Izrada android "Clicker" igre korištenjem Unity platforme

Posavec, Martin

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVO

MARTIN POSAVEC

IZRADA ANDROID „CLICKER“ IGRE KORIŠTENJEM UNITY PLATFORME

ZAVRŠNI RAD

ČAKOVEC, 2021.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

STRUČNI STUDIJ RAČUNARSTVO

MARTIN POSAVEC

IZRADA ANDROID „CLICKER“ IGRE KORIŠTENJEM UNITY PLATFORME

MAKING OF ANDROID „CLICKER“ GAME USING PLATFORM UNITY

ZAVRŠNI RAD

Mentor:

Nenad Breslauer, v. pred.

ČAKOVEC, 2021.

***Zahvala*:**

Želio bih se zahvaliti svojoj obitelji koja mi je pružila mogućnost studiranja te svim prijateljima i kolegama koji su mi pružali podršku u studiranju.

Također velik hvala mom mentoru Nenadu Breslaueru na savjetima i pomoći tijekom izrade završnog rada.

SAŽETAK

Videoigre su jedan od najpopularnijih izvora zabave u današnje vrijeme, te su u velikom usponu kada je riječ o izradi i zaradi. Industrija videoigara vrijedi preko 150 milijardi američkih dolara što industriju videoigara čini većom i vrjednijom od filmske industrije. Vrijedi spomenuti da se industriji videoigara predviđa rast od 100% u sljedećih 5 godina što bi značilo da bi industrija videoigara u 2025. godini mogla vrijediti oko 300 milijardi američkih dolara.

Igrači videoigara su u porastu kao što je i cijela industrija, oko 59% igrača je muške populacije. Dok oko 41% čini ženska populacija, što je poprilično ujednačen omjer. Muška populacija je više naklonjena igranju videoigara na konzolama ili računalima, dok ženska populacija više igra videoigre preko svojih pametnih uređaja. Neke od najvećih platformi za igranje igara su Microsoft Xbox, Sony Playstation, Steam, Epic Games, Origin na računalima te Apple AppStore i Google Play trgovina na pametnim uređajima.

Videoigre polako ulaze i u sport te se nazivaju esport (engl. Esports). Esports igre su kompetitivne multiplayer igre koje omogućuju natjecanje protiv drugih igrača, a neki od najpopularnijih esports liga bilježe gledanost veću od NBA finala. Neke od najpopularnijih esports igara su League of Legends, Dota 2, Counter Strike, Fortnite i Rocket League.

Ovim radom prikazan je razvoj igre koja je namijenjena android platformi i android pametnim uređajima. Kroz rad prikazana je izrada modela, objekata te skripti potrebnih za igru. Ujedno je prikazan i cijeli proces izrade igre od same ideje do gotovog proizvoda spremnog za objavljivanje.

Na tržištu trenutno postoji nekoliko popularnih programa za izradu videoigara koji omogućuju da se igra kreira po želji i zamisli kreatora, a neki od njih su Unity, Unreal Engine i GameMaker: Studio. U izradi ovoga rada koristio se program Unity, a njegove mogućnosti i značajke predstavljene su u izradi ovog rada. Kroz ovaj rad pokazane su i mogućnosti programa Adobe Photoshop kao programa za izradu 2D modela te Microsoft Visual Studio 2019 kao programa za pisanje skripti i programskog koda.

Ključne riječi: *Unity 3D, Microsoft Visual Studio, Android game, Photoshop , Clicker*

Sadržaj

1. UVOD

Industrija videoigara postala je jedna od najvećih i vodećih industrija na svijetu. svjetsko tržište videoigara vrijedi više od 150 milijardi dolara i nastavlja rasti. Godinama je industrija videoigara u usponu i to omogućuje mnogo mjesta za zapošljavanje. Koliko je izrada videoigara napredovala govori nam činjenica da su u prošlosti bili potrebni cijeli timovi kako bi se izradila videoigra, dok je danas potrebna samo jedna osoba. Razvoj i izrada videoigre može biti vrlo kompliciran proces, sve to ovisi o vrsti i tipu videoigre koja se izrađuje. Kada se izrađuje kompliciranija videoigra onda se zapošljaju ljudi sa različitim stručnim znanjima, poput animatora, dizajnera, dizajnera nivoa (engl. Level), pisci, 3D modelara, programera i projekt menadžera. Kod izrade manje kompliciranе videoigre jedna osoba može nositi poslove i zadatke svih nabrojanih osoba i izraditi videoigru koja možda nije toliko kompliciranа, ali je i dalje dobra i kvalitetna.[6]

Igra koja je opisana ovim radom je Clicker žanra. To je pod žanr velikog žanra Ne zahtjevne igre(engl. Casual) koji ima vrlo jednostavnu mehaniku, a cilj joj je nakratko zabaviti igrača. Clicker žanr je žanr u kojem se sva mehanika i kretanje igrača svode na jedan obični klik tj. pritisak tipke ili zaslona.

2. CILJ RADA

Cilj rada je prikaz procesa izrade android videoigre koja je spremna za igranje i objavljivanje na Google Play trgovini, te komponiranje znanja iz područja računarstva u cjelinu, odnosno u videoigru. Također cilj je prikaz i predstavljanje računarstva mlađim generacijama kao zabavnog i zanimljivog područja koje nas može naučiti puno toga te nam pomoći u postizanju i materijalizaciji naizgled nezamislivih scenarija i priča.

Tijekom izrade rada proširit će se znanje i vještine potrebne za proces izrade android igre.

3. RAZVOJNA OKOLINA

Skup programa koji su potrebni za izradu videoigre i svih komponenti koje se koriste za izradu igre čine razvoju okolinu. Unity 3D je program koji se koristi kako bi se u jednu cjelinu tj. videoigru spojile sve komponente koje se koriste. Adobe Photoshop CS6 je program koji se koristi za izradu 2D modela koji se kasnije koriste u Unity-u. Microsoft Visual Studio 2019 program je koji kreira skripte u C# programskom jeziku.

3.1 Adobe Photoshop CS6

Adobe Photoshop je softver koji se koristi za uređivanje slika, grafički dizajn i digitalnu umjetnost. Koristi slojevitost(engl. *layering*) kako bi se omogućila dubina i fleksibilnost u izradi dizajna. Razvila ga je tvrtka Adobe System, a moguće ga je koristiti na Windows i MacOs računalima, kao i na IOS i Android pametnim telefonima ili tabletima.[3]

3.2 Unity 3D

Unity 3D vodeći je program za izradu videoigara u industriji videoigara. Pomoću Unity-a moguća je izrada 2D i 3D videoigre. Prva verzija Unity-a stavljena je na tržište 6. lipnja 2005 godine. Od tada do danas izašlo je mnogo verzija Unity-a kroz koje je moguće vidjeti veliku razliku u grafici, zvuku te mnoštvu drugih stvari koje izradu videoigara čine jednostavnijom i lakšom. Neke vrlo popularne igre poput *Hearthstonea, Rusta, Assasin's Creed*: *Identityja* i *Temple Runa* su samo neke od igara kreirane u Unity-u. Unity nudi veliki izbor platformi za koje se mogu razvijati videoigre poput Windowsa, Linuxa, MacOSa, iOSa, Androida, Oculus Rifta i mnogih drugih. Programski jezik kojim se kreiraju skripte potrebne za kreiranje mehanika igre je C#. U prošlosti se koristio JavaScript jezik, ali pošto su mogućnosti C# jezika veće, JavaScript se više ne koristi. Unity nam omogućuje pristup svojoj vlastitoj trgovini (engl. *Asset store*) u kojoj se mogu pronaći animacije, scene, likovi, objekti, skripte koje je moguće kupiti ili, ukoliko su besplatni samo preuzeti. Smisao postojanja takve platforme je da se ne troši vrijeme na izradu stvari koje su drugi ljudi napravili i koje nam mogu biti korisne.[1]

3.3 Microsoft Visual Studio 2019

Program Microsoft Visual Studio razvila je od tvrtka Microsoft. Korištenjem ovog programa moguća je izrada skripti i pisanje koda u raznim programskim jezicima poput C++, C#, C , Python, JavaScript te raznih drugih. Za izradu računalne igre Unity koristi skripte pisane C# programskim jezikom. Microsoft Visual Studio omogućuje automatsko pronalaženje pogrešaka pomoću alata koji pronalazi pogreške (engl. *Debugger*) te posjeduje mogućnost analize programskog koda.

4. Dizajn videoigre

Dizajn videoigre su elementi koji stvaraju koncept igre, mehanike igre i njezina pravila. Dizajn videoigre određuje temu igre. Prilikom dizajniranja videoigre specificiraju se njezine mehanike, odlučuju se pravila, postavljaju se sljedovi igre, definiraju se radnje igrača, kao i početak i kraj igre te se odlučuju uvjeti pobjede.[4]

4.1. Ghost Portals

Ghost Portals je ime igre koja se spominje u ovom radu. Iz samog imena možemo iščitati o kakvoj se videoigri radi. Naime, *Ghost Portals* je u suštini igra preživljavanja, u kojoj svakim prolaskom kroz portal dobivamo bodove. Cilj igre je proći kroz što više portala dok istovremeno izbjegavamo protivnike. Smisao igre je što duže preživljavanje i što duže izbjegavanje neprijatelja uz ostvarivanje što većeg broja bodova.

4.1.1. Priča igre

Mi uzimamo ulogu igrača i kontroliramo duha koji pokušava izbjeći neprijateljske duhove te u isto vrijeme mora proći kroz portal jer će se u suprotnome zaletjeti u granicu ekrana i umrijeti.

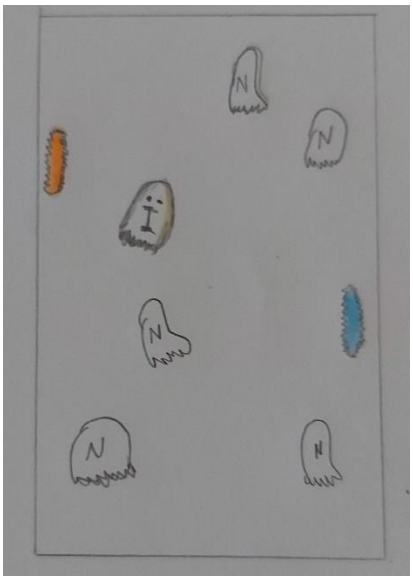
4.1.2. Glavni objekti u igri

1. Igrač
2. Neprijatelji
3. Portali

4.1.3. Pravila igre

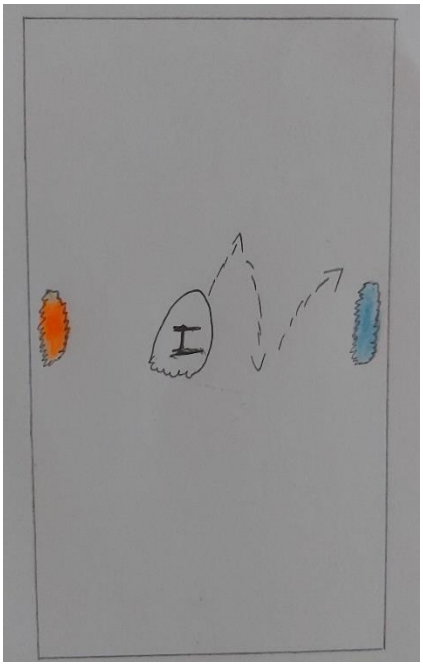
1. Uvjet pobjede: u ovoj igri zapravo nema pobjede, nego je cilj igre sakupiti što više bodova
2. Igrač može umrijeti samo ako udari u neprijatelja ili granice ekrana
3. Igrač skuplja bodove prolaskom kroz plavi portal
4. Igrač ima određen broj života
5. Broj života se može povećati ako se pokupi srce koje se pojavljuje na ekranu nakon nekog vremena

4.1.4. Prikaz vizualnog dizajna videoigre



Slika 1. Dizajn igre, igrač, neprijatelji, portali

Izvor: Autor



Slika 2. Dozvoljene kretnje igrača

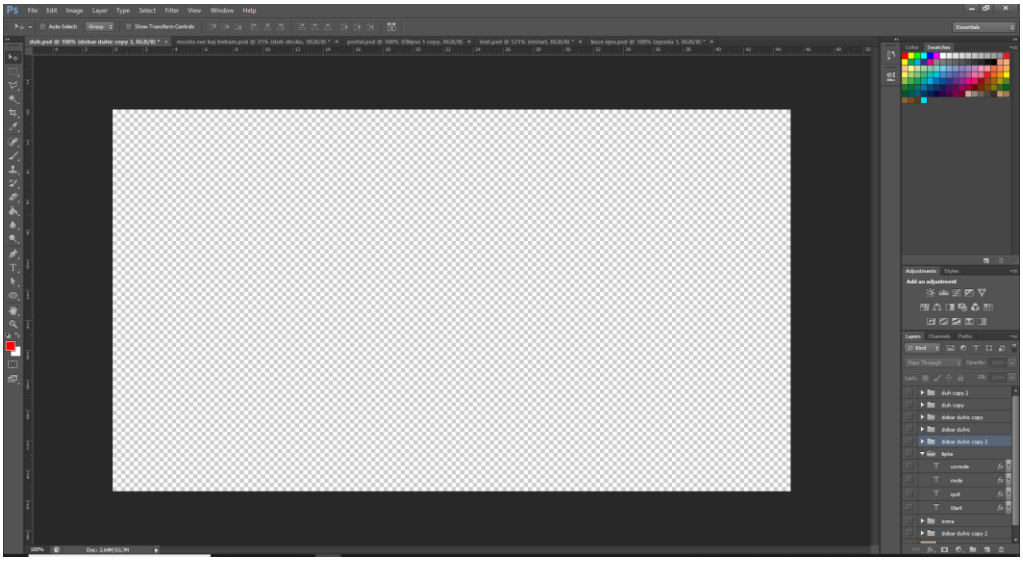
Izvor: Autor

5. Proces izrade igre

Prikazuje se proces izrade *Ghost Portals* igre kao i izrada modela te programski kod i suradnja između razvojne okoline.

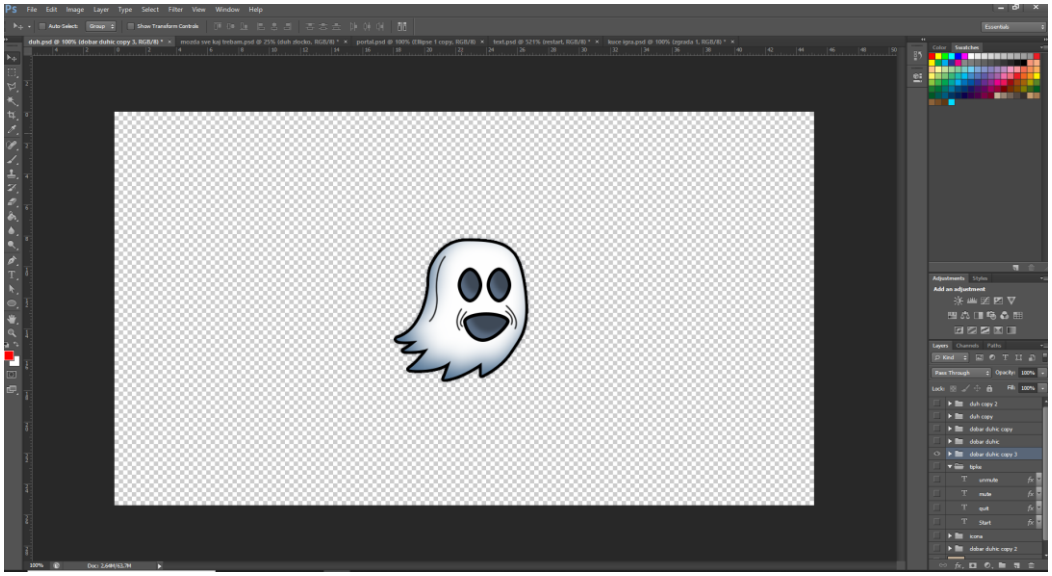
5.1 Izrada 2D modela u alatu Adobe Photoshop

U programu Adobe Photoshop CS6 kreirani su modeli duha, zločestog duha, portala, srca, pozadinske slike, tipki za *play*, *mute*, *unmute*, *exit*, *menu* i *restart*. Na sljedećim slikama prikazana su Photoshop sučelja te modeli u Photoshop-u prije nego što su umetnuti u program Unity.



Slika 3. Početni prizor Photoshop-ovog sučelja

Izvor: Autor



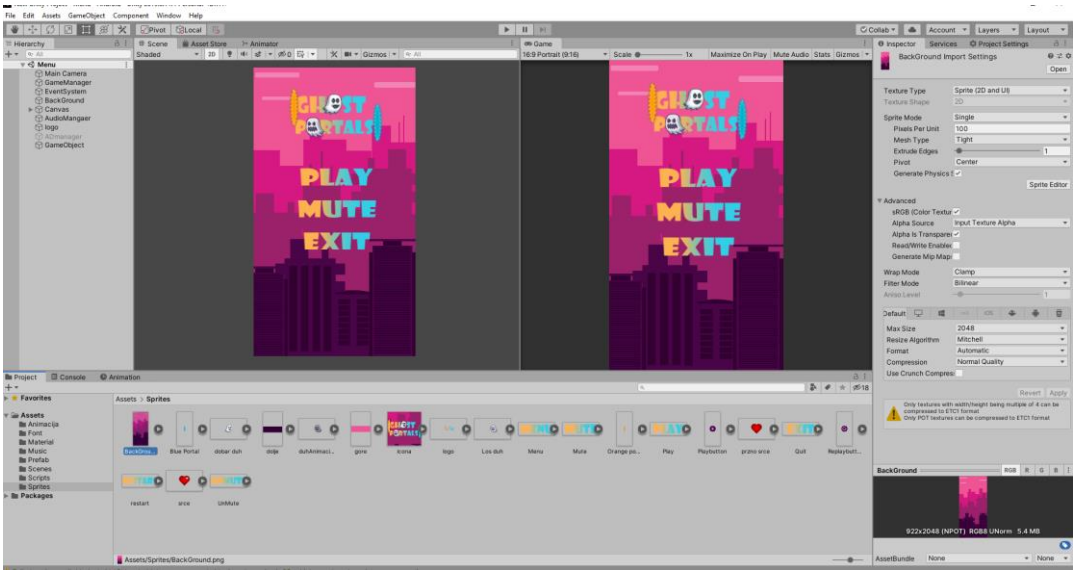
Slika 4. Izrada duha u Photoshopu

Izvor: Autor

Istim načinom kreirani su i ostali objekti koji će se koristiti u igri. U Photoshopu su uz pomoć olovka alata (engl. *Pen tool*) kreirani svi objekti. Alat olovka funkcionira tako što se njome, uz pomoć miša, crta na prazno platno, poput crtanja s olovkom na praznom papiru. Nakon što su svi modeli napravljeni, isti se izvode (engl. *export*) u PSD ili PNG formatu i ubacuju se u Unity.

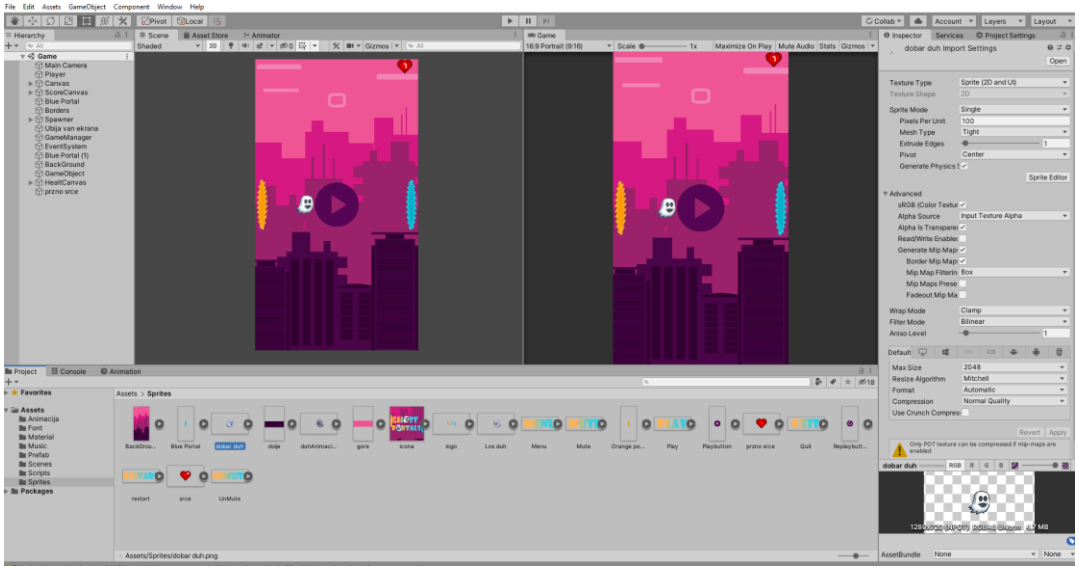
5.2 Primjena Unity 3D softvera i programski kod

Unity 3D program moguće je pokrenuti na Windows, MacOS i Linux operativnom sustavu, ali videoigre koje se kreiraju u njemu kompatibilne su s različitim platformama poput Androida, iOSa, PlayStationa, Xboxa, Facebooka te brojnih drugih. Kod kreiranja videoigara za različite platforme važno je obratiti pozornost na različite detalje u izradi igre jer nije isto igrati igru na Android pametnom uređaju, na iOS uređaju ili pak na Playstationu, o svemu tome se mora voditi računa prilikom izrade videoigre. [1]



Slika 5. Primjer Unity sučelja

Izvor: Autor



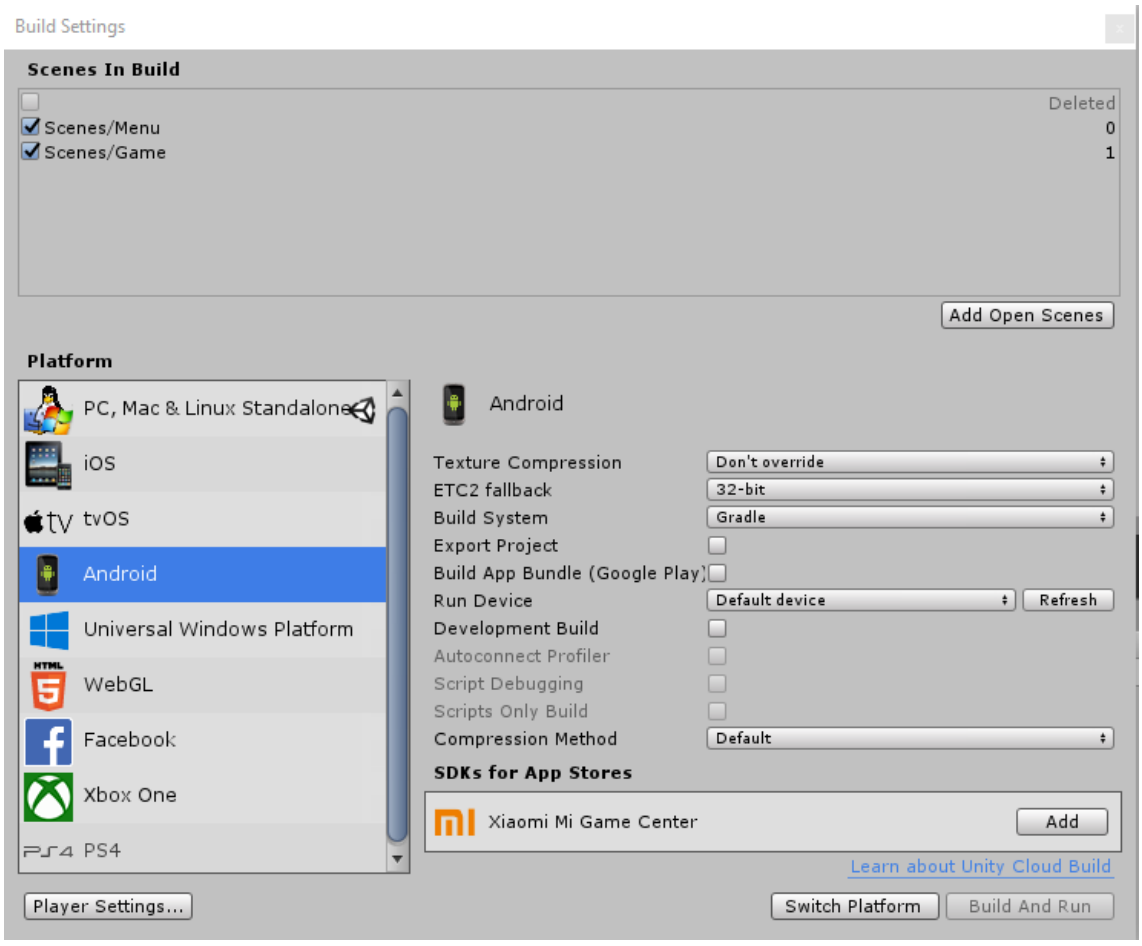
Slika 6. Primjer Unity sučelja

Izvor: Autor

Na slikama je prikazano Unity sučelje i dvije različite scene. Prva scena je izbornik (engl. *Menu*), odnosno scena koja nam se otvori kada se pokrene videoigra, a druga je scena koja nam se prikaže kada se stisne na tipku play.

5.2.1 Stvaranje projekta

Prilikom otvaranja Unity programa pojavljuje prozor koji nam nudi mogućnost da otvorimo postojeći projekat ili stvorimo novi projekt. Ukoliko odaberemo izradu novog projekta. tada se moramo odlučiti hoće li novi projekat biti 3D ili 2D tipa. Za kreiranje ove videoigre odabran je projekt 2D tipa. Pošto je videoigra namijenjena android korisnicima mora se otići u postavke izrade (engl. *build settings*) i prebaciti ih na android platformu.[1]

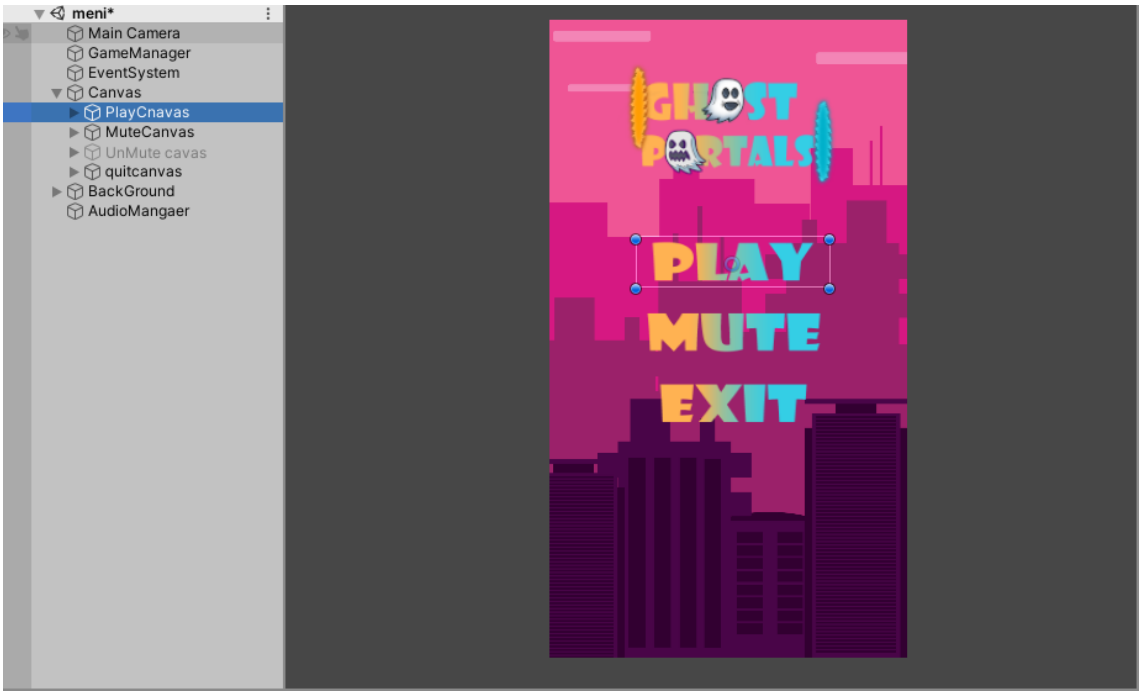


Slika 7. Primjer Unity build settings

Izvor: Autor

5.2.2 Stvaranje meni scene

Kod otvaranja programa Unity on nam sam kreira prvu scenu koju možemo izbrisati, ali i iskoristiti. Scena izbornika je kreirana dodavanjem pozadinske slike (engl. *Background*), platna (engl. *Canvas*) te tipki za pokretanje igre, gašenje ili paljenje zvuka i izlazak iz igre. Platno je područje koje služi za upravljanjem izbornika (engl. *Menu*). Svi elementi izbornika kojima se želi upravljati moraju biti djeca objekta „Canvas“. Interakciju između korisnika i platna omogućuje sustav događaja (engl. *Event System*).[4]



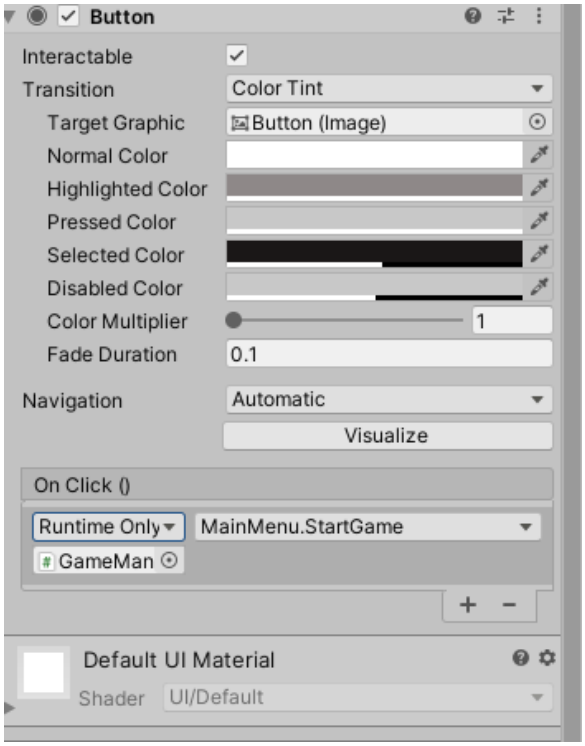
Slika 8. Meni scena

Izvor: Autor

Na slici je prikazana izbornik scena na kojoj vidimo pozadinsku sliku i platno koje sadrži pod platna koji predstavljaju tipke *play*, *mute*, *unmute*, *exit*.

5.2.2.1. Dodavanje tipki na scenu

Svaka tipka ima svoje platno koje je dijete glavnog platna. Svaka tipka izvršava određenu metodu pritiskom na tipku. Metode se pozivaju iz mainMenu skripte koja se nalazi na GameManager objektu.



Slika 9. Primjer tipke Play

Izvor: Autor

Na slici je prikazano kako se funkcijom On Click sa objekta GameManager iz skripte MainMenu poziva metoda StartGame.

5.2.2.2. Programski kod izbornika

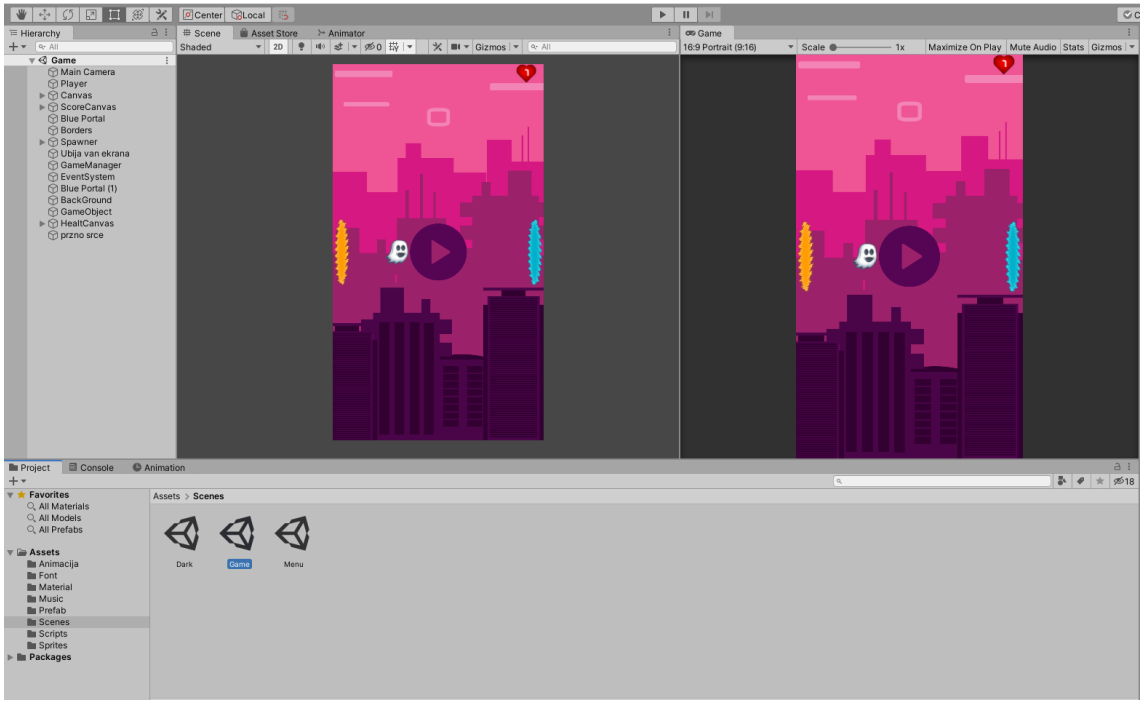
Programski kod izbornika sastoji se od metoda koje se pozivaju pritiskom na tipke *start*, *mute*, *unmute* i *exit*. Metode za *mute* i *unmute* također skrivaju i otkrivaju platna za tipke *mute* i *unmute* prilikom pritiska na njih.[5]

```
    public GameObject muteCanvas;
    public GameObject UnMuteCanvas;

public void StartGame()
    {
        SceneManager.LoadScene("Game");
    }

public void Quit()
    {
        Application.Quit();
    }

public void Mute()
    {
        muteCanvas.SetActive(false);
        UnMuteCanvas.SetActive(true);
```

```
    }
    public void UnMute()
    {
        muteCanvas.SetActive(true);
        UnMuteCanvas.SetActive(false);
    }
```

5.2.3. Stvaranje Game scene

Stvaranjem game scene stvaramo videoigru. Scena je mjesto na kojem nalaze igrač i svi objekti koji su potrebni za igru, odnosno tu se događa igranje (engl. *Gameplay*) igre. Za ovu scenu korištena je ista pozadina kao u izbornik sceni. Game sceni dodajemo igrača, neprijatelje, portale, rezultat te živote.



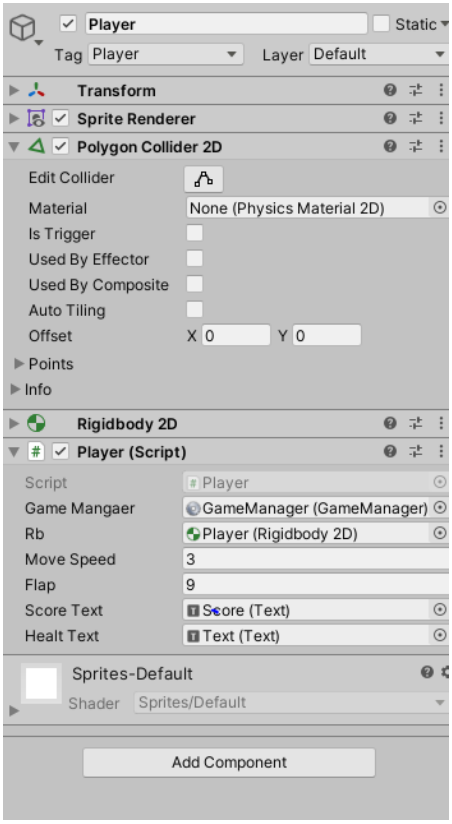
Slika 10. Gotova game scena

Izvor: Autor

5.2.4. Kreiranje igrača

Prvi korak kreiranja igrača je dodavanje 2D modela koji je napravljen u Photoshop-u. Na njega su dodane komponente (engl. *Component*) rigidbody 2D, sudarač (engl. *Collider*) te skripta koja pokreće igrača i omogućuje interakciju s okolinom. Skripta

sadrži kod za brzinu kretnje igrača, visinu skoka te za interakciju s drugim objektima kao sudaranje s neprijateljem ili prolazak kroz portal.



Slika 11. Komponente dodane igraču

Izvor: Autor

5.2.4.1. Programski kod koji omogućuje kretanje igrača

Programski kod koji omogućuje kretanje igrača po horizontalnoj osi i skok igrača na vertikalnoj osi uzima varijablu moveSpeed i njenom vrijednosti se kreće po horizontalnoj osi, potom ako pritisnemo na ekran uzima varijablu flap i njezinom vrijednosti skoči po vertikalnoj osi.[5]

```
void Update()
{
    rb.velocity = new Vector2(moveSpeed, rb.velocity.y);

    if (Input.GetMouseButtonDown(0))
    {
        rb.velocity = new Vector2(rb.velocity.x, flap);
    }

}
```

5.2.4.2. Programski kod koji omogućuje teleportaciju

Programski kod koji omogućuje teleportaciju bilježi lokaciju narančastog portala i prilikom sudara tj. prolaska igrača kroz plavi portal šalje igrača na njegovu novu poziciju koja je upravo lokacija narančastog portala.

```
public void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.tag == "Player")
        {
            StartCoroutine(Teleport());
        }
    }


    IEnumerator Teleport()
    {
        yield return new WaitForSeconds(0.01f);
        Player.transform.position = new
Vector2(Portal.transform.position.x+0.5f, Portal.transform.position.y);
    }
```

5.2.4.3. Programski kod koji dodaje život

Programski kod koji nam kada pokupimo srce, odnosno kada se sudarimo se s njim daje jedan dodatan život.

```
  int playerhealt = 1;

if (col.gameObject.tag == "Shild")
        {

            playerhealt++;
            healtText.text = playerhealt.ToString();


        }
```

5.2.4.5. Programski kod koji oduzima život

Programski kod koji nam kada se sudarimo s neprijateljem oduzima jedan život ili sve živote ukoliko se sudarimo s rubom ekrana.

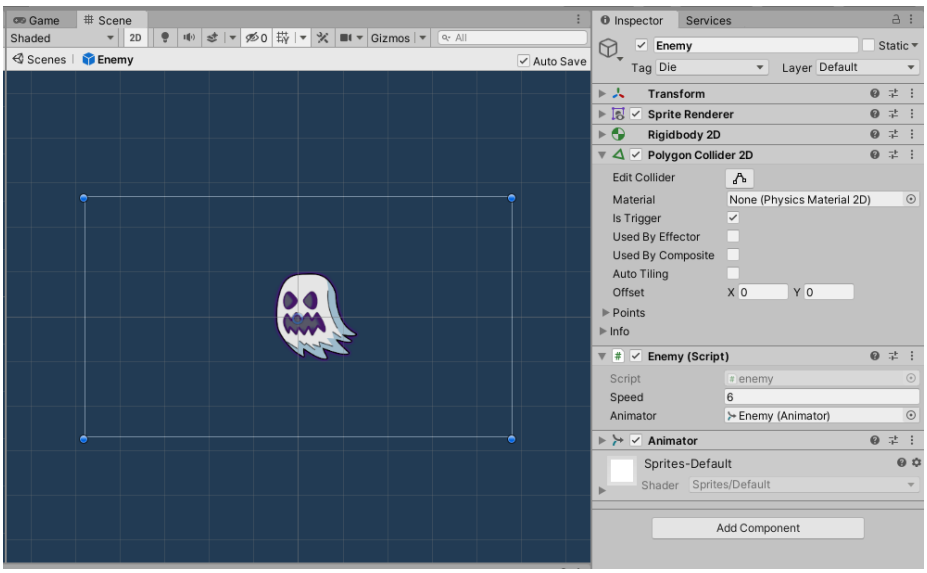
```
int playerhealt = 1;
if (col.gameObject.tag == "Die")
            {
```

```
                playerhealt--;
                healtText.text = playerhealt.ToString();
                if (playerhealt == 0)
                {
                    gameMangaer.GameOver();
                }

            }

if (col.gameObject.tag == "Border")
        {

            playerhealt=playerhealt-101;

            if (playerhealt <= 0)
            {
                gameMangaer.GameOver();
            }

        }
```

5.2.5. Kreiranje neprijatelja

Prvi korak kreiranja neprijatelja je dodavanje modela neprijatelja napravljenog u Photoshopu, a potom i dodavanje komponenti rigidbody 2D, sudarač (engl. *Collider*), te skripte na njega. Nakon toga on je pretvoren u prefab model.

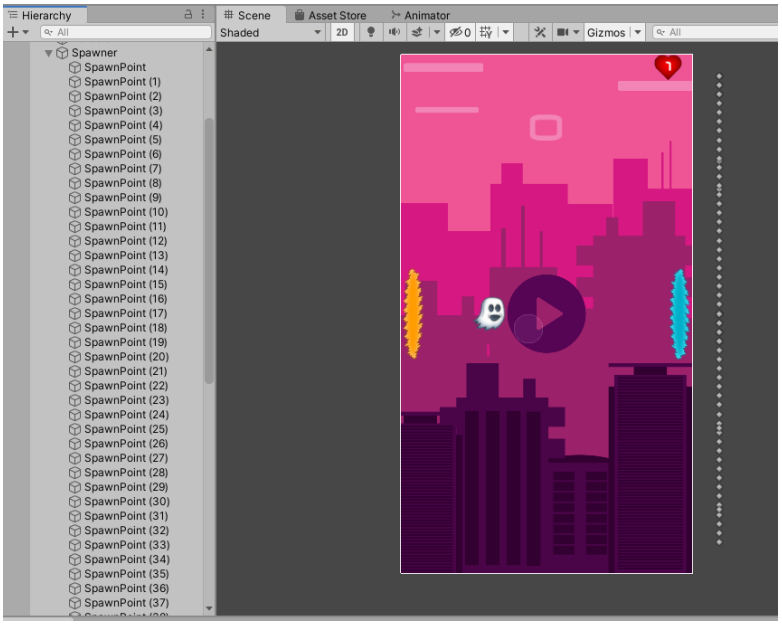
Prefab je igraći objekt (engl. *Gameobject)* koji na sebi ima spremljene sve dodane komponente te se poziva na scenu nakon njezina početka.



Slika 12. Enemy prefab

Izvor: Autor

Dodavanje neprijatelja napravljeno je pomoću praznih igraćih objekata (engl *Gameobject*). Spawner je glavni prazni igraći objekt i sadrži šezdeset četvero djece, također praznih igraćih objekata, nazvane SpawnPoint, koji označavaju lokaciju stvaranja neprijatelja te skripte koja omogućuje stvaranje prefab-a neprijatelja na scenu. Korištene su šezdeset četiri lokacije kako bi se pokrio cijeli ekran te kako bi se dobilo više raznolikosti u putanji neprijatelja.



Slika 13. Prikaz Spawner i SpawnPoint

Izvor: Autor

5.2.5.1. Programski kod neprijatelja

Programski kod koji omogućuje kretanje igrača uzima varijablu speed te se vrijednosti te varijable vektorski kreće u lijevo. Varijabla speed je zadana u Unityu.[5]

```
public float speed;

void Update()
{

    transform.Translate(Vector2.left * speed * Time.deltaTime);

}
```

Programski kod sastoji se i od okidača (engl. *Trigger*) koji kada dođe do sudara s drugim igraćim objektom jednostavno uništi neprijatelja ili u slučaju sudara s igračem pokrene animaciju padanja neprijatelja.

```
void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.tag == "Out")
        {

            Destroy(gameObject);

        }
        if (other.gameObject.tag == "BluePortal")
        {

            Destroy(gameObject);

        }
        if (other.gameObject.tag == "OrangePortal")
        {

            Destroy(gameObject);

        }
        if (other.gameObject.tag == "Player")
        {

            this.GetComponent<Rigidbody2D>().gravityScale = 3f;
            animator.SetBool("Alive", false);

        }

    }
```

5.2.5.2. Programski kod spawnera

Programski kod spawnera sastoji se od varijable spawnSpots koja predstavlja lokaciju SpawnPointa te varijabli vremena koje sadrže vrijednosti početnog vremena stvaranja neprijatelja, najmanjeg mogućeg vremena stvaranja neprijatelja, koliko se početno vrijeme smanjuje i igraćeg objekta koji predstavlja neprijatelja.

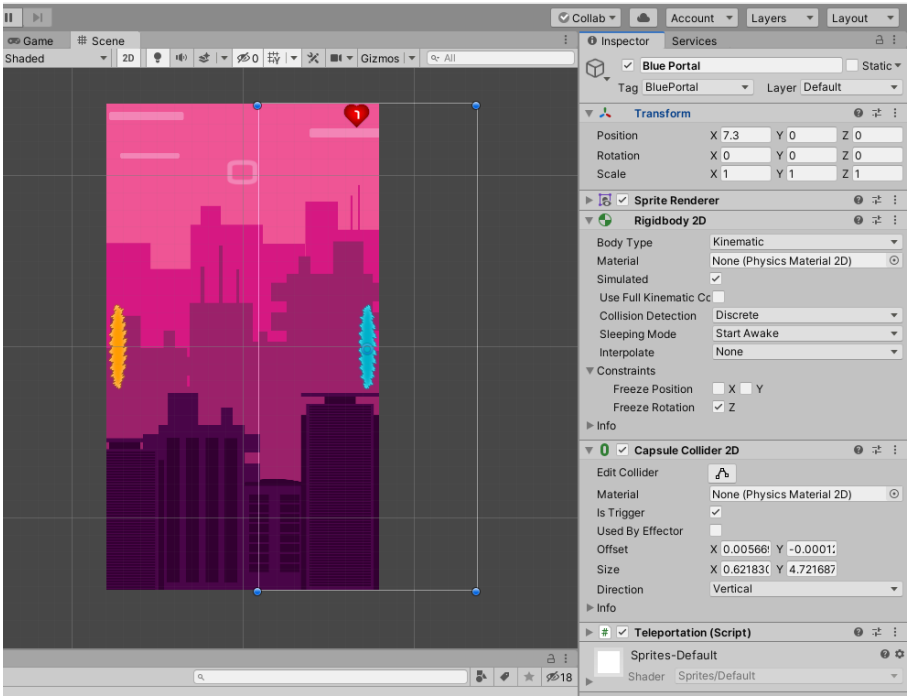
```
public GameObject enemy;
    public Transform[] spawnSpots;
    private float timeSpawn;
    public float StartTime;
    public float descTime;
    public float minTIme;
```

```
    void Start()
    {

    }


    void Update()
    {

        if (timeSpawn <= 0)
        {
            int randPos = Random.Range(0, spawnSpots.Length);
            Instantiate(enemy, spawnSpots[randPos].position,
Quaternion.identity);
            timeSpawn = StartTime;

            if (StartTime > minTIme)
            {

                StartTime -= descTime;
            }

        }
        else
        {
            timeSpawn -= Time.deltaTime;
        }
    }
```

Skripta nasumice odabire poziciju u kojoj će stvoriti neprijatelja te to radi u vremenskim okvirima koji se smanjuju dok ne dođe do najmanjeg vremena između stvaranja.

5.2.6. Kreiranje portala

Prvi korak kreiranja portala je dodavanje modela kreiranih u Photoshop-u, te dodavanje komponenti rigidbody 2D, sudarača (engl. *Collider*) i kreiranje programskog koda koji će mijenjati lokaciju portala prilikom prolaska igrača.



Slika 14. Portali

Izvor: Autor

5.2.6.1. Programski kod portala

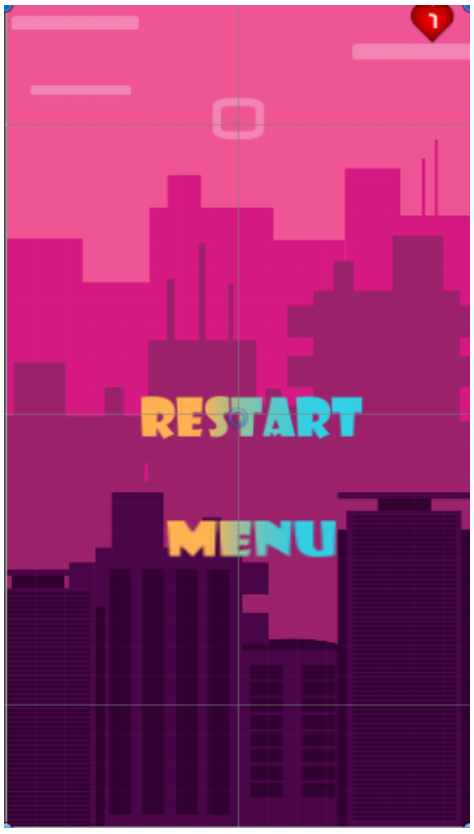
Programski kod portal sadrži lokaciju portala i svaki put kada igrač prođe kroz portal uzima nasumičnu vrijednost na vertikalnoj osi i pomiče portal gore ili dolje po njoj, dok je na horizontalnoj osi vrijednost uvijek ista da ne dođe do pomaka lijevo i desno.

```
    float x;
    float y;
    float x1;
    Vector2 pos;
    Vector2 pos1;


    void Start()
    {
        x = 7.3f;
        x1 = -7.3f;
    }

    void Update()
    {
        y = Random.Range(10.5f, -10.5f);
        pos = new Vector2(x, y);
        pos1 = new Vector2(x1, y);
    }

if (col.gameObject.tag == "BluePortal")
            {
                col.gameObject.transform.position = pos;
```

```
            }

if (col.gameObject.tag == "OrangePortal")
        {
            col.gameObject.transform.position = pos1;

        }
```

5.2.7. Početno platno

Početno platno (engl. *Canvas*) sadrži tipku za početak koju pritišćemo kada želimo započeti igru, tipku za ponovo pokretanje (engl. *Restart*) kada izgubimo sve živote te tipku izbornika (engl. *Menu*) za povratak u početni izbornik. Platnom upravlja igraći objekt (engl. *Gameobject*) koji se zove Game Manager i koji sadrži GameManager skriptu koja sadrži kod za upravljanje platnom.



Slika 15- Tipke restart i menu

Izvor: Autor

5.2.7.1. Game Manager kod

GameManager skripta se sastoji od metoda koje se pozivaju prilikom smrti igrača ili prilikom pritiska tipke.

```
public void Play()
    {
        startCanvas.SetActive(false);
        Time.timeScale = 1;
    }

public void Replay()
    {
        SceneManager.LoadScene("Game");
    }
public void GameOver()
    {
        gameOverCanvas.SetActive(true);
        Time.timeScale = 0;
    }
public void MenuMeni()
    {
        SceneManager.LoadScene("Menu");

    }
```

5.2.8. Bodovanje

Sustav bodovanja je implementiran tako da, ako igrač prođe kroz plavi portal, dobije jedan bod. Sakupljeni bodovi prikazani su na platnu nazvanom scoreCanvas. Programski kod zadužen za bodovanje uzima tekst s platna i povećava ga za jedan svaki put kada igrač prođe kroz plavi portal.

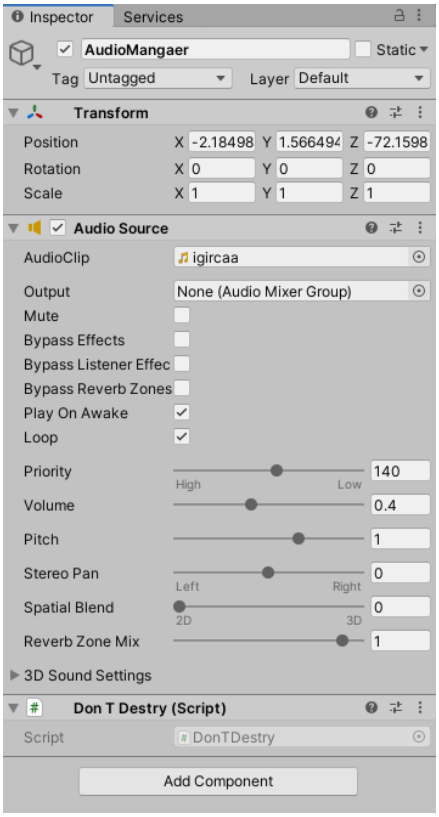
```
public Text scoreText;
int score = 0;
if (col.gameObject.tag == "BluePortal")
            {

                col.gameObject.transform.position = pos;
                score++;
                scoreText.text = score.ToString();

            }
```

5.2.9. Zvuk

Zvuk u ovoj videoigri ostvaren je pomoću komponente AudioSource koja se nalazi na praznom igraćem objektu nazvanom Audio Manager. Zvuk je izveden tako da počinje na

prvoj sceni i svira bez prekida na sljedećoj. I to je ostvareno pomoću skripte koja, kaže Audio Manageru da se ne ugasi prilikom prelaska između scena.

```
void Awake()
    {
        DontDestroyOnLoad(transform.gameObject);
    }
```



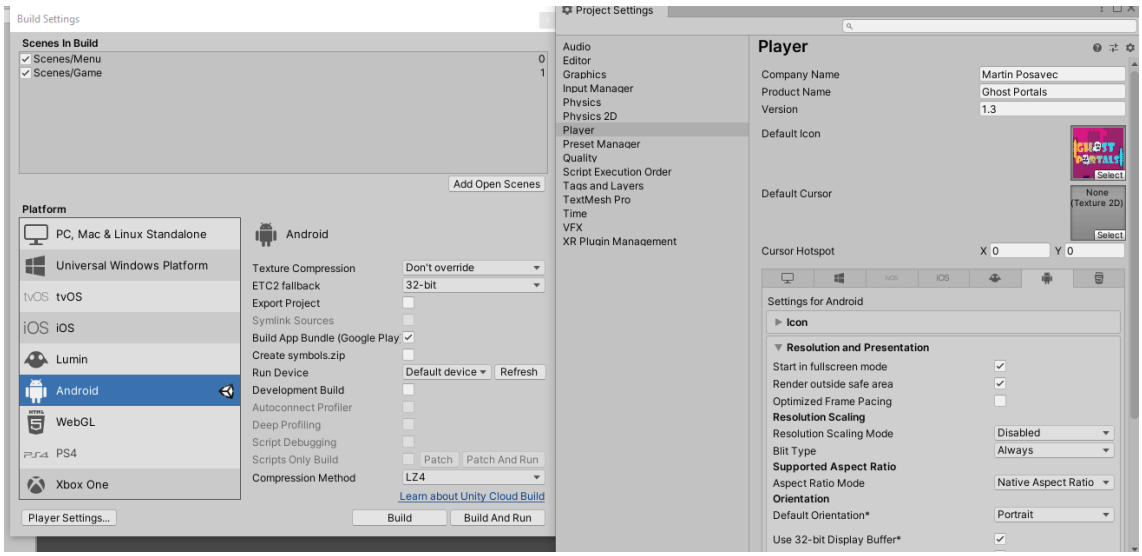
Slika 16. Audio Manager

Izvor: Autor

5.2.10. Izrada izvršne datoteke

Kada je završena izrada videoigre potrebno je izgraditi (engl. *Build*) datoteku koja je kompatibilna s android uređajima. Unity nam nudi mogućnost izrade videoigara za mnoštvo platform, a pošto je ova igra predviđena za android uređaje mora se odabrati android platforma. Prilikom izrade prvo je potrebno dodati scene koje su izrađene onim redoslijedom koji želimo da se pojavljuju na ekranu. U slučaju ovog projekta prvo se dodaje izbornik (engl. *Menu*) scena, a zatim game scena koja se pokreće pomoću

programskog koda. Poslije pravilnog dodavanja scena trebamo urediti postavke za rezoluciju, ikone, imena i verzije igre te odlučiti se za verzije android sustava koji će moći pokretati igru.[1]



Slika 17. Postavke igre

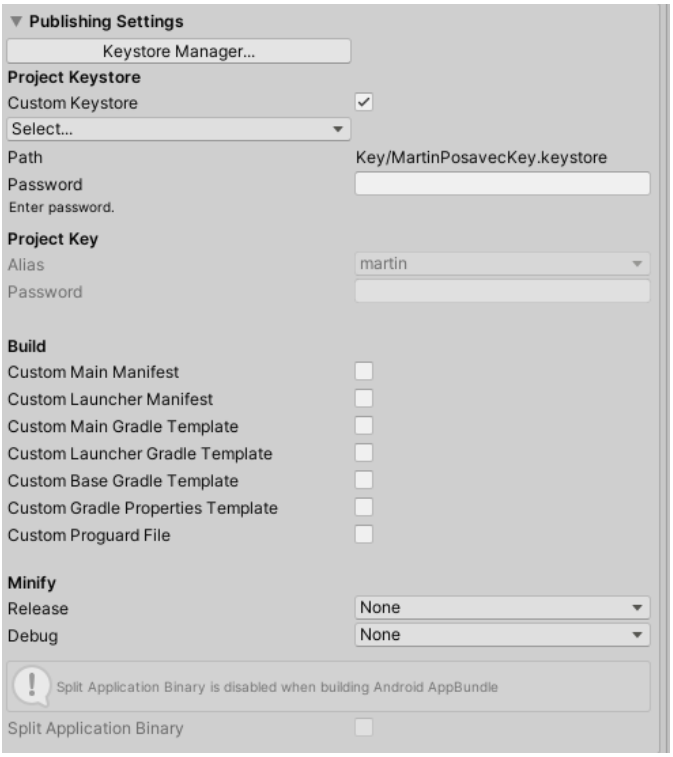
Izvor: Autor

6. Objavljivanje igre u trgovini Google Play

Trgovina Google Play je mjesto koje dolazi instalirano na svim android pametnim uređajima i uz pomoć kojeg preuzimamo ostale aplikacije na naš uređaj. Sadržaj trgovine više nisu samo aplikacije već je spajanjem Google glazbe, Google filmova i Google e-knjige postala mjesto na kojem možemo pronaći sve za naš uređaj. Trgovina Google Play broji 2.5 milijarde korisnika.

6.1. Postavke za objavljivanje u Unity-u

U Unity-u potrebno je podesiti postavke objavljivanja (engl. *Publishing settings*). Najvažnije je napraviti pohranu ključa (engl. *Keystore*) i to izradom datoteke koja u sebi ima spremljen ključ aplikacije. Ključ aplikacije služi za naknadno objavljivanje unapređenja (engl. *Update*) ili dodavanja novog sadržaja, bez ključa aplikaciju nije moguće objaviti ili staviti novu verziju aplikacije u trgovinu. Nakon završetka podešavanja postavki potrebno je izraditi izvršnu datoteku tipa (.apk).

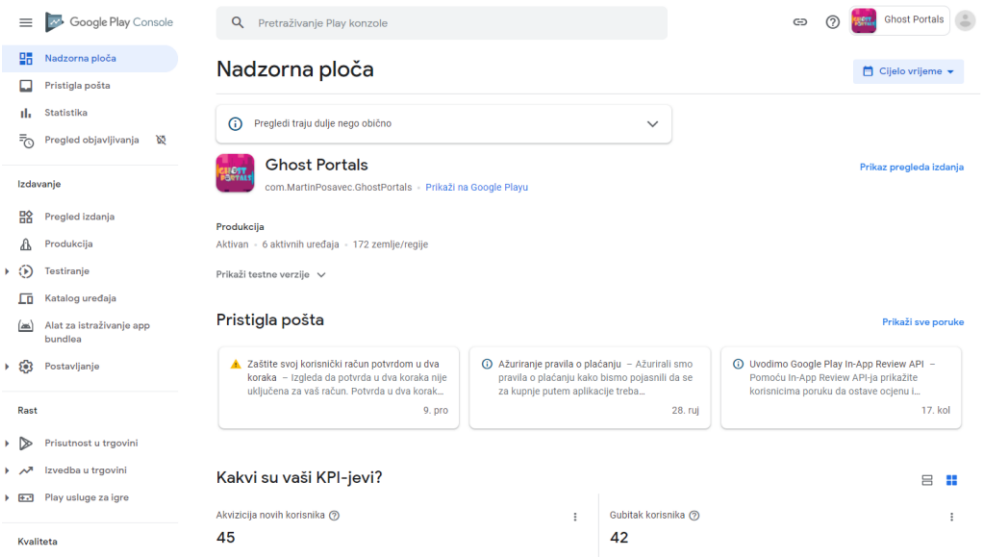


Slika 18. Publishing settings

Izvor: Autor

6.2. Objava igre u Google Play trgovini

Ako želimo objaviti igru na trgovini Google Play najprije je potrebno kupiti i izraditi razvojni račun (engl. *Developer account*) koji nam omogućuje objavu aplikacija i videoigara u trgovini Google Play. Cijena razvojnog računa iznosi 160 hrvatskih kuna. Nakon kupnje računa potrebno je prijaviti se na Google Play Console web stranici koja omogućuje cjelokupni proces objavljivanja igre. Potrebno je pritisnuti na tipku *Izradite aplikaciju* nakon toga otvara nam se prozor u koji unosimo naziv aplikacije biramo jezik te određujemo radi li se o videoigri ili aplikaciji te hoće li aplikacija biti besplatna ili će se naplaćivati. Nakon toga nam se otvara nadzorna ploča u kojoj možemo kontrolirati sve vezano uz našu aplikaciju i pratiti njezin rast ili pad. S nadzorne ploče potrebno je otići na karticu produkciju te izraditi novo izdanje. Nakon pritiska tipke *Izradi novo izdanje* otvara nam se prozor u koji je potrebno dodati izvršnu datoteku tipa (.apk), odrediti verziju izdanja te napraviti opis izdanja, a nakon toga spremiti izdanje. Nakon spremanja Google Play će obaviti kontrolu vaše aplikacije da se utvrdi radi li se o zlonamjernom programu ili ne, Kontrola traje nekoliko dana. Da bi aplikacija bila u potpunosti spremna za objavljivanje potrebno je sa nadzorne ploče otići na karticu rast na kojoj pod karticom prisutnost u trgovini i njenoj pod kartici glavni unos u trgovini je potrebno unijeti naziv aplikacije, kratki opis, puni opis te grafičke elemente koji su ikona, istaknuta grafika te barem 2 slike zaslona igre i ako želimo moguće je dodati link na videozapis koji prikazuje igru.

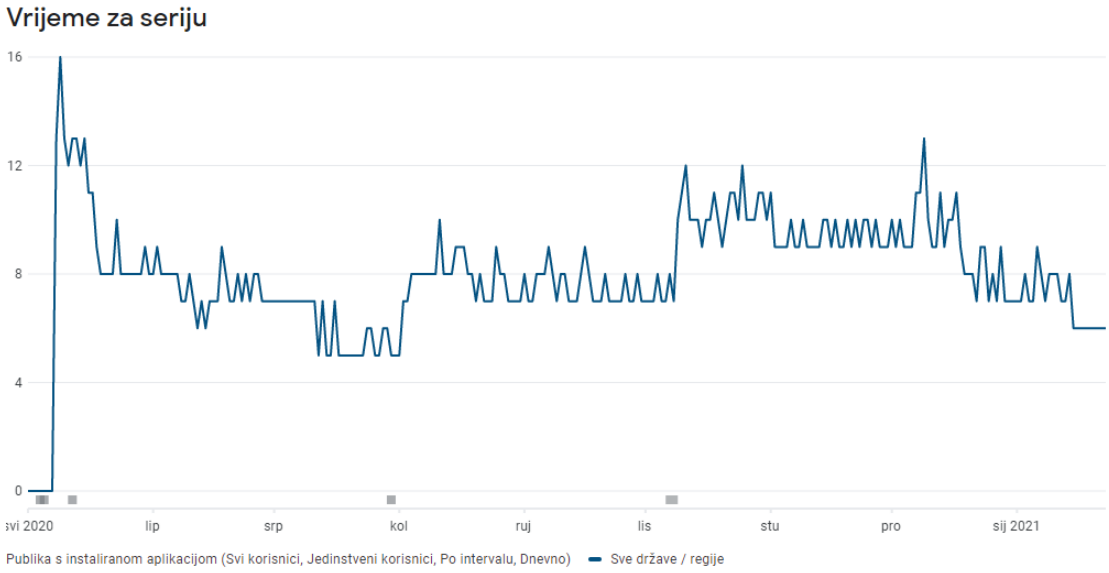


Slika 19. Nadzorna ploča Google Play Console

Izvor: Autor

6.3. Statistika igre u Google Play trgovini

Igra je na trgovinu Google Play objavljena početkom svibnja 2020. godine bez ikakvog marketinga ili oglašavanja. Igra je podijeljena među grupom prijatelja i ukupno je 48 korisnika. Od tih 48 korisnika 45 je novih, a 3 su ponovna korisnika koji su instalirali igru, 42 korisnika su izbrisala igru u nekom vremenskom razdoblju, a trenutno igru ima instalirano 6 korisnika. Igru je ocijenilo 8 korisnika i to s prosječnom ocjenom 4.375 što je 45,83% više u odnosu na prosječnu ocjenu u kategoriji ne zahtjevnih videoigara. Od 8 korisnika 6 je uz ocjenu ostavilo i pozitivnu recenziju.



Slika 20. Aktivni korisnici

Izvor: Autor

Slika prikazuje broj aktivnih korisnika igre u razdoblju od 1. svibnja 2020. godine do 24. siječnja 2021. godine, trenutni broj aktivnih korisnika je 6.

7. ZAKLJUČAK

Ovim radom prikazan je proces izrade android clicker igre. Ova igra zahtijeva svakodnevna poboljšanja, jer svaka nova verzija igre stvara sve kompleksniju igru koja postaje zanimljivija. Trenutno igra posjeduje sve bitne dijelove: interakciju objekata sa ostalim objektima, animacije, teksturiranje, zvuk. Izrada igre je zahtjevan i dugotrajan proces. Prikazani su svi dijelovi izrade igre od prvotne ideje, preko dizajniranja do realizacije ideje. Proces izrade igre uključuje mnoštvo različitih područja, poput crtanja, programiranja, animacije, izrade modela kako bi se dobila cjelina. Ovim radom prikazano je kako jedna osoba kombinira sva ta područja i sama izrađuje i kreira dijelove iz tih područja i zanimanja.

Možemo reći kako je cilj ovog rada ispunjen. Prikazan nam je način izrade vidoigre u besplatnom i jednim od najpopularnijih programa na svijetu te prikazano je kako se igra objavljuje na Google Play trgovini. Možemo vidjeti kako videoigra objavljenja bez ikakvog marketinga neće biti popularna sama od sebe jer je današnje tržište je popunjeno videoigrama pa se kreatori moraju se isticati igrom, ali i marketinškom kampanjom. Također se vidi da ne trebamo cijeli tim ljudi za sva područja animacije, modeliranje, programiranje da bi se napravila igra. Industrija videoigara je jedna od najvećih industrija u svijetu i naravno da možemo očekivati još veći rast i još veću potrebu za videoigrama, a i radnicima u industriji. Možemo pretpostaviti da će industrija videoigara u bliskoj budućnosti postati najveća industrija na svijetu jer se konstantno razvija i broj korisnika konstantno raste.

8. LITERATURA

[1] Unity Learn, https://unity3d.com/learn (15.7.2020)

[2] Microsoft Visual Studio , https://www.visualstudio.com/vs/ (15.7.2020)

[3] Photoshop User Guid, https://helpx.adobe.com/photoshop/user-guide.html (15.7.2020)

[4] Udemy tečaj izrade 2D mobilne igre, https://www.udemy.com/course/the-ultimate-guide-to-mobile-game-development-with-unity/ (15.7.2020)

[5] Microsoft C# vodič, https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/ (20.7.2020)

[6] Industrija videoigara, https://www.marketwatch.com/story/videogames-are-a-bigger-industry-than-sports-and-movies-combined-thanks-to-the-pandemic-11608654990 (12.1.2021)

Popis slika