



UNIVERZITET U NOVOM SADU,
PRIRODNO-MATEMATIČKI FAKULTET

SEMINARSKI RAD C

TEMA: RAZVOJ VIDEO IGRE U PROGRAMU UNITY



Profesor:
Vladimir Kurbalija

Student:
Andrej Rajkov

Novi Sad, 2024. godine

SADRŽAJ

2. Motivacija	3
2.1 Subway Surfers	3
2.2 Zašto 2D igra.....	4
3. Scene i Meniji	5
3.1 Startni meni.....	6
3.2 Glavna scena	7
4. Velika iluzija: Igrač se ne pomera	8
4.1 Pomeranje prepreka	9
4.2 Pomeranje šina.....	11
4.3 User input igrača	12
4.4 Animacija igrača	13
5. Sistem za poene (Scoring system)	14
5.1 Računanje najvećeg rezultata.....	14
6. Shop	15
7. Audio sistem	16
8. Zaključak.....	16

Motivacija

2.1 Subway Surfers

Subway Surfers je video igra za mobilne telefone gde igrač beskrajno trči. Igru su razvili Kiloo i SYBO Games, privatne kompanije sa sedištim u Danskoj.

U igri igrači imaju ulogu mladog crtača grafita koji je uhvaćen na delu i sada trči po železničkoj pruzi da bi pobegao od inspektora i njegovog psa.

Dok trči, igrač sakuplja novčiće, pojačanja i druge predmete i istovremeno izbegava sudar sa vozovima i ostalim preprekama. On može da skoči na voz ili surfuje na hoverbordu kako bi izbegao hvatanje.

Igra se završava kada se lik udari u prepreku, zgazi ga voz ili ga uhvati inspektor. Nagrade i novi likove mogu se dobiti u specijalnim događajima poput lova na reči.



2.2 Zašto 2D igra

Originalna igra je u 3 dimenzije, zašto onda preći u 2D okruženje?

Isključivanje fizike iz igre olakšava razvoj, ali zahteva novi pristup u osmišljavanju mehanika za zadržavanje zabave i izazova. U 2D formatu, interfejs je jednostavniji, a dizajn prepreka zahteva kreativnost radi održavanja dinamike.

S obzirom na to da zvanična 2D verzija igre Subway Surfers ne postoji, bilo je potrebno početi od nule i osmisliti sve elemente igre kako bi se prilagodili novom formatu. To uključuje dizajn likova, okoline, prepreka, zvukova i ostalih elemenata koji čine identitet igre. Bilo je potrebno pažljivo razmotriti kako će se ovi elementi uklopiti u 2D svet i kako će doprineti ukupnom iskustvu igre.

Prelaženjem u dvodimenzionalni svet, igrica je poprimila potpuno novi oblik i izgled, zadržavajući dobro poznatu i obožavnu mehaniku originala.

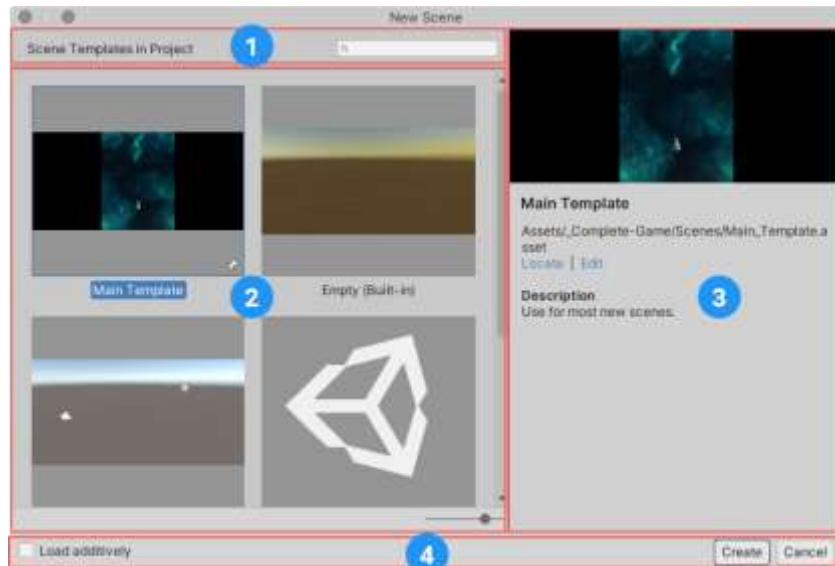


3. Scene i Meniji

Šta su scene u Unity softveru?

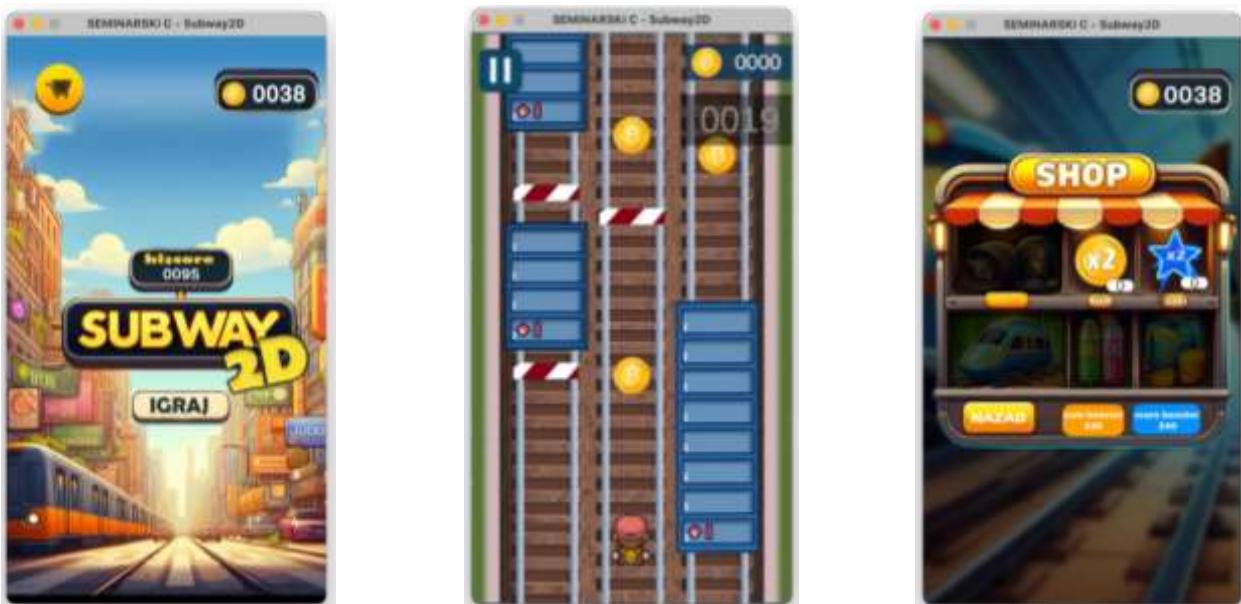
To je lako pitanje. Scene su sve što igrač treba da vidi na ekranu u jednom momentu.

Scena je mesto gde radite sa sadržajem u Unity-ju. To su resursi koji sadrže ceo ili deo igre. Na primer, možda ćete izgraditi jednostavnu igru u jednoj sceni, dok biste za složeniju igru mogli koristiti jednu scenu po nivou, svaka sa svojim okruženjem, likovima, preprekama, dekoracijama i korisničkim interfejsom.



Kao I u pozorištu, scene se menjaju po potrebi, a naš projekat ima 3 scene:

Start, Glavna scena i Shop meni



3.1 Startni meni

Prvi meni koji korisnik vidi kada pokrene igricu je startni meni. Zbog toga je jako bitno da ovaj meni bude vizuelno privlačan.

Ovde je primenjen jedan trik iz estetskih razloga, a i zbog lakoće dizajniranja. Dugmad koja vidite na sceni nisu prava dugmad, sve je jedna velika slika, a na mestima gde trebaju biti, su stavljena providna dugmad.



Osim vizuelnog aspekta, start meni nema preterano mnogo kompleksnih elemenata. Skripta za start meni sadrži metode za menjanje scena koje se pokreću klikom na dugmad, kao i učitavanje najvećeg rezultata i ukupnog zarađenog novca iz PlayerPrefs-a, o kojem ćemo detaljnije u delu o sistemu za poene.

```
public void Start()
{
    highscore = PlayerPrefs.GetFloat("highscore", 0);
    HighScoreText.text =
        Mathf.FloorToInt(highscore).ToString("D4");
    int ukupanNovac = PlayerPrefs.GetInt("novac", 0);
    NovacText.text = ukupanNovac.ToString("D4");
}

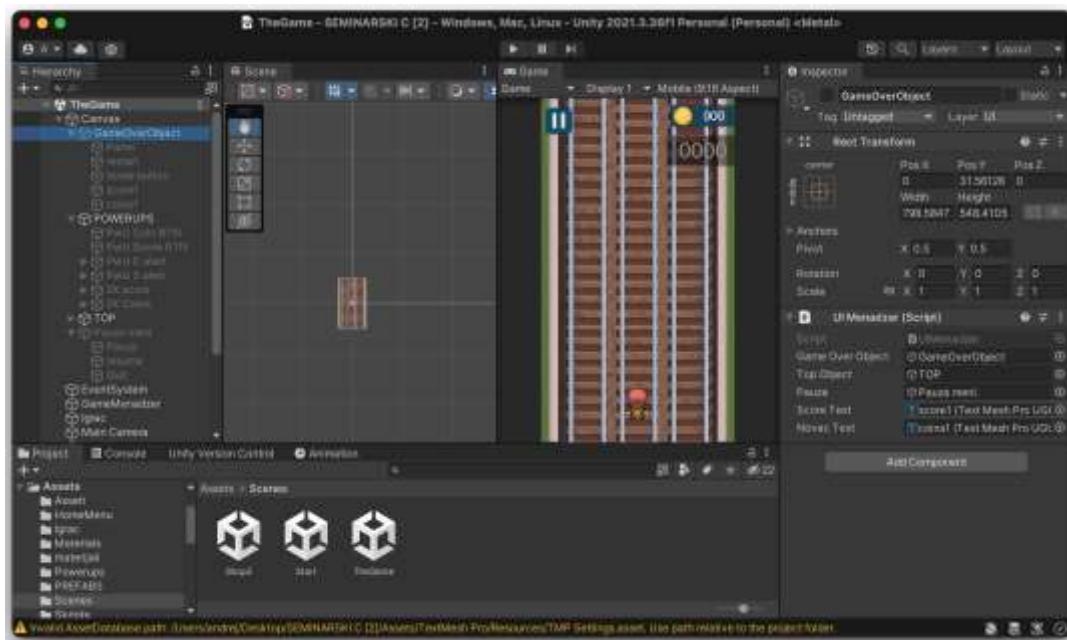
public void PlayGame()
{
    SceneManager.LoadScene(1);
}

public void OpenShop()
{
    SceneManager.LoadScene(2);
}

public void QuitGame()
{
    Debug.Log("kvit");
    Application.Quit();
}
```

3.2 Glavna scena

Glavna scena je ubedljivo najbitniji deo igre. Ovde se odigrava sva akcija i bez ove scene nema ni same igrice.



Osim što na ovoj sceni se odigrava sva akcija, na njoj su i pauza meni i game over screen. Oni su na samom početku neaktivni ali se aktiviraju pokretanjem skripte pri gubitku igre ili klikom na dugme za pauzu.

```
public void KrajIgre()
{
    player.gameObject.SetActive(false);
    spawner.gameObject.SetActive(false);
    spinner.gameObject.SetActive(false);
    spinner2.gameObject.SetActive(false);
    spawner2.gameObject.SetActive(false);
    brzinajpre = 0f;
    enabled = false;
    back.Stop();
    TextS.SetActive(false);
    TextC.SetActive(false);

    //prikazi score i novac u od trenutne igre
    deadMoney.text = novac.ToString("0'.'0");
    deadScore.text = Mathf.FloorToInt(skor).ToString("0'.'0");
    UpdateHiScore();
    Preprekeli obstacles = FindObjectOfType<Preprekeli>();
    foreach (var obstacle in obstacles)
    {
        Destroy(obstacle.gameObject);
    }
    TopObject.SetActive(false);
    GameOverObject.SetActive(true);
}
```

```
public void Pauziraj()
{
    Time.timeScale = 0;
    DisableT();
    GameManager.Instance.back.Pause();
    Pauza.SetActive(true);
}
```

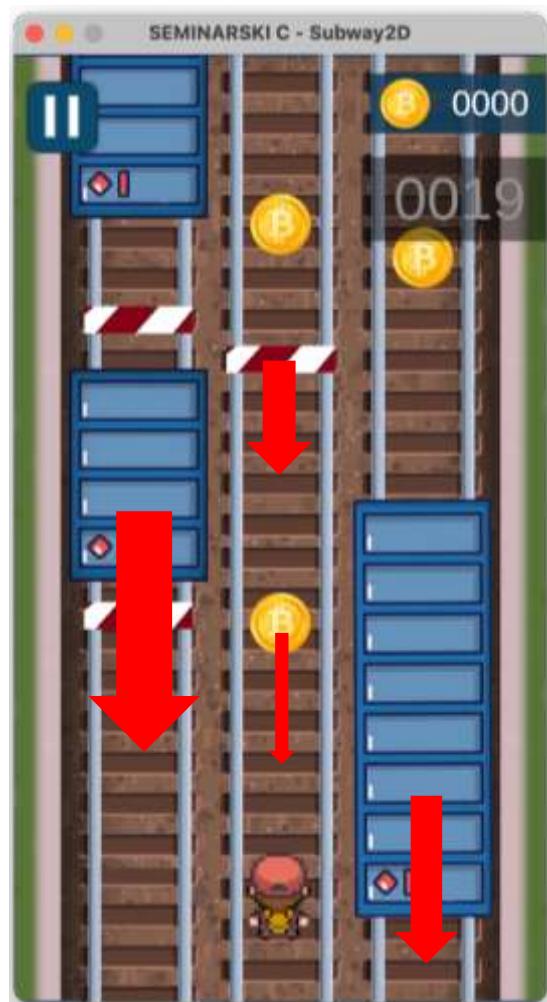
Isti je slučaj i za dugmad za pojačanja kupljena u prodavnici. Dugmad se prikazuju prvih 5 sekundi igre i nestaju nakon toga.

O sceni prodavnice ćemo malo kasnije, u delu gde ćemo takođe objasniti i pojačanja i kako ona rade.

4. Velika iluzija: Igrač se ne pomera

Iako je veliki Galileo Galilej dokazao da se Sunce ne okreće oko Zemlje, već Zemlja oko Sunca, u našem slučaju ta nauka ne važi.

Igrač se ne pomera, osim kad treba da se pomeri levo ili desno na ekranu, dok se šine, prepreke i novčići idu ka njemu.



Hajde da vidimo kako smo tu iluziju napravili:

4.1 Pomeranje prepreka

U ovoj igri imamo tri vrste prepreka: veliki vagon od voza, mali vagon od voza i crveno-bela rampa. Sve tri zajedno sinhronizovane sa šinama koje se pomeraju čine iluziju da igrač trči po mapi, a ne da mapa trči ka igraču.

```
public class Prepreke : MonoBehaviour
{
    private float donjaIvica;

    private void Start()
    {
        donjaIvica = Camera.main.ScreenToWorldPoint(Vector3.zero).y - 3f;
    }

    // Samo treba da se posera nadole u y-= 1 kad je van ekrana da nestane.
    void Update()
    {
        transform.position -= Vector3.down * GameManager.Instance.brzinaIgre * Time.deltaTime;
        if (transform.position.y < donjaIvica)
        {
            Destroy(gameObject);
        }
    }
}
```

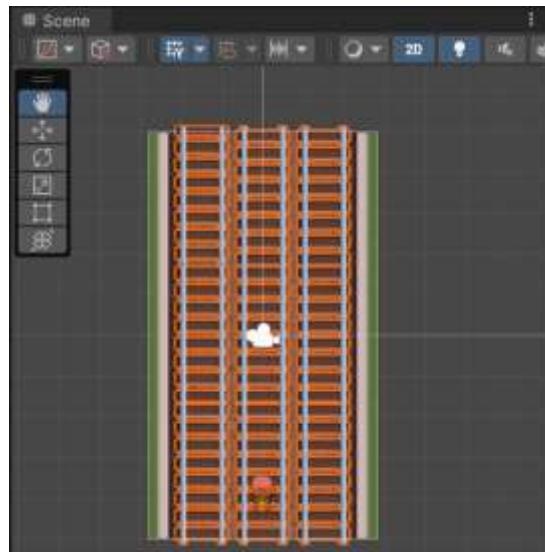
```
private void OnTriggerEnter2D(Collider2D other){
    if (other.CompareTag("Prepreka"))
    {
        GameManager.Instance.KrajIgre();
    }
    if (other.CompareTag("Novcic"))
    {
        GameManager.Instance.novcici += GameManager.Instance.coinmultiplier;
        GameManager.Instance.novacText.text = GameManager.Instance.novcici.ToString("04");
        //Sound kad je pokupljen novcic
        src.clip = sfx1;
        src.Play();
        Destroy(other.gameObject);
    }
}
```

Prepreke se stvaraju pomoću spawnera. Na mapi su tri spawner objekta, po jedan za svaku šinsku traku. One se stvaraju u gornjem delu mape, malo iznad ekrana, a zatim se pomeraju sve dok ne pređu donju ivicu ekrana, kada se uništavaju.

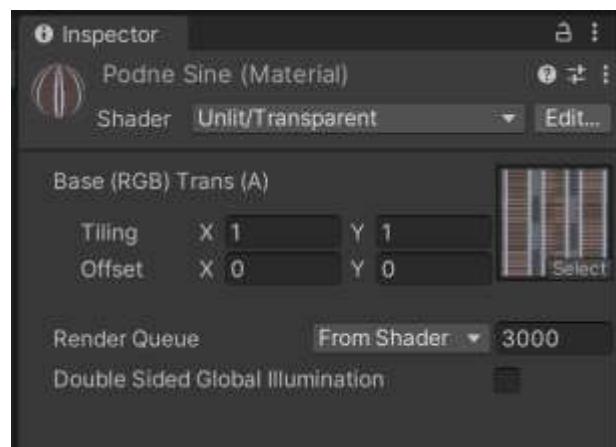
Spawner stvara prepreke ili novčiće na osnovu random šanse koja mu je zadata, a trenutno spawner je naklonjen više ka stvaranju novčića, a može se podesiti da stvara više prepreka i time bi igrica bila izazovnija za korisnika.



4.2 Pomeranje šina



Da bismo postigli potpunu iluziju da se samo igrač kreće, a ne cela scena, implementirali smo pokretanje šina. Umesto klasičnog **Sprite Rederer-a** koristili smo **Mesh Renderer** kako bismo omogućili beskonačno ponavljanje pokreta šina. Za pokretanje šina koristimo skriptu nazvanu **VozneSine.cs**.



```
public class sinenavozu : MonoBehaviour
{
    private MeshRenderer meshRenderer;
    private void Awake()
    {
        meshRenderer = GetComponent<MeshRenderer>();
    }

    // Update is called once per frame
    void Update()
    {
        float speed = GameManager.Instance.brzinaIgre/30;
        meshRenderer.material.mainTextureOffset += Vector2.up * speed * Time.deltaTime;
    }
}
```

4.3 User input igrača

Kao i u originalnoj igri Subway Surfers, igrač se kreće levo i desno na šinama, ne proizvoljno već na tri trake, tražeći prostor bez prepreka ili jureći novčiće.

Igrač se pomera pomoću strelica na tastaturi $\leftarrow \rightarrow$, a skripte pomeraju igrača i vode računa da on ne izđe sa mape.

```

7  public class Movement : MonoBehaviour
8  {
9      private Transform Player;
10
11     private bool Lane1 = false;
12     private bool Lane2 = true;
13     private bool Lane3 = false;
14
15     private void Start()
16     {
17         Player = GetComponent<Transform>();
18         Vector2 position = transform.position;
19         position.x = 0;
20         transform.position = position;
21     }
22
23 }
```

```

void Update()
{
    if (Input.GetKeyDown(KeyCode.LeftArrow))
    {
        //ako je u prvoj leđu ne radi nista
        if (Lane2)
        {
            Lane2 = false;
            Lane1 = true;
            PomeriNa1();
        }
        if (Lane3)
        {
            Lane3 = false;
            Lane2 = true;
            PomeriNa2();
        }
    }
    else if (Input.GetKeyDown(KeyCode.RightArrow))
    {
        if (Lane2)
        {
            Lane2 = false;
            Lane3 = true;
            PomeriNa3();
        }
        if (Lane1)
        {
            Lane1 = false;
            Lane2 = true;
            PomeriNa2();
        }
    }
}
```

```

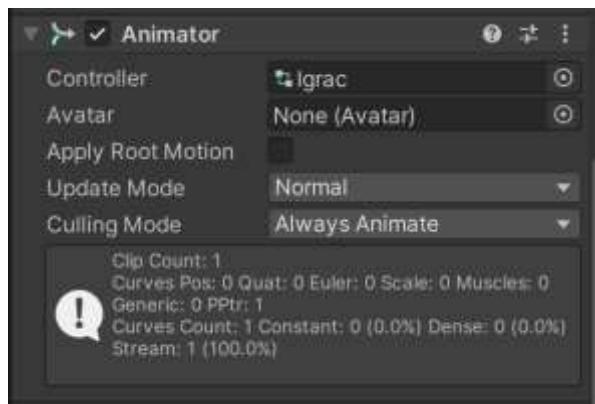
void PomeriNa1()
{
    Vector2 position = transform.position;
    position.x = (float)-1.5;
    transform.position = position;
}

void PomeriNa2()
{
    Vector2 position = transform.position;
    position.x = (float)0;
    transform.position = position;
}

void PomeriNa3()
{
    Vector2 position = transform.position;
    position.x = (float)1.5;
    transform.position = position;
}
```

4.4 Animacija igrača

Kako bi igrač delovao fluidnije i življe, a i kako bi doprineli iluziji da se igrač zapravo kreće, dodata je i animacija igrača.



Animacija se sastoji od 4 slike koje se menjaju na svakih $\frac{1}{4}$ sekunde.



5. Sistem za poene (Scoring system)

Rezultat dodaje takmičarski duh svakoj video igri.

Poeni u našog igri se računaju pomoću GameManager skripte, na osnovu ubrzanja igre, proteklog vremena i score multiplera (koji nam treba pri aktivaciji pojačanja u igri).

```
private void Update()
{
    brzinaIgre += ubrzanjeIgre*Time.deltaTime;
    skor += (ubrzanjeIgre * Time.deltaTime*8)*scoremultiplier;
    scoreText.text = Mathf.FloorToInt(skor).ToString("D4");
}
```

Novčići su takođe jako bitan element, jer pomoću njih možemo da kupujemo pojačanja.

```
private void OnTriggerEnter2D(Collider2D other){
    if (other.CompareTag("Novcic"))
    {
        GameManager.Instance.novcici+=GameManager.Instance.coinmultiplier;
        GameManager.Instance.novacText.text = GameManager.Instance.novcici.ToString("D4");
    }
}
```

5.1 Računanje najvećeg rezultata

Highscore je najviši postignuti rezultat u video igri. Kako bi zabeležili najveći rezultat i ukupne zarađene novce korišćena je klasa PlayerPrefs.

PlayerPrefs je klasa koja čuva igračke postavke između sesija igre, omogućavajući skladištenje string, float i integer vrednosti u korisnički registar platforme.

```
private void UpdateHiScore()
{
    float hi = PlayerPrefs.GetFloat("highscore", 0);
    if (skor > hi)
    {
        hi = skor;
        PlayerPrefs.SetFloat("highscore", hi);
    }
    int ukupanNovac = PlayerPrefs.GetInt("novac", 0);
    ukupanNovac += novcici;
    PlayerPrefs.SetInt("novac", ukupanNovac);
}
```

6. Shop

Shop je digitalna prodavnica unutar igre u kojoj igrači mogu kupovati virtualne predmete koristeći ugrađenu valutu igre ili stvarni novac.

U našem slučaju, koristićemo novac koji smo zaredili preskakajući vagone i potrošiti ga na
Coin booster – pri aktivaciji svaki prikupljeni novčić vredi duplo
Score booster – pri aktivaciji vaši poeni se dupliraju

I ovde ćemo koristiti **PlayerPrefs** skriptu za trgovinu novcem i čuvanje podataka koliko specijalnih unapređenja imamo

```
public void KupiCoins(){
    //skida sa racuna
    if(ukupanNovac>=30){
        ukupanNovac-=30;
        PlayerPrefs.SetInt("novac", ukupanNovac);
        NovacText.text = ukupanNovac.ToString("04");
        //dodaje powerup coinsbooster
        c2+=1;
        PlayerPrefs.SetInt("c2", c2);
        C2Text.text = c2.ToString();
    }
}

public void KupiScore(){
    //skida sa racuna
    if(ukupanNovac>=40){
        ukupanNovac-=40;
        PlayerPrefs.SetInt("novac", ukupanNovac);
        NovacText.text = ukupanNovac.ToString("04");
        //dodaje powerup scorebooster
        s2+=1;
        PlayerPrefs.SetInt("s2", s2);
        S2Text.text = s2.ToString();
    }
}
```



7. Audio sistem

Svaka dobro urađena igrica ima dodat zvuk. U nasem slučaju to su zvuk na sakupljeni novčić, muzika koja ide u pozadini i podiže atmosferu, kao i zvuk kad igrač izgubi (game over sound).



```
//Sound kad je pokupljen novcic
src.clip = sfx1;
src.Play();
```

```
public void NovaIgra()
{
    back.clip = muzika;
    back.Play();
    back.loop = true;
```

8. Zaključak

Prateći ovaj projekat, videli smo kako izgleda proces pravljenja 2d igre u unity softveru. U ovom slučaju veći akcenat je stavljen na mehanizme igre i same skripte koje je pokreću, a zanemareni su artistički elementi (dizajn player modela, dizajn scene, dizajn menija kao i audio dizajn).

Iako je primarni fokus na igračevom izbegavanju prepreka, sve što ga okružuje – player movement, powerups, score management i audio dizajn upotpunjaju igru i čine je još interaktivnijom.

Naravno Unity poseduje dosta više mogućnosti za pravljenje igara i ova igra je demonstracija samo nekih od mogućnosti ovog game engina.

Literatura korišćena

- I. [2D Endless Runner - Unity Tutorial - YouTube](#)
- II. [2D Top Down Movement UNITY Tutorial - YouTube](#)
- III. [COIN COUNTER in Unity 2D Tutorial - YouTube](#)
- IV. [Endless Runner Tutorial for Unity | Sharp Coder Blog](#)
- V. [How to Make 2D Endless Runner Game in Unity - Dino Game - YouTube](#)
- VI. [How To Make An Endless Runner In Unity - YouTube](#)
- VII. [How to make Dino Game in Unity \(Complete Tutorial\) 🦕🦖 - YouTube](#)
- VIII. [I REMADE Chicken Invaders in Unity - YouTube](#)
- IX. [Increasing the Difficulty - Build a 2D Endless Runner in Unity - 2D Tutorial Series #6 - YouTube](#)
- X. [Making a 2D endless runner? : r/Unity3D](#)
- XI. [Question - How to make the illusion of movement in a vertical 2d runner - Unity Forum](#)
- XII. [Runner, a Unity C# Tutorial](#)
- XIII. [START MENU in Unity - YouTube](#)
- XIV. [Unity Pause Tutorial: Best Practices to Pause Your Game the Right Way - YouTube](#)
- XV. [Unity Vertical Infinite Scrolling Background + Camera Follow Tutorial - YouTube](#)

PRILOG UZ SEMINARSKI RAD

Source code + Build za Windows, Intel x64

[[Google Drive link](#)]