuk ili nekakav zvučni efekt poput pozadinskog zvuka, zvuka za skupljanje novčića, zvuka za skakanje i slično.

Kako bi simulirao pozicijski učinak, Unity zahtijeva nastanak zvuk iz sastavnog dijela izvornog zvuka (engl. Audio Source) koja je postavljena na predmete. [26] Te emitirane zvukove tada odabire sastavni dio slušača zvuka (engl. Audio Listener) koja je postavljena na drugi predmet poput glavne kamere. [26] Unity tada može simulirati učinak od izvorne udaljenosti i pozicije. [26]

6. PRIMJENA UNITYJA U RAZVOJU IGRE

Budući da je ideja završnog rada bila izrada dvodimenzionalne igre s horizontalnim pomicanjem u Unity razvojnom okruženju, u ovom je poglavlju objašnjena izrada takve igre.

Tema igre je bila povijesna, točnije upravljanje pračovjekom prolazeći mapama i u isto vrijeme skupljati bodove, gubiti snagu, skupljati životne bodove i uništavati protivnike. U igri postoje tri nivoa, odnosno mape. Prva mapa je prašuma, druga je pustinja, a treća je pećina.

7. POČETNI IZBORNIK

Svaka razvijena igra sadrži početni izbornik, odnosno mjesto gdje sve počinje. U izborniku se nalaze razne tipke s raznim mogućnostima poput tipke za početak igre, tipke za izlaz i slično.

Izbornik sadržava tri tipke, a to su tipka pokretanja nove igre (engl. play button), tipka pomoći (engl. how to play button) i tipka izlaza (engl. quit button). Svaka od tih tipki radi svoj zadatak koji je napisan u programskom jeziku C#.

Slika 4.1. Početni izbornik igre [27]

Tipka pokretanja nove igre ima zadatak da prebaci početni izbornik na drugu scenu, odnosno da pokrene novu igru. Osim toga, liku daje sve životne bodove, ali to će biti objašnjeno kasnije.

```
01.  public void PlayGame()
02.      {
03.          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
04.
05.          PlayerPrefs.SetInt("PlayerCurrentLives", maxhealth);
06.      }
```

Slika 4.2. Funkcija igranja igre početnog zaslona

Zadatak tipke pomoći je da isključi vidljivost ploče početnog zaslone i uključi vidljivost svoje ploče u kojoj su objašnjene naredbe igre.

Slika 4.3. Ploča tipke pomoći

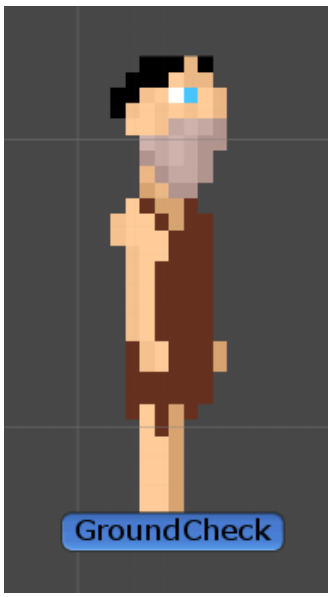
Tipka izlaza ima zadatak da izađe iz igre i da obriše sve spremljene preferencije igrača (engl. player preferences) poput količina skupljenih bodova, najbolji rezultat.

```
01.  public void QuitGame()
02.      {
03.          Application.Quit();
04.          PlayerPrefs.DeleteAll();
05.      }
```

Slika 4.4. Funkcija izlaska iz igre početnog zaslona

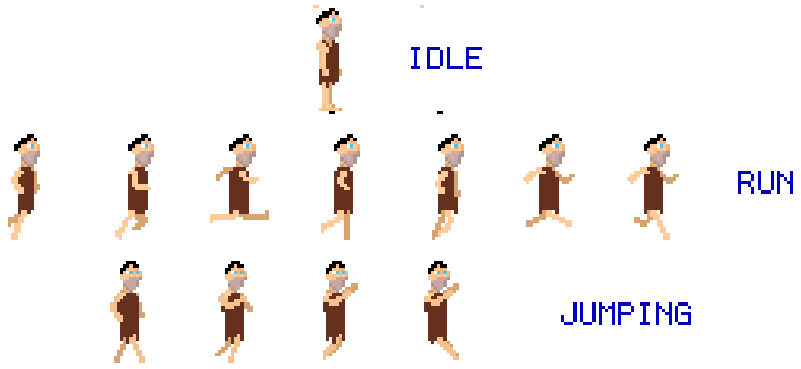
7.2. Lik

Kao što svaka današnja igra ima razne likove, bili oni dvodimenzionalni ili trodimenzionalnu, ova razvijena igra ima svog lika, odnosno glavnog junaka. Taj junak je zapravo jedan pračovjek kojeg igrač upravlja preko tipkovnice i na taj način skuplja bodove, odnosno prelazi igru.



Slika 4.5. Prikaz glavnoj junaka

Nakon što se pritisne tipka pokretanja nove igre na početnom izborniku, odmah se može uočiti tog glavnoj junaka pomoću kojeg se igra može igrati. Za glavnoj junaka postoje tri grafička listića preko kojih su se dobili animirani isječci tako da nastane ljepši ugođaj prilikom upravljanja.



Slika 4.6. Tri grafička listića glavnog junaka [28]

Nakon što su se dobili animirani isječci, pisanjem koda, odnosno određene skripte za upravljanje glavnog junaka, pritiskom određene tipke na tipkovnici, glavni junak se pokreće kao i taj animirani isječak.

Za pokretanje lika, u skripti su definirane razne varijable koje omogućuju to pokretanje. Primjerice, sadrži varijablu za maksimalnu brzinu koja omogućuje kretanje lika. Sadrži još varijablu sile skakanja koja omogućuje skakanje lika.

Također, stvoreni lik u sebi ima predmet provjere tla omogućuje da se lik kreće samo po tlu.

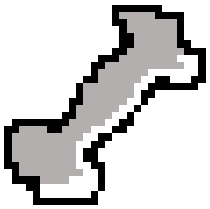
Lik se pokreće preko tipkovnice tipkama „a“ i „d“, a skakanje tipkom razmaka ili „w“. Postoji funkcija zaokretanja (Flip) koja zaokreće lik u smjeru njegovog kretanja.

```
public float maxSpeed = 10f;
bool facingRight = true;
Animator anim;
bool grounded = false;
public Transform groundCheck;
float groundRadius = 0.2f;
public LayerMask whatIsGround;
public float jumpForce = 200f;

void Start ()
{
    anim = GetComponent<Animator>();
    gm = GameObject.Find("GameManager").GetComponent<GameManagerScript>();
    curHealth = PlayerPrefs.GetInt("PlayerCurrentLives");
}

void FixedUpdate ()
{
    grounded = Physics2D.OverlapCircle(groundCheck.position, groundRadius, whatIsGround);
    anim.SetBool("Ground", grounded);
    anim.SetFloat("vSpeed", GetComponent<Rigidbody2D>().velocity.y);
    float move = Input.GetAxis("Horizontal");
    anim.SetFloat("Speed", Mathf.Abs(move));
    GetComponent<Rigidbody2D>().velocity = new Vector2(move * maxSpeed, GetComponent<Rigidbody2D>().velocity.y);
    if (move > 0 && !facingRight)
    {
        Flip();
    }
    else if (move < 0 && facingRight)
    {
        Flip();
    }
}

void Update()
{
    if(grounded && (Input.GetKeyDown(KeyCode.Space) || Input.GetKeyDown(KeyCode.W)))
    {
        SoundManager.PlaySound("jump");
        anim.SetBool("Ground", false);
        GetComponent<Rigidbody2D>().AddForce(new Vector2(0, jumpForce));
    }
    if (curHealth > maxHealth)
    {
        curHealth = maxHealth;
    }
    if(curHealth <= 0)
    {
        Die();
    }
}

void Flip()
{
    facingRight = !facingRight;
    Vector3 theScale = transform.localScale;
    theScale.x *= -1;
    transform.localScale = theScale;
}
```

Slika 4.7. Prikaz dijela skripte upravljanja glavnog junaka

7.2.1. Skupljanje bodova i najbolji rezultat

Kao što skoro svaka dvodimenzionalna igra s horizontalnim pomicanjem ima implementiran koncept skupljanja bodova, cilj ove igre je skupiti što više bodova, odnosno kostiju koje su napravljene na internetskoj stranici Piskel.



Slika 4.8. Prikaz kosti, tj. bodova

Pošto smo predmetima lika i kostiju dodali sastavne dijelove dvodimenzionalnih sudarivača, Unity razvojno okruženje može lakše zaključiti o čemu se radi. Sadrži i funkciju okidača (isTriggered) koja nam govori ponaša li se sudarivač kao okidač ili ne.

```
01.  void OnTriggerEnter2D(Collider2D other)
02.  {
03.      if (other.tag == "Bone")
04.      {
05.          SoundManager.PlaySound("coin");
06.          Destroy(other.gameObject);
07.          gm.BonesCollected();
08.      }
09.  }
```

Slika 4.9. Funkcija pronalaska kosti

Također, nekim objektima su dodane oznake zbog lakšeg korištenja. Na primjer, kostima je dodana oznaka „Bone“, a funkcijom uništavanja (Destroy) uništava se određeni predmet.

Još se može vidjeti funkcija skupljenih kostiju (BonesCollected). Ona nam govori da će se skupljeni bodovi povećati za jedan. Ona u sebi još sadrži i prikazivanje skupljenih bodova na zaslonu, ali to će biti objašnjeno naknadno.

```
01.  public void BonesCollected()
02.  {
03.       bones += 1;
04.       bonesText.text = "Score: " + bones;
05.       highscoreText.text = "Highscore: " + highscore;
06.  }
```

Slika 4.10. Funkcija skupljanja bodova

Igra sadržava i koncept najboljeg rezultata. Ako prilikom skupljanja bodova lik nestane, odnosno ako se objekt lika uništi, spremaju se skupljeni bodovi, te se promatra kada je taj broj bio najveći.

```
01.  if (PlayerPrefs.HasKey("Highscore"))
02.  {
03.       if(PlayerPrefs.GetInt("Highscore") < gm.bones)
04.       {
05.            PlayerPrefs.SetInt("Highscore", gm.bones);
06.       }
07.  }
```

Slika 4.11. Dio koda za pronalaženje najboljeg rezultata

7.2.2. Gubljenje i skupljanje životnih bodova

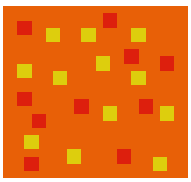
Lik ima svoju energiju, točnije pet životnih bodova (engl. health points) koje dobije prilikom pokretanje igre iz početnog izbornika. Nestaje nakon što izgubi svih pet, odnosno uklanja se taj predmet lika. Postoje dva slučaja u kojima on može izgubiti životne bodove, a to su:

1. Preko zamke
2. Preko protivnika

Prvi slučaj u kojem može izgubiti životne bodove je preko predmeta postavljenog na tlu preko kojih lik izgubi po jedan životni bod čim ih dodirne. Svaka mapa ima svoje zamke. U prvoj mapi se nalaze trnje, druga sadrži živi pijesak, a treća lavu.

Slika 4.12. Prikaz trnja prve mape

Slika 4.13. Prikaz živog pijeska druge mape



Slika 4.14. Prikaz lave treće mape

Ova tri predmeta imaju implementiran svoj sudarivač i čim ga lik dodirne, izgubi jedan životni bod funkcijom šteta (Damage) i odbacuje ga funkcijom odbacivanja (Knockback).

```
01. void OnTriggerEnter2D(Collider2D col)
02. {
03.     if (col.CompareTag("Player"))
04.     {
05.         player.Damage(1);
06.         StartCoroutine(player.Knockback(0.02f, 700, player.transform.position));
07.     }
08. }
```

Slika 4.15. onTrigger funkcija gubljenja jednog životnog boda

Funkcija štete prima jačinu udarca kao parametar te oduzima trenutne životne bodove za tu jačinu, u ovom slučaju oduzima za jedan životni bod. Funkcija sadrži animirani isječak u kojem se lik zacrveni kada mu se životni bodovi smanjuju.

```
01. public void Damage(int dmg)
02. {
03.     SoundManager.PlaySound("hurt");
04.     curHealth -= dmg;
05.     PlayerPrefs.SetInt("PlayerCurrentLives", curHealth);
06.     gameObject.GetComponent<Animation>().Play("redflash");
07. }
```

Slika 4.16. Funkcija količine udarca

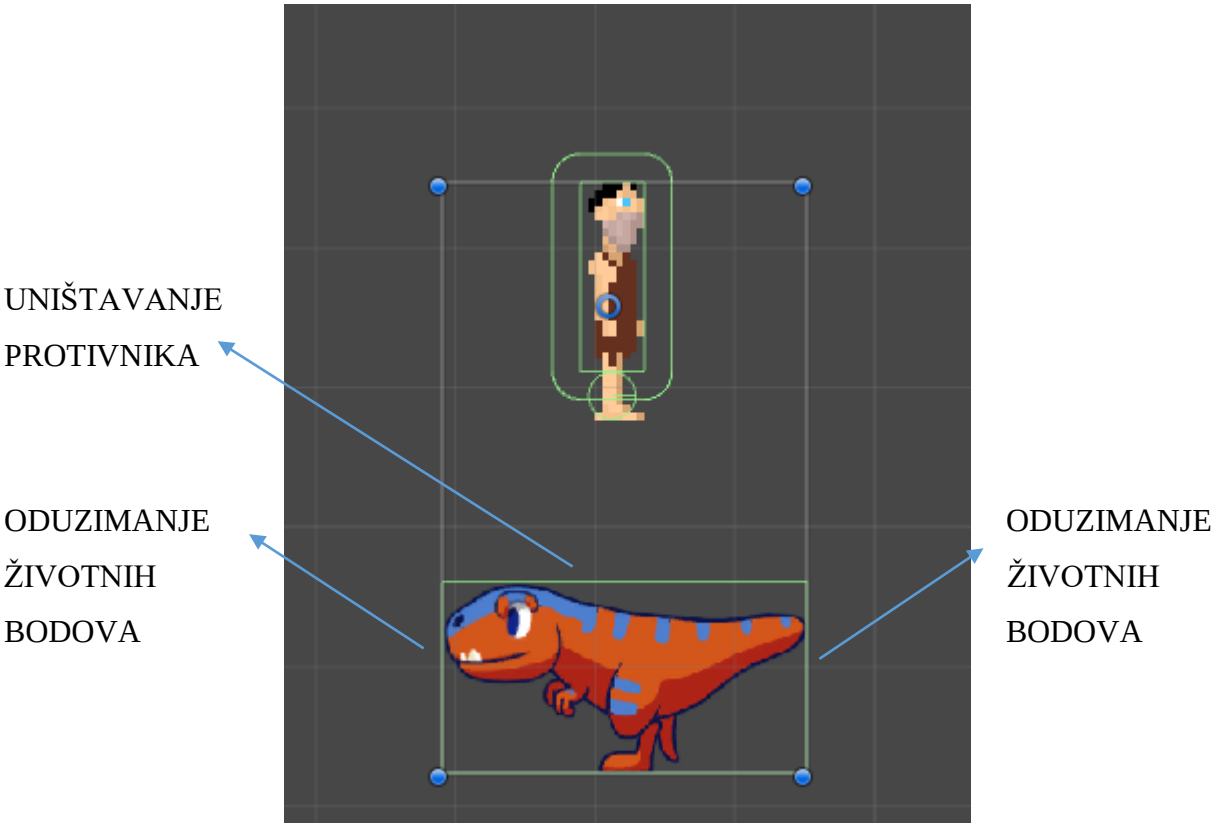
Funkcija odbacivanja odbacuje lik u određenom pravcu i određenom snagom.

```
01. public IEnumerator Knockback(float knockDur, float knockbackPwr, Vector3 knockDir)
02. {
03.       float timer = 0;
04.       while (knockDur > timer)
05.       {
06.       timer += Time.deltaTime;
07.       GetComponent<Rigidbody2D>().velocity = new Vector2(0, 0);
08.       GetComponent<Rigidbody2D>().AddForce(new Vector3(knockDir.x*-50, knockbackPwr, transform.position.z));
09.       }
10.    yield return 0;
11. }
```

Slika 4.17. Funkcija odbacivanja lika

Drugi slučaj gubljenja životnih bodova je preko protivnika. Kao i kod zamki, postoje tri vrste protivnika, dinosaurus u prvom nivou, divovska zmija u drugom, a medvjed u trećem.

Pošto predmet protivnika ima sastavni dio sudarivača, lik ako dotakne lijevi ili desni dio, gubi životne bodove, a ako dotakne gornji dio sudarivača, uništi predmet protivnika.



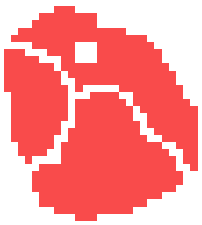
Skica 4.1. Prikaz načina uništavanja protivnika ili oduzimanja životnih bodova lika

Igrač dok upravlja likom, mora skočiti na gornji dio sudarivača koji je u skripti opisan na način da se očitava y os vektora. Uništavanje protivnika će biti objašnjeno kasnije. Također, postoji funkcija odbacivanja koja odbacuje lik u određenom pravcu i određenom jačinom.

```
01.  void OnCollisionEnter2D(Collision2D collision)
02.  {
03.        enemy enemy = collision.collider.GetComponent<enemy>();
04.        if(enemy != null)
05.        {
06.           foreach(ContactPoint2D point in collision.contacts)
07.           {
08.               if(point.normal.y >= 0.9f)
09.               {
10.                    Vector2 velocity = GetComponent<Rigidbody2D>().velocity;
11.                    velocity.y = 13;
12.                    GetComponent<Rigidbody2D>().velocity = velocity;
13.                    enemy.Hurt();
14.               }
15.               else
16.               {
17.                    StartCoroutine(Knockback(0.02f, 500, transform.position));
18.                    Damage(1);
19.               }
20.           }
21.        }
22.   }
```

Slika 4.18. Funkcija uništavanja protivnika ili oduzimanja životnih bodova lika

Gubljenjem energije, odnosno oduzimanjem životnih bodova, uvijek postoji mogućnost da se ti životni bodovi povećaju. Za povećanje u igri je potrebno naći komad mesa, točnije predmet koji liku povećava energiju za jedan životni bod.



Slika 4.19. Meso za povećanje životnog boda

Ovaj predmet, s oznakom „meat“ također ima svoj sastavni dio sudarivača. Aktivira se nakon što ju lik, odnosno igrač dodirne. Ali ako lik ima svih pet životnih bodova, tada se predmet samo uništi. Bez da mu se poveća energija.

```
01.  void OnTriggerEnter2D(Collider2D other)
02.  {
03.        if (other.tag == "meat")
04.        {
05.            SoundManager.PlaySound("meatbite");
06.            Destroy(other.gameObject);
07.            if (curHealth < 5)
08.            {
09.              curHealth += 1;
10.              PlayerPrefs.SetInt("PlayerCurrentLives", curHealth);
11.            }
12.        }
13.  }
```

Slika 4.20. onTrigger funkcija dodavanje životnih bodova

7.2.3. Uklanjanje lika

Nakon što lik izgubi svih pet životnih bodova, nestaje, odnosno premet se uklanja, te se trenutni nivo ponovno pokrene uz uvjet da se obriše količina skupljenih bodova na tom nivou.

```
01.  void Die()
02.  {
03.       SoundManager.PlaySound("death");
04.       if (PlayerPrefs.HasKey("Highscore"))
05.       {
06.           if(PlayerPrefs.GetInt("Highscore") < gm.bones)
07.           {
08.           PlayerPrefs.SetInt("Highscore", gm.bones);
09.           }
10.       }
11.       else
12.       {
13.           PlayerPrefs.SetInt("Highscore", gm.bones);
14.       }
15.           PlayerPrefs.SetInt("PlayerCurrentLives", maxHealth);
16.           SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
17.  }
```

Slika 4.21. Funkcija „Die“ za uklanjanje lika

Funkcija smrti (Die), koja osim što ponovno pokreće nivo, provjerava najbolji rezultat tako što ga uspoređuje s količinom skupljenih bodova prije smrti i zadnjeg najboljeg rezultata.

7.3. Protivnik

Kao što je već rečeno, igra ima tri vrste protivnika. Prvi nivo ima dinosauruse, drugi ima divovske zmije, a treći medvjede. Da bi ih se uništilo, igrač s likom mora skočiti na njih, ali ako dodirne njihovu lijevu ili desnu stranu, gubi životne bodove.

Svaki od tih protivnika ima svoj grafički listić pomoću kojih se dobio animirani isječak kako bi se dobio ljepši ugođaj prilikom igranja.

Slika 4.22. Grafički listić dinosaura [29]

Slika 4.23. Grafički listić divovske zmije [30]

Slika 4.24. Grafički listić medvjeda [31]

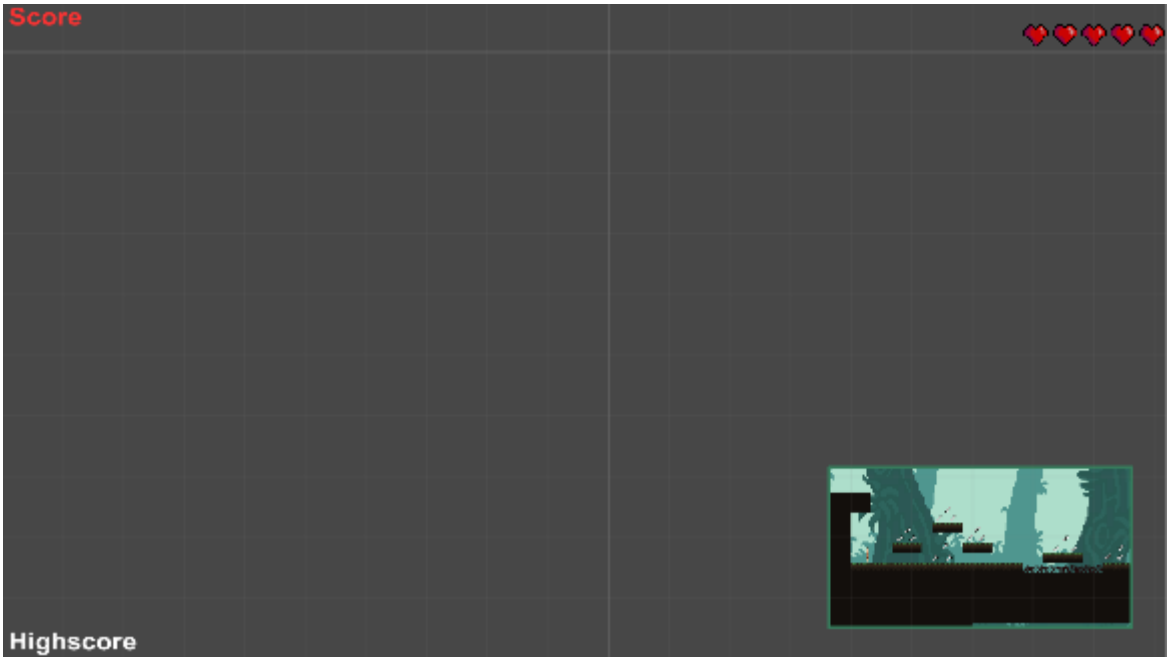
Prilikom igranja igre, odnosno upravljanja likom, vidljivo je da se protivnici kreću. Uvijek se kreću u jednom pravcu sve dok ne naiđu na rub sudarivača. Točnije provjerava je li ispred njega tlo ili nije. Ako nije tlo, protivnik se okrene za sto osamdeset stupnjeva. Za rotaciju u C# se koriste Eulerovi kutovi.

```
01.  if (!isgrounded)
02.  {
03.      Vector3 currRot = myTrans.eulerAngles;
04.      currRot.y += 180;
05.      myTrans.eulerAngles = currRot;
06.  }
```

Slika 4.25. Dio koda za provjeru tla

7.4. Glavni zaslon

Svakoj su igri potrebne informacije na zaslonu kako bi igrači bili obavješteni o nečemu što je vezano za tu igru i kako bi im se pružila određena pomoć. Taj prikaz informacija na zaslonu se naziva Glavni zaslon (eng. Head-up display), odnosno statusna traka.



Slika 4.26. Prikaz Glavnog zaslona

Igra se sastoji od tri dijela prikaza na glavnom zaslonu, a to su:

1. Prikaz skupljenih bodova i najboljeg rezultata
2. Prikaz životnih bodova
3. Prikaz minijaturne karte

7.4.1. Prikaz skupljenih bodova i najboljeg rezultata

Ovo su zapravo tekstualni predmeti koji prikazuju podatke o bodovima na zaslonu. Na početku količina skupljenih bodova i najbolji rezultat ostaju nula, ali čim igrač prilikom igranja skupi bod, automatski se prikaže na zaslonu. Najbolji rezultat ostaje nula sve dok se lik ne ukloni, odnosno nestane.

```
01. void Update ()
02. {
03.     bonesText.text = "Score: " + bones;
04.     highscoreText.text = "Highscore: " + highscore;
05. }
```

Slika 4.27. Funkcija koja prikazuje skupljene bodove i najbolji rezultat

Funkija ažuriranja (Update) se ažurira svaku novu sličicu, te na taj način se ti bodovi lakše mogu prikazivati na zaslonu. [32]

7.4.2. Prikaz životnih bodova

Kako bi se lakše moglo utvrditi koliko lik ima životnih bodova, koristi se traka životnih bodova koje je sastavljeno od onoliko srca koliko lik ima životnih bodova.

Slika 4.28. Prikaz grafičkog listića srca [33]

Koncept je vrlo jednostavan. Svako smanjenje životnog boda gubi se po jedno srce, a svako povećanje boda dobiva se jedno srce. Ako lik izgubi sve životne bodove, nivo se ponovno pokreće i doba se pet novih životnih bodova.

```
01.  void Update()
02.  {
03.       if (player.curHealth >= 0)
04.          HeartUI.sprite = HeartSprites[player.curHealth];
05.  }
```

Slika 4.28. Funkcija koja provjerava sve dok lik ima životnih bodova

7.4.3. Prikaz minijaturne mape

Prilikom igranja na velikim mapama postoji veliki problem s kamerama. Mape su ponekad male, a ponekad velike, te zbog toga igrač ne može vidjeti bitne dijelove mape. Kako bi se mogao riješiti takav problem i kako bi se igrač mogao lakše orijentirati na mapi i da zna gdje treba ići, uvode se minijaturne mape.

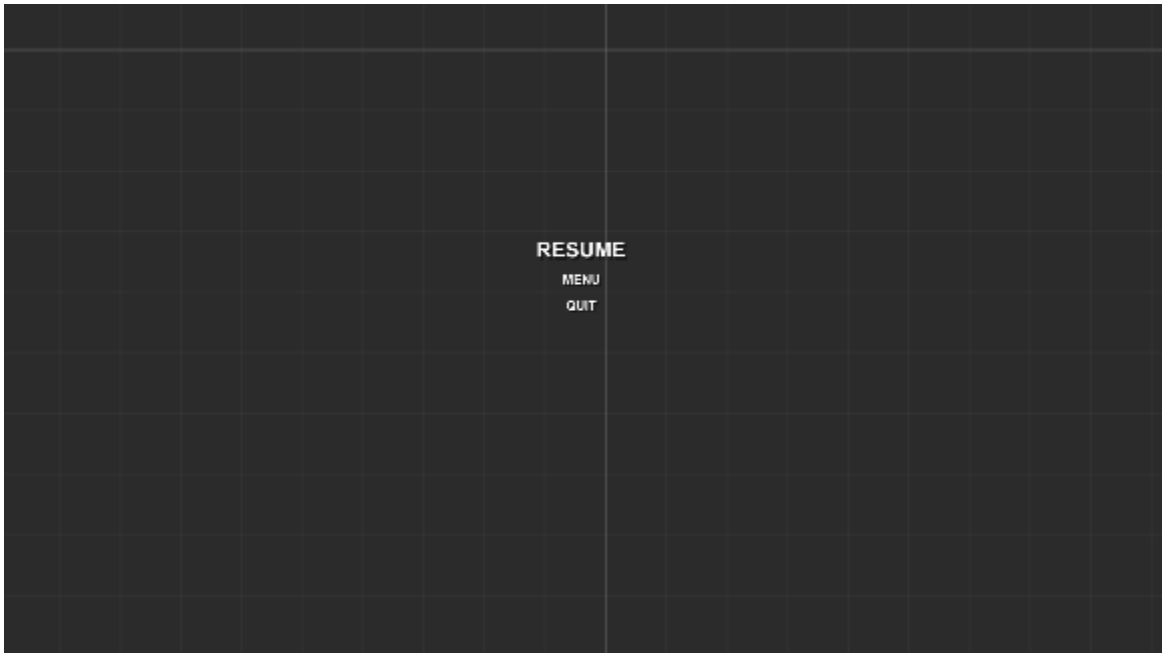
Takva mapa je zapravo druga kamera koja na zaslonu prikazuje dijelove mape koje igrač ne može vidjeti preko glavne kamare i na taj mu način to pruži pomoć prilikom igranja.

Slika 4.29. Prikaz minijaturne mape

Na slici 4.29., osim minijaturne mape, mogu se vidjeti i primjeri ostalih dijelova glavnog izbornika. Lik trenutno ima jedan životni bod, skupio je šest bodova, a najbolje rezultat sto dva.

7.5. Izbornik prilikom pauziranja (zaustavljanja)

Pritiskom tipke „Escape“, vrijeme igre se zaustavlja i dobiva se izbornik stanke. Taj izbornik sadrži tipku nastavljanja (eng. resume button), tipku početnog izornika (engl. menu button) i tipku izlaza (engl. quit button).



Slika 4.30. Prikaz izbornika stanke

Vidljivost izbornika stanke će biti isključena sve dok igrač ne pritisne tipku „Escape“. Kada igrač pritisne tipku „Escape“, vidljivost se uključuje. Tipka nastavljanja služi za nastavak igre, odnosno vrijeme se nastavi, a vidljivost izbornika stanke se isključuje.

Tipka početnog izbornika prebacuje trenutnu scenu na početni izbornik, a tipka izlaza izlazi iz igre i briše sve prikupljene podatke o liku.

```
01.  public void LoadMenu()
02.  {
03.        SceneManager.LoadScene(0);
04.        Resume();
05.  }
```

Slika 4.31. Funkcija početnog izbornika stanke

Ako igrač pritisne tipku „Escape“, vrijeme u igri će se zaustaviti. Upravo zbog toga je potrebno da i funkcija početnog izbornika nastaviti to vrijeme.

```
01.  public void QuitGame()
02.  {
03.      PlayerPrefs.DeleteAll();
04.      Application.Quit();
05.  }
```

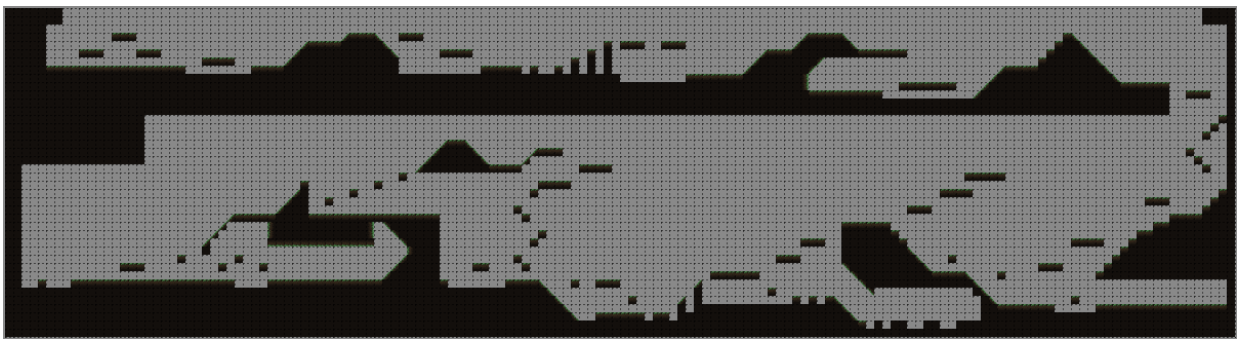
Slika 4.31. Funkcija izlaska iz igre izbornika stanke

7.6. Mape

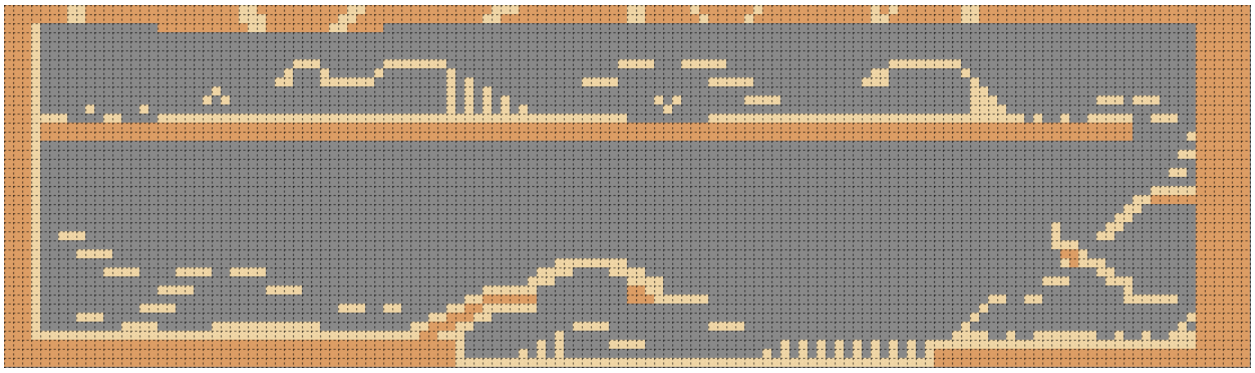
Kao što je već bilo spomenuto, postoji tri vrste mapa. Svaka mapa predstavlja po jedan nivo prilikom igranja igre i cilj je da se dođe do kraja svake mape.

Mape su rađene u već objašnjenim programima Tiled i Tiled2Unity koji su omogućili brži i jednostavniji način razvoja mapa.

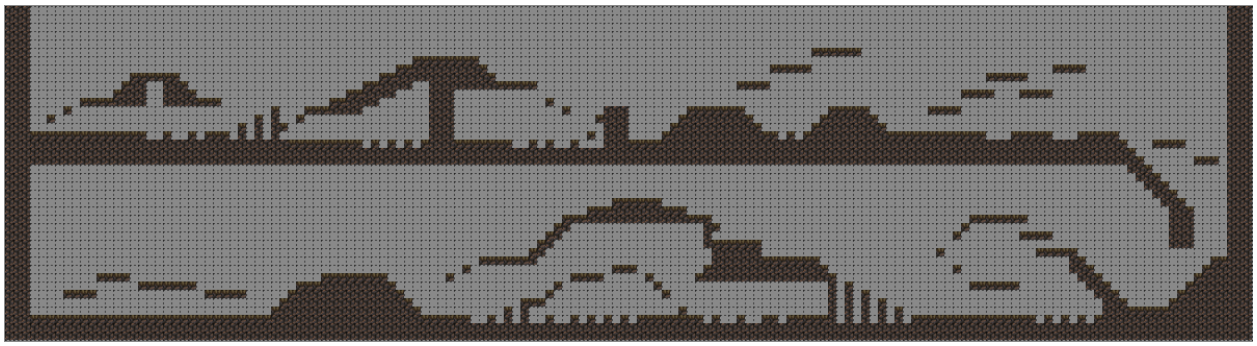
U prvoj mapi, lik prolazi kroz prašumu, bori se s dinosaurusima i preskače trnje. U drugoj mapi, prolazi kroz vruću pustinju, bori se s divovskim zmijama i ne želi upasti u živi pijesak. A u trećoj, bori se s medvjedima i preskače lavu.



Slika 4.31. Prikaz prve mape [34]



Slika 4.32. Prikaz druge mape [35]



Slika 4.32. Prikaz treće mape [36]

Na kraju svake mape postoji logorska vatra čija je funkcija da prebaci na sljedeći nivo, odnosno da spremi igru.

Slika 4.32. Prikaz logorske vatre [37]

Kako bi igrač otišao na idući nivo, potrebno mu je doći do logorske vatre i pritisnuti tipku „e“ koja mu to omogućuje.

Slika 4.33. Prikaz prilaska logorskoj vatri u igri

```
01.  void OnTriggerStay2D(Collider2D col)
02.  {
03.      if (col.CompareTag("Player"))
04.      {
05.          if (Input.GetKeyDown("e"))
06.          {
07.              SaveScore();
08.              SoundManager.PlaySound("level");
09.              SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
10.          }
11.      }
12.  }
```

Slika 4.34. Prikaz dijela kod prebacivanja na sljedeći nivo

Pritiskom tipke „e“ količina skupljenih bodova se spremi, te se indeks scene poveća za jedan, odnosno prebaci se na sljedeću nivo. A pritiskom tipke „e“ na zadnjem nivou, prebacuje se na početni izbornik jer je igra završena.

7.7. Zvukovi

U igri postoji dva različita pozadinska zvuka, jedan je zvuk početnog izbornika [38], a drugi je pozadinski zvuk nivoa [39]. Osim ta dva zvuka, igra sadrži sedam zvukova vezanih za skupljanje bodova, lik, protivnike i mijenjanje nivoa. Svi zvukovi su upravljani preko skripte upravitelja zvukova.

```
01.  public static void PlaySound(string clip)
02.  {
03.      switch (clip)
04.      {
05.          case "coin":
06.              audioSrc.PlayOneShot(coinPick);
07.              break;
08.          case "hurt":
09.              audioSrc.PlayOneShot(playerHurt);
10.              break;
11.          case "jump":
12.              audioSrc.PlayOneShot(playerJump);
13.              break;
14.          case "death":
15.              audioSrc.PlayOneShot(playerDeath);
16.              break;
17.          case "enemykilled":
18.              audioSrc.PlayOneShot(enemyDeath);
19.              break;
20.          case "meatbite":
21.              audioSrc.PlayOneShot(meatBite);
22.              break;
23.          case "level":
24.              audioSrc.PlayOneShot(level);
25.              break;
26.      }
27.  }
```

Slika 4.35. Prikaz PlaySound funkcije upravitelja zvukova

Korišteni zvukovi su prijelaz nivoa [40], žvakanje [41], uništavanje lika [42], oduzimanje životnih bodova [43], skakanje [44], skupljanje bodova [45], te uništavanje protivnika [46]

8. ZAKLJUČAK

Zadatak je završnog rada stvaranje dvodimenzionalne akcijske igre s horizontalnim pomicanjem u razvojnom okruženju Unity te opisivanje bitnih i primijenjenih dijelova, odnosno programa za razvoj višeplatformskih igara.

Također treba napomenuti da je Unity, zbog svoje licence za osobnu upotrebu, potpuno besplatno razvojno okruženje, te zbog toga svaki zainteresirani korisnik za razvoj igara može naučiti raditi u tom programu, ali uz uvjet da zna barem jedan od programskih jezika koje Unity zahtjeva. Programski se jezici također mogu besplatno naučiti korištenjem određenih internetskih stranica.

Za stvaranje, odnosno razvoj vlastitih dvodimenzionalnih igara, Unity nudi bezbroj mogućnosti koje mogu pomoći prilikom tog stvaranja.

Postoji problem kod grafičkih sličica i grafičkih listića. Mnogi ljudi koji koriste Unity programsko okruženje ne zna crtati, odnosno praviti sličice i listiće koji su potrebni za razvoj njihovih dvodimenzionalnih igara. Zbog toga traže tuđu grafiku i ne koriste svoju.

LITERATURA

[1] Unity (game engine)
https://en.wikipedia.org/wiki/Unity_(game_engine)

[2] Is Unity Engine written in Mono/C#
https://web.archive.org/web/20140709084653/http://answers.unity3d.com/questions/9675/is-unity-engine-written-in-monoc-or-c.html

[3] New Unity products and prices launching soon
https://blogs.unity3d.com/2016/05/31/new-products-and-prices/

[4] Unity multiplatform
https://unity3d.com/unity/features/multiplatform

[5] Virtualna stvarnost – tehnologija budućnosti
http://www.spsistemi.hr/column/virtualna-stvarnost-tehnologija-buducnosti/

[6] WebGL
https://en.wikipedia.org/wiki/WebGL

[7] Learning the Interface
https://docs.unity3d.com/510/Documentation/Manual/LearningtheInterface.html

[8] Project Browser
https://docs.unity3d.com/510/Documentation/Manual/ProjectView.html

[9] Hierarchy
https://docs.unity3d.com/510/Documentation/Manual/Hierarchy.html

[10] Inspector
https://docs.unity3d.com/510/Documentation/Manual/Inspector.html

[11] Scene View
https://docs.unity3d.com/510/Documentation/Manual/SceneView.html

[12] Microsoft Visual Studio
https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

[13] Introduction – About Tiled
http://doc.mapeditor.org/en/stable/manual/introduction/#about-tiled

[14] Tiled2Unity: Tilled Support for Unity
http://www.seanba.com/tiled2unity

[15] Piskel – free online sprite editor
https://www.piskelapp.com/

[16] Vektor
https://hr.wikipedia.org/wiki/Vektor

[17] Vector2
https://docs.unity3d.com/ScriptReference/Vector2.html

[18] Vector3
https://docs.unity3d.com/ScriptReference/Vector3.html

[19] Camera
https://docs.unity3d.com/ScriptReference/Camera.html

[20] Camera.orthographicSize
https://docs.unity3d.com/ScriptReference/Camera-orthographicSize.html

[21] Camera.fieldOfView
https://docs.unity3d.com/ScriptReference/Camera-fieldOfView.html

[22] An Introduction to Sprtesheet Animation
https://gamedevelopment.tutsplus.com/tutorials/an-introduction-to-spritesheet-animation--gamedev-13099

[23] Scripting
https://docs.unity3d.com/Manual/ScriptingSection.html

[24] Rigidbody 2D
https://docs.unity3d.com/Manual/class-Rigidbody2D.html

[25] Collider 2D
https://docs.unity3d.com/Manual/Collider2D.html

[26] Audio Overview
https://docs.unity3d.com/Manual/AudioOverview.html

[27] 971849-cool-prehistoric-background-1920x1080.jpg
http://getwallpapers.com/wallpaper/full/d/6/3/971849-cool-prehistoric-background-1920x1080.jpg

[28] Character animations (caveman)
https://opengameart.org/content/character-animations-caveman

[29] Dinosaur Trex.png
http://treeworld.wikia.com/wiki/File:Dinosaur_Trex.png

[30] Snakes
https://www.spriters-resource.com/pc_computer/spelunky/sheet/57053/

[31] Beauty and the Beast:Roar of the Beast
https://www.spriters-resource.com/genesis_32x_scd/batbroarofthebeast/sheet/28839/

[32] MonoBehaviour.Update()
https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html

[33] Hearths.png
https://drive.google.com/file/d/0B0osKu5IdBZgMS1pNEkyNE1fQVU/view

[34] yzfCODk.png
https://imgur.com/yzfCODk

[35] Groundtiles.png
https://opengameart.org/content/ground-tiles-1

[36] Set-cave_bright.png
https://opengameart.org/content/cave-tileset-2

[37] Campfire_16x16.png
https://opengameart.org/content/16x16-animated-campfire

[38] Game background Music loop short
https://freesound.org/people/yummie/sounds/410574/

[39] Jungle Shake Loop
https://freesound.org/people/rhodesmas/sounds/321388/

[40] teleport.wav
https://freesound.org/people/Sergenious/sounds/55853/

[41] Chewing, Carrot, A.wav
https://freesound.org/people/InspectorJ/sounds/412068/

[42] Grunt2 – Death Pain.wav
https://freesound.org/people/AlineAudio/sounds/416838/

[43] Hurt 2 – (Male).wav
https://freesound.org/people/Christopherderp/sounds/342230/

[44] MiniJump.wav
https://freesound.org/people/coby12388/sounds/222571/

[45] Retro Coin Collect
https://freesound.org/people/DrMinky/sounds/166184/

[46] 8-bit PEW I
https://freesound.org/people/ProjectsU012/sounds/333691/

SAŽETAK

Dvodimenzionalna akcijska igra s horizontalnim pomicanjem

Unity je razvojno okruženje koje se koristi za razvoj višeplatformskih igara. Kao takav program, korisnicima bi trebao biti jednostavniji za korištenje, te bi trebalo biti dostupno puno više literature i priručnika koje bi mogle pomoći prilikom korištenja tog programa. U završnom radu je opisan razvoj dvodimenzionalne akcijske igre s horizontalnim pomicanjem, te su opisane osnovne značajke, osnovni elementi i korišteni dijelovi Unity razvojnog okruženja. Prilikom razvoja igre, osim Unityja, koristio se Microsoft Visual Studio koji je služio za pisanje skripti u programskom jeziku C#, Piskel internetska stranica za izradu grafičkih listića i sličica, te su koristili Tiled i Tiled2Unity za izradu nivoa. Igra je prilikom razvoja imala je neke određene greške koje su popravljene, a nakon izvršavanja igre, pokrenuta je na Windows 10 i radi normalno na najvećoj rezoluciji.

Ključne riječi: Unity, razvojno okruženje, Unity uređivač višeplatformski, dvodimenzinalno, Microsoft Visual Studio, Piskel, Tiled, Tiled2Unity, sudarivač

2D action game with horizontal movement

Unity is a development environment which is used for developing multiplatform games. As such, the program’s usage should be simpler and there should be available much more literature and manuals which could help use the program. This paper describes the development of a 2D action game with horizontal movement. More precisely, the basic features, basic elements and used parts of development environment Unity are elaborated on. In addition to Unity, for the game development, Microsoft Visual studio was used to write scripts in C# programming language, web page Piskel for creating sprite sheets and sprites and Tiled and Tiled2Unity for creating levels. The game had some certain errors which have been fixed. Upon the game development, it was run on Windows 10 which showed that it works normally on the highest resolution.

Keywords: Unity, development environment, Unity Editor, multiplatform, 2D, Microsoft Visual Studio, Piskel, Tiled, Tiled2Unity, Collider

ŽIVOTOPIS

Marin Varnica rođen je 4. ožujka 1995. godine u Đakovu. Osnovnu školu „Ivana Gorana Kovačić“ Đakovo završava 2010. Nakon toga upisuje Srednju strukovnu školu „Braća Radić“ u Đakovu (danas Ekonomska škola „Braća Radić“) te ju završava 2014. Iste te godine upisuje Elektrotehnički fakultet u Osijeku (danas Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek), smjer računarstvo.