

Učenci: Jakov Biškup i Karla Hajdarović

Razred: 8

Škola: OŠ "Petar Zrinski" Šenkovec

Mentorica: Lidija Matoša

Vanjski mentor: Mateo Črep

SADRŽAJ:

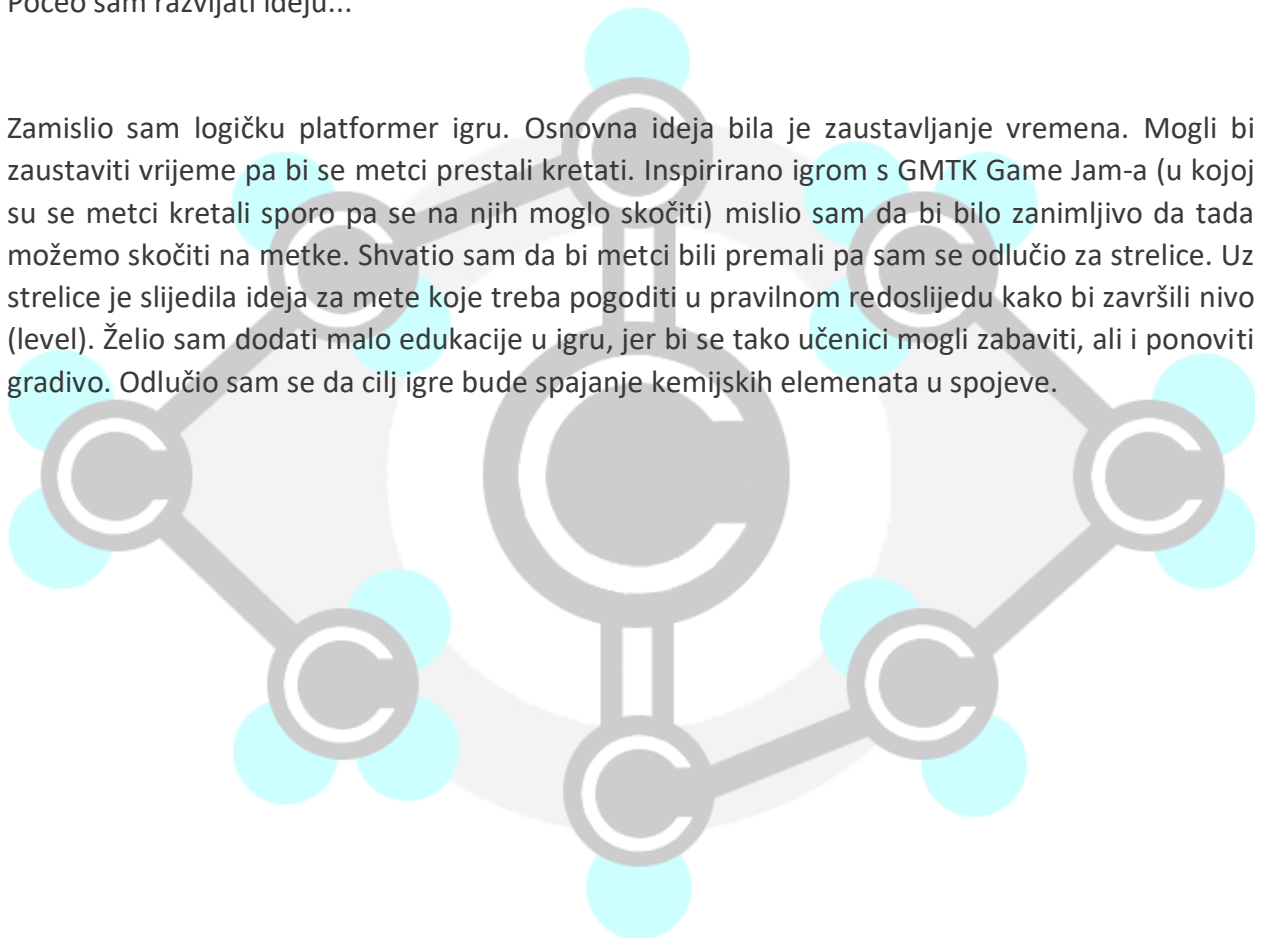
UVOD:.....	3
REALIZACIJA IDEJE.....	4
OBJAŠNJENJE PROGRAMA.....	5
Kretanje.....	5
Zaustavljanje vremena.....	6
Pucanje.....	8
Metu.....	9
Meni.....	11
DIZAJN.....	12
PLANOVI ZA BUDUĆNOST.....	14
LITERATURA.....	15
Korišteni tutoriali:.....	15
Korištene stranice:.....	15
Glazba:.....	15
POSEBNE ZAHVALE:.....	16

UVOD:

Želim predstaviti logičko-edukativnu igru namijenjenu ponavljanju gradiva kemije za sedmi razred.

Naučio sam koristiti Unity kroz ljetovanje te sam želio produbiti svoja znanja. Kada sam čuo da će se u našoj školi održavati natjecanje u razvoju softvera, odlučio sam iskoristiti priliku i napraviti igru. Počeo sam razvijati ideju...

Zamislio sam logičku platformer igru. Osnovna ideja bila je zaustavljanje vremena. Mogli bi zaustaviti vrijeme pa bi se metci prestali kretati. Inspirirano igrom s GMTK Game Jam-a (u kojoj su se metci kretali sporo pa se na njih moglo skočiti) mislio sam da bi bilo zanimljivo da tada možemo skočiti na metke. Shvatio sam da bi metci bili premali pa sam se odlučio za strelice. Uz strelice je slijedila ideja za mete koje treba pogoditi u pravilnom redoslijedu kako bi završili nivo (level). Želio sam dodati malo edukacije u igru, jer bi se tako učenici mogli zabaviti, ali i ponoviti gradivo. Odlučio sam se da cilj igre bude spajanje kemijskih elemenata u spojeve.



REALIZACIJA IDEJE

Za pomoć u grafičkom dizajnu zamolio sam prijateljicu Karlu Hajdarović. Igru sam odlučio napraviti u programu Unity koristeći jezik C#. Za programiranje koristio sam Visual Studio 2019. Uz standardne dijelove Unity-a koristio sam i besplatan Asset TextMesh PRO koji omogućuje lakše i bolje uređivanje UI-a.



OBJAŠNJENJE PROGRAMA

Kretanje

Igrač na sebi ima **Rigidbody2D** i **BoxCollider2D** gotove dijelove Unity-a zbog kojih na njega djeluje gravitacija te ne propada kroz pod (koji na sebi također ima **BoxCollider2D**). Za kretanje koristimo **transform.Translate** koji pomiče igrača i **Input.GetAxis("Horizontal")** koji vraća 1, 0 ili -1 ovisno o tome koje tipke igrač pritisne ('A' ili 'D' ili oboje). Kako se ne bi mogli kretati kroz zid koristimo naše funkcije **OnWallLeft()** i **OnWallRight()** koje koriste **Raycast** kako bi saznale je li igrač pokraj zida.

```
if ((!OnWallLeft() || transform.localScale.x > 0) && (!OnWallRight() || transform.localScale.x < 0)) //kako igrač nebi ptoazio kroz zidove
{
    transform.Translate(new Vector3(movementSpeed * Input.GetAxis("Horizontal") * Time.deltaTime, 0, 0)); // kretanje po x osi
}
```

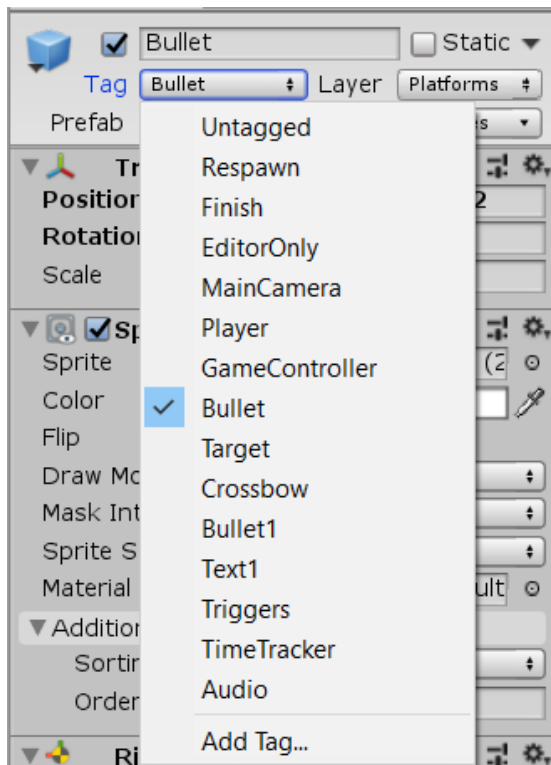
Za skakanje koristimo **Input.GetButtonDown()** funkciju kako bi dobili input i **Rigidbody2D.AddForce()** kako bi pokrenuli igrača.

```
if (Input.GetKeyDown("w"))
{
    if (OnGround())
    {
        rigidBody.AddForce(new Vector3(0, jumpForce, 0));
    }
}
```

Zaustavljanje vremena

Za zaustavljanje vremena koristimo **GameObject.FindObjectsWithTag()** kako bi pronašli sve strelice.

```
void Start ()
{
    bullets = new GameObject[GameObject.FindGameObjectsWithTag("Bullet").Length];
}
```



Tada svim strelicama **Rigidbody2D.bodyType** postavimo na **Static** kako se ne bi mogle kretati.

```
if (timeStoped && !flipFlop /*samo kada vrijeme jos uvijek nije bilo zaustavljeno*/)
{
    //stopTime.Play(); //odigra zvuk zaustavljanja vremena
    for (int i = 0; i < bullets.Length; i++)
    {
        bullets[i].GetComponent<Rigidbody2D>().bodyType = RigidbodyType2D.Static; //na te strelice više ne djeluje sile pa time zastaju na mjestu
        cameraMain.backgroundColor = Color.gray; //mijenja boju pozadine
    }
    for (int i = 0; i < bullets1.Length; i++)
    {
        bullets1[i].GetComponent<Rigidbody2D>().bodyType = RigidbodyType2D.Static; //na te strelice više ne djeluje sile pa time zastaju na mjestu
        cameraMain.backgroundColor = Color.gray; //mijenja boju pozadine
    }
    mem = weapon.canShoot; //pamtimo kakvo je stanje bilo prije zaustavljanja vremena
    flipFlop = true;
}
```

*bullets su strelice koje ispaljuje igrač, a bullets1 su strelice koje ispaljuje samostrel

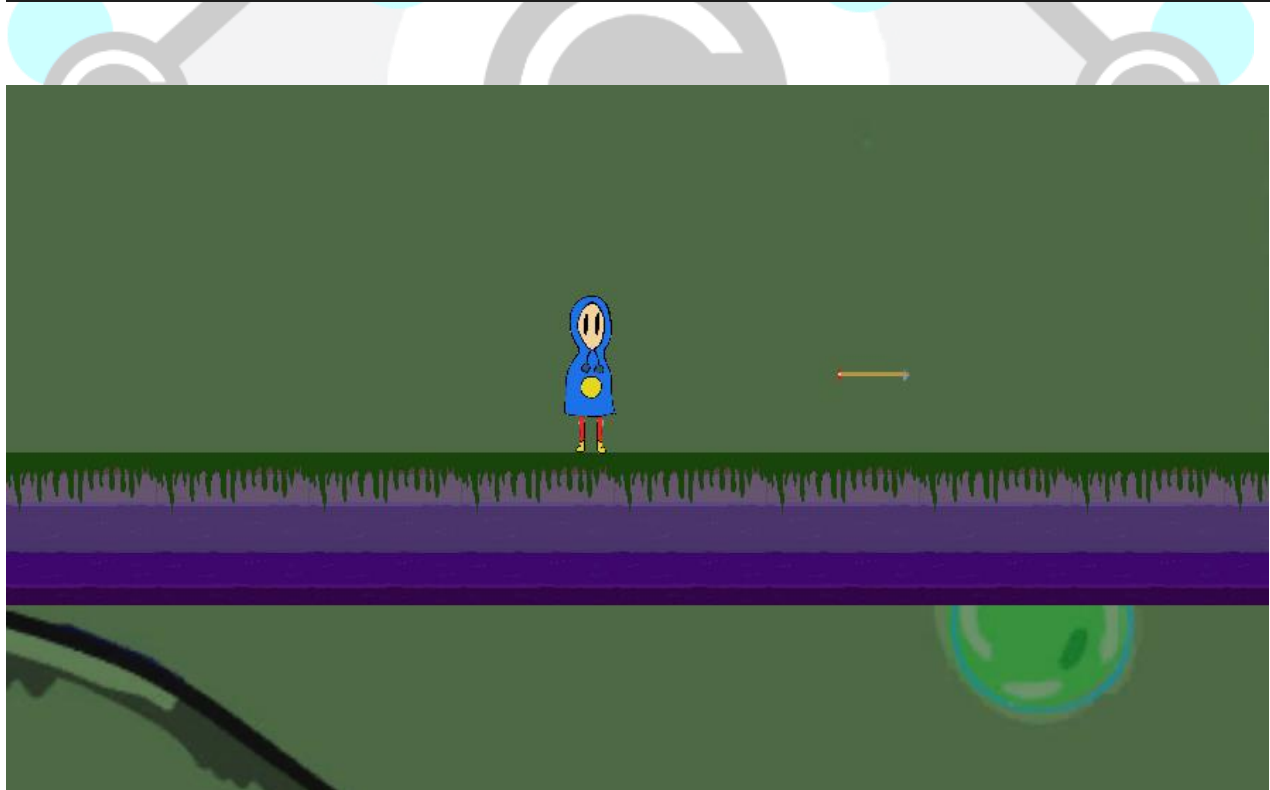


Pucanje

Kada igrač pritisne gumb 'L' koristimo **Instantiate()** kako bi stvorili strelicu. U slučaju da je vrijeme zaustavljeno ili strelica još putuje **canShoot** se postavlja na **false** te tada igrač ne može pucati.

```
void Shoot()
{
    Instantiate(bullet, firePoint.position, firePoint.rotation); //stvaranje strelice
}

// Update is called once per frame
0 references
void Update () {
    if (Input.GetButtonDown("Fire1") && canShoot)
    {
        canShoot = false;
        Shoot();
    }
}
```



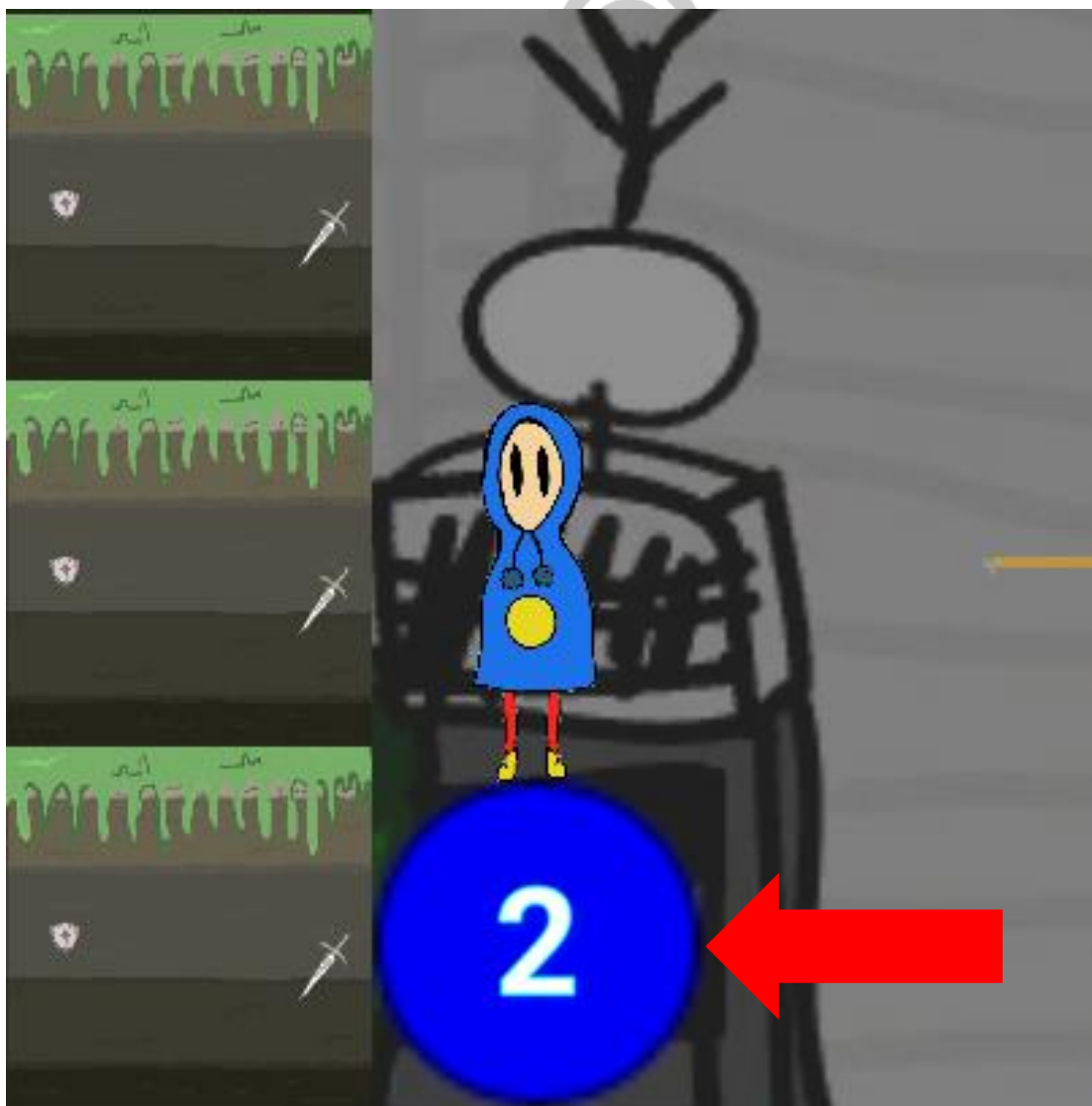
Metre

U sceni imamo objekt **Target Manager** koji pamti koju metu trebamo pogoditi. Kada je meta pogođena, program provjeri je li ta meta slijedeća u redoslijedu, ako je redoslijed dobar, meta pozeleni, a ako nije onda resetira sve mete. U slučaju da je **order = -1** ta meta nije dio spoja pa pocrveni i zamijeni svoj tekst imenom elementa koji je na meti.

```
void Update () {
    if (hit)
    {
        if (order == -1)
        {
            spriteRenderer.color = Color.red;
            for (int i = 0; i < targets.Length; i++) {
                if (targets[i].GetComponent<Target>().order != -1)
                {
                    targets[i].GetComponent<SpriteRenderer>().color = Color.blue;
                }
                targetManagment.currentTarget = 0;
                hit = false;
            }
            myText.enabled = false;
            myText2.enabled = true;
        }
        else if (targetManagment.currentTarget == order)
        {
            spriteRenderer.color = Color.green;
            targetManagment.currentTarget += 1;
            hit = false;
        }
    }
    else
    {
        for (int i = 0; i < targets.Length; i++)
        {
            if (targets[i].GetComponent<Target>().order != -1)
            {
                targets[i].GetComponent<SpriteRenderer>().color = Color.blue;
            }
            targetManagment.currentTarget = 0;
            hit = false;
        }
    }
}
```

```
public class TargetManagement : MonoBehaviour {  
  
    public int currentTarget = 0;  
    public int nummberOfTargets;  
  
    // Update is called once per frame  
    0 references  
    void Update () {  
        if (currentTarget == nummberOfTargets)  
        {  
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);  
        }  
    }  
}
```

*SceneManager.LoadScene() otvori scenu



Meni



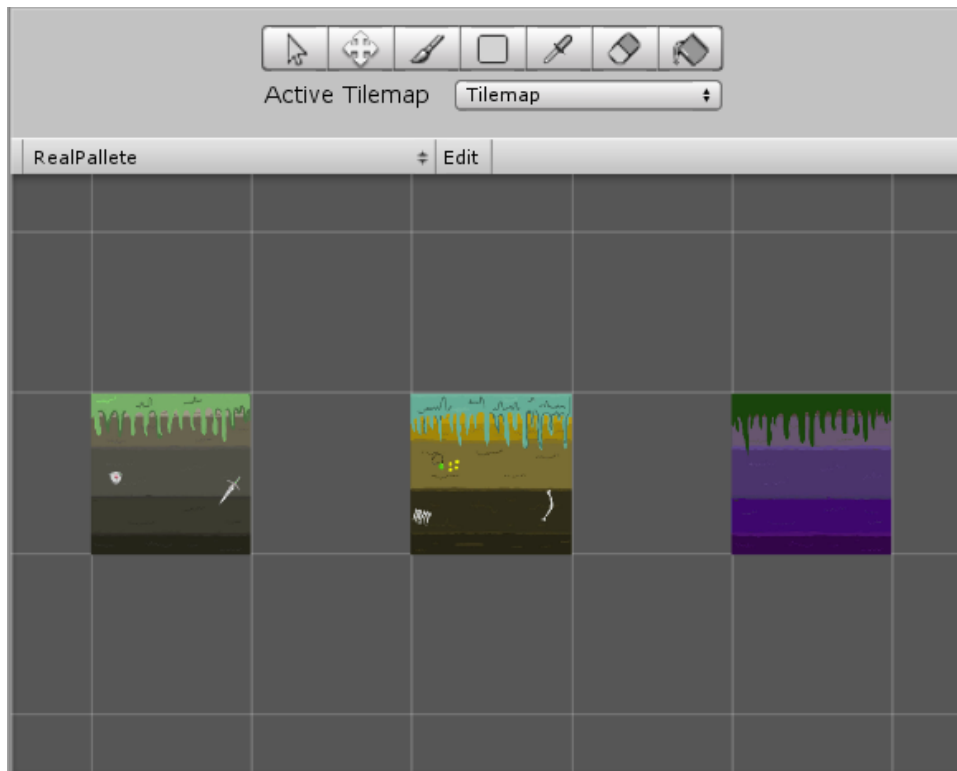
Igraj - pokreće igru

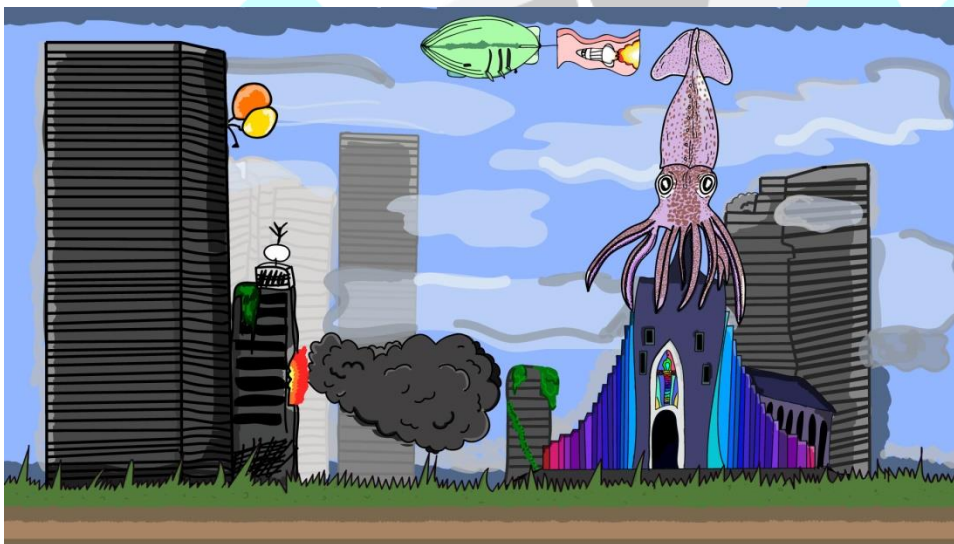
Postavke – otvara postavke u kojima možemo podesiti glasnoću

Izađi - izlazi iz igre

DIZAJN

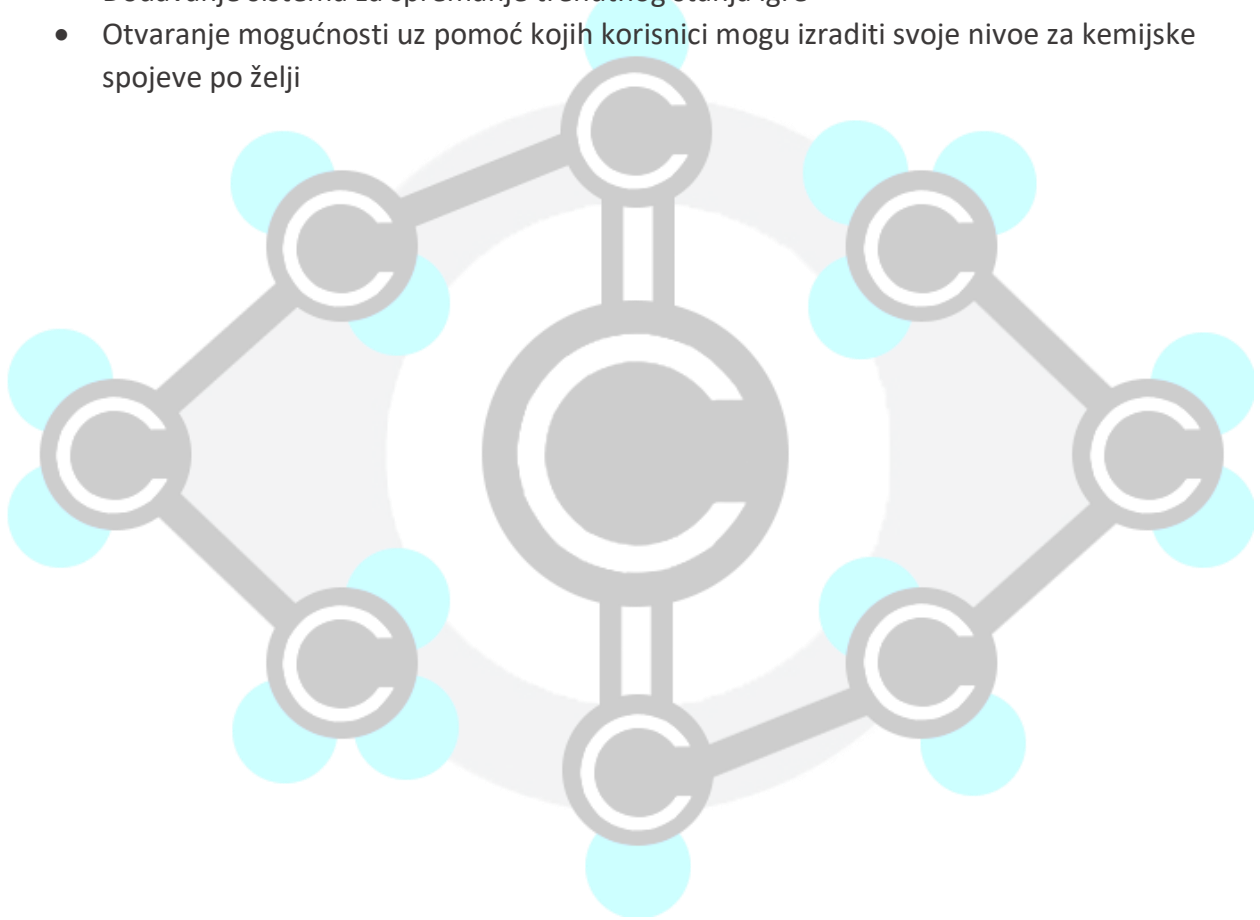
Za nivoe (levele) smo koristili tri različite pozadine. Također smo koristili i tri vrste tla koje sam postavljao uz pomoć Unity-ovog gotovog Tileset sistema. Karla ih je nacrtała u programu Adobe Illustrator.





PLANOVI ZA BUDUĆNOST

- Dodavanje još nivoa različitih razina složenosti
- Dodavanje još pozadina i vrsta tla
- Dodavanje posebnih slika za zidove, podzemne dijelove platforma i mete
- Dodavanje animacija za igrača i samostrele
- Dodavanje još elementa nivoa (protivnici, zamke...)
- Dodavanje sistema za spremanje trenutnog stanja igre
- Otvaranje mogućnosti uz pomoć kojih korisnici mogu izraditi svoje nivoe za kemijske spojeve po želji



LITERATURA

Korišteni tutoriali:

https://www.youtube.com/watch?v=ryISV_nH8qw

<https://www.youtube.com/watch?v=wkKsl1Mfp5M>

https://www.youtube.com/watch?v=zc8ac_qUXQY

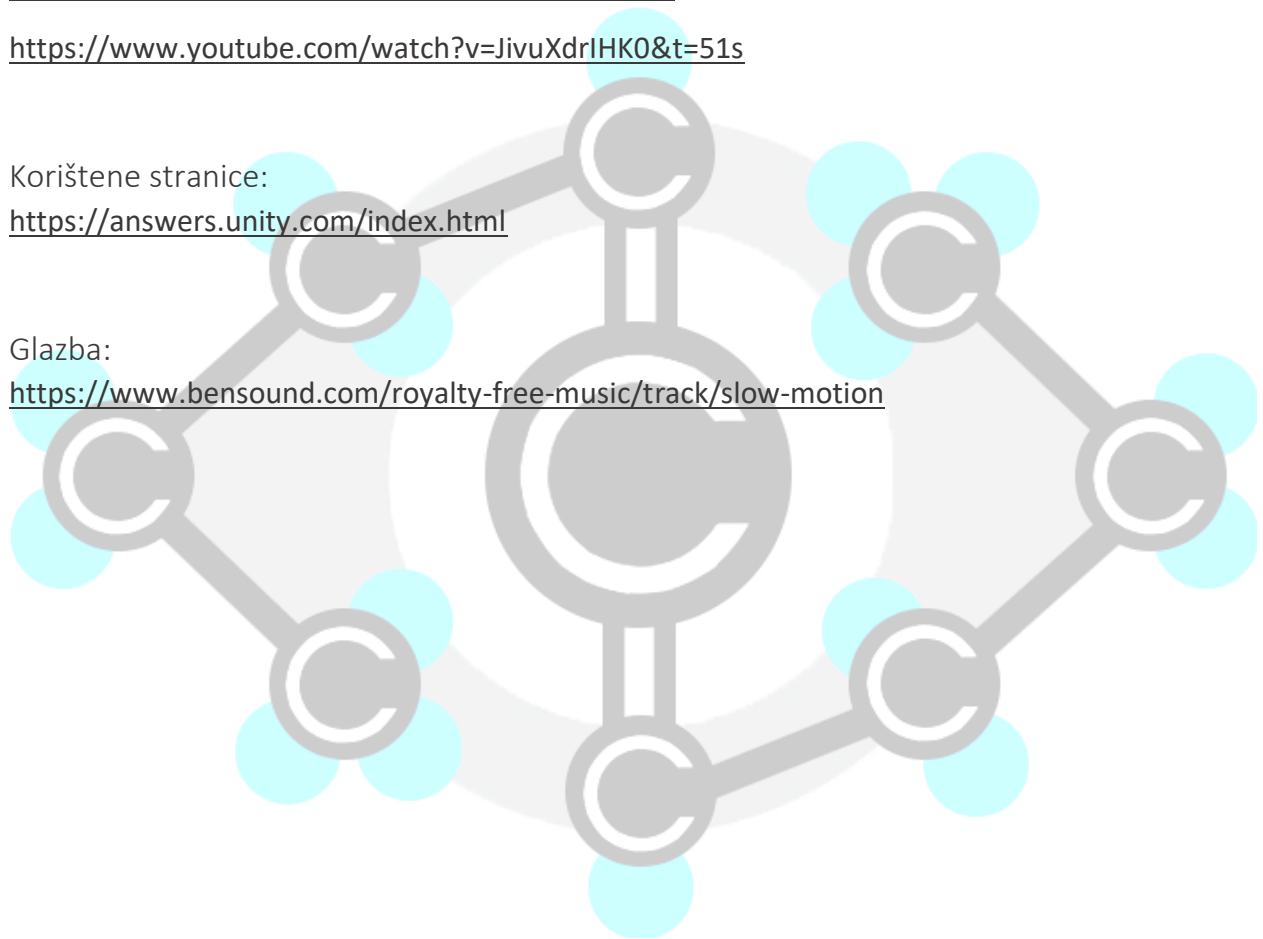
<https://www.youtube.com/watch?v=JivuXdrIHK0&t=51s>

Korištene stranice:

<https://answers.unity.com/index.html>

Glazba:

<https://www.bensound.com/royalty-free-music/track/slow-motion>



POSEBNE ZAHVALE:

- Mentorima Lidiji Matoša i Mateu Črepu koji su mi pomogli razviti ideju za igru
- Bratu Iliji Biškupu koji mi je pomogao napraviti logo u Adobe Photoshopu

