

Cuprins

Cuprins

<u>1. Tema proiectului</u>	<u>3</u>
<u>2. Descrierea jocului</u>	<u>3</u>
<u>2.1 Modul de utilizare a aplicației</u>	<u>3</u>
<u>2.2 Meniul principal</u>	<u>4</u>
<u>2.3 Jocul</u>	<u>5</u>
<u>2.4 Meniul de pauza.....</u>	<u>6</u>
<u>3.Tehnologii</u> <u>HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx"utilizate</u>	<u>7</u>
<u>4.Dezvoltarea jocului</u>	<u>7</u>
<u>4.1.Modelarea jocului</u>	<u>7</u>
<u>4.2.Codul sursa</u>	<u>..8</u>
<u>4. HYPERLINK "file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx"</u>	
<u>HYPERLINK "file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx"3 HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" HYPERLINK</u>	
<u>"file:///E:/Must/Facultate/SM/Norbecar/Norbecar/ProiectSM.docx" Probleme intampinate</u>	<u>10</u>

<u>5.Concluzii</u>	11
<u>6.Bibliografie</u>	11

1. Tema proiectului

Proiectul consta in realizarea unui joc 3D in mediul de dezvoltare **UNITY**.1]
Jocul ales este unul foarte cunoscut, anume Tetris. Am facut aceasta alegere deoarece este un joc vechi,am jucat mult acest joc si este captivant.[2]

Provocarea a constat in faptul ca este o tehnologie noua pentru mine si a fost necesar sa ma documentez mult pentru a duce la bun sfarsit jocul.

2. Descrierea jocului

2.1 Modul de utilizare a aplicației

La pornirea jocului prima data va aparea meniul principal, format din trei butoane, si anume: butonul de START care va trimite utilizatorul in joc, butonul HELP unde ne sunt prezentate combinatiile de taste utilizate in joc si butonul QUIT care va duce la parasirea jocului.

Jocul porneste cand primul buton este apasat. Scopul jocului este de a reusi sa positionam cat mai multe block-uri inainte de a atinge linia care va duce la oprirea jocului.

Jocul se poate controla folosind urmatoarea combinatie de butoane:

- Butoanele W/A/S/D pentru mutarea block-ului
- Butoanele Q/E pentru rotirea block-ului
- SPACE pentru aruncarea block-ului
- Camera se va misca prin folosirea mouse-ului
- SHIFT pentru rotirea camerei
- ESC pentru a accesa meniul de pauza

Jocul se termina atunci cand un block va depasi linia positionata deasupra.

2.2 Meniul principal

Este constituit din trei butoane: START, HELP, QUIT.

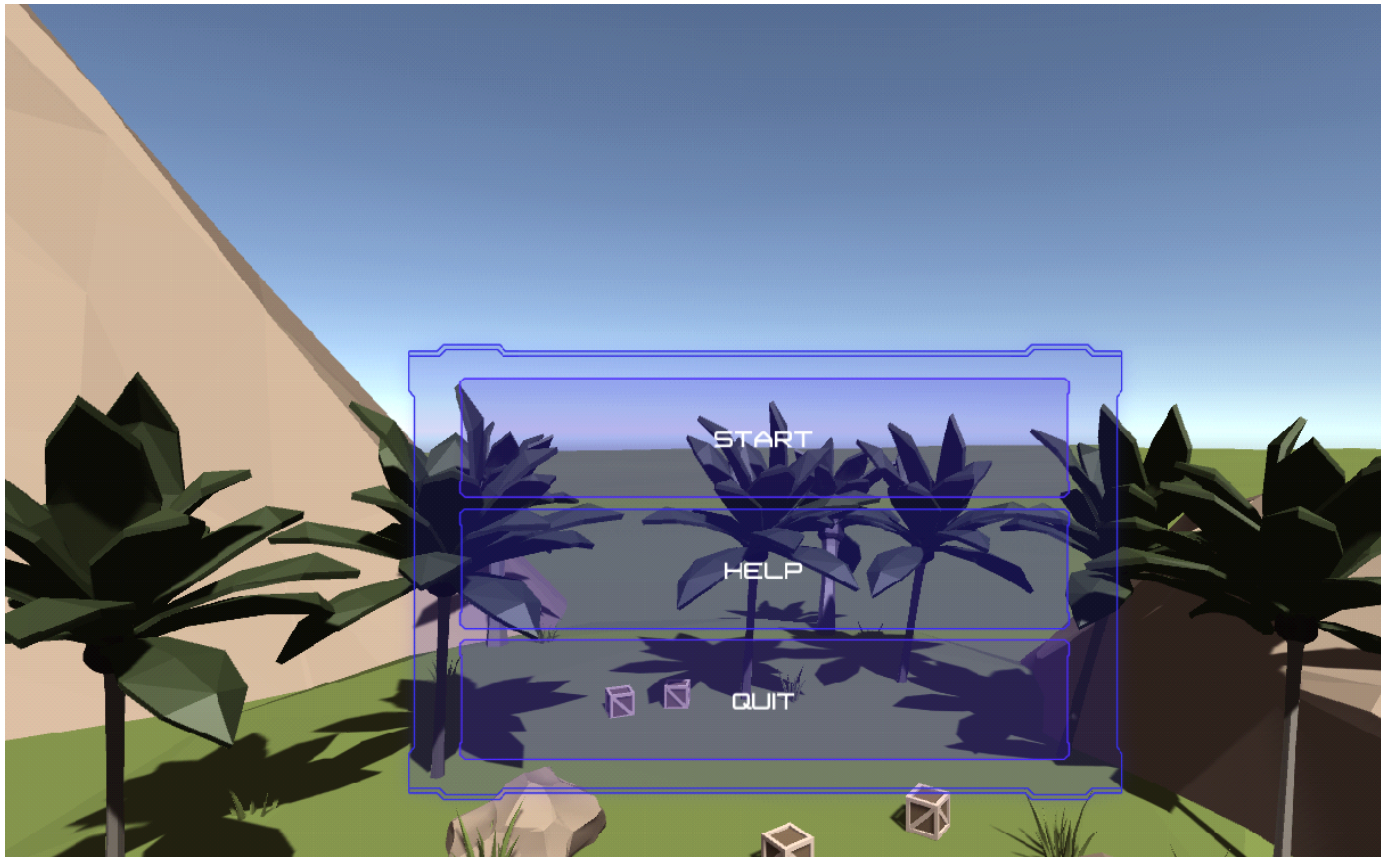


Figura 1-Meniul principal

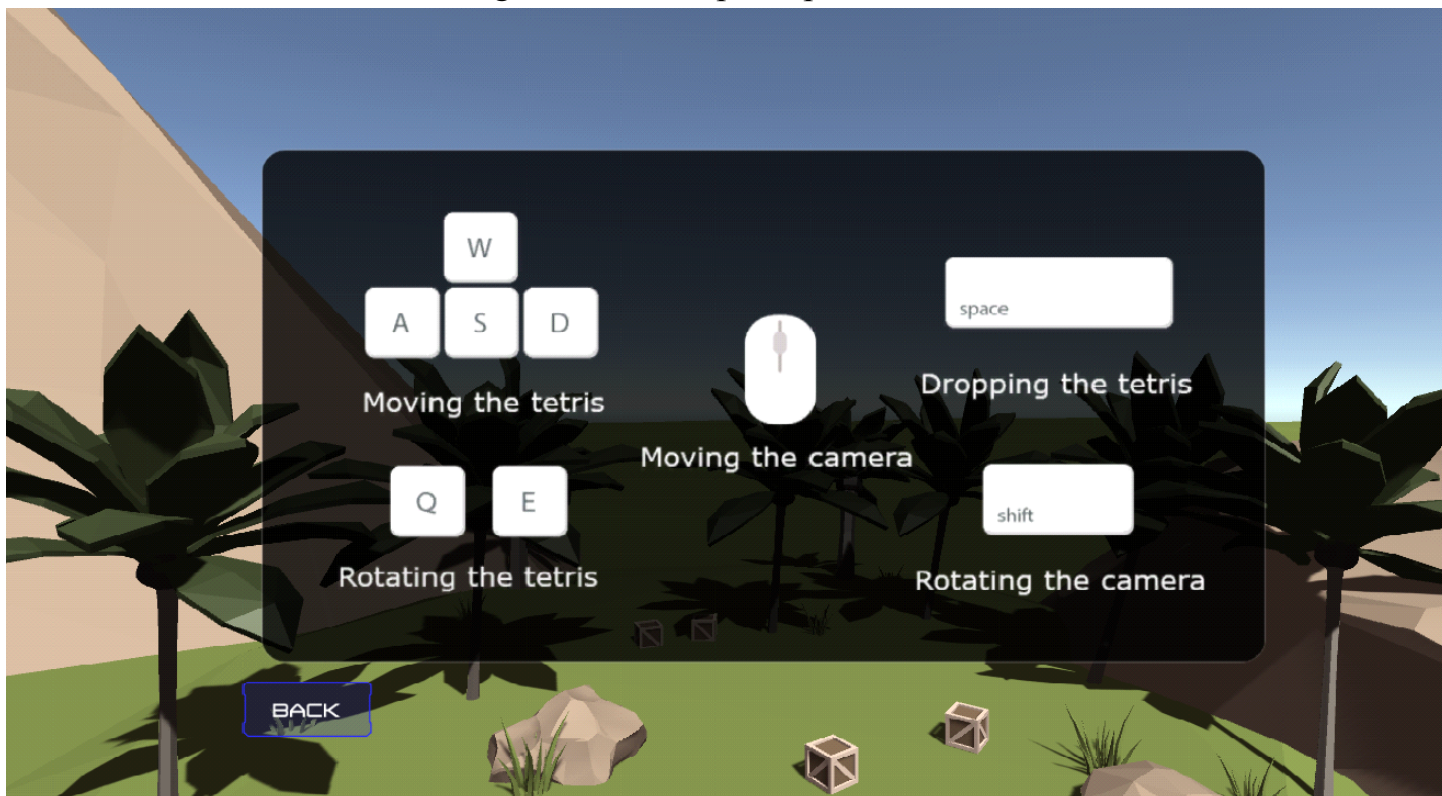


Figura 2-Meniul butonului HELP

2.3 Jocul

Jocul genereaza aleator block-uri pe care utilizatorul trebuie sa le aranjeze astfel incat sa reuseasca sa genereze un scor cat mai mare inainte de a depasi linia pozitionata deasupra.



Figura 3-Inceperea jocului

Fiecare block asezat cu succes va adauga 5 puncte la scor. Jocul se va opri in momentul in care linia pozitionata deasupra este depasita.



Figura 4-Sfarsitul jocului

2.4 Meniul de pauza

Prin apasarea tastei ESC se poate accesa meniul de pauza, care este compus din urmatoarele butoane: CONTINUE, RESTART si BACK TO MENU.



Figura 5-Meniul de pauza

3.Tehnologii utilizate

C#

MonoBehaviour, our parent class form our scripts

System.Collections belongs to .NET

We assign this Tag, Target, to all objects except the plane

Unity

Cu ajutorul Unity atribuim proprietati fizice jocului nostru,iata cateva cele mai importante:

- Masa, care masoara in Kgr.
- Frecarea, forta cu care un obiect se poate roade de pamant.
- Angular Drag. Friction of the air when in front of a torque force
- Gravitatia

- Vremea

4.Dezvoltarea jocului

4.1.Modelarea jocului

[3]Am inceput crearea jocului prin modelarea hartii. Pentru acest lucru am importat obiectele necesare, cum ar fi copacii, bolovanii, tipul terenului, etc.

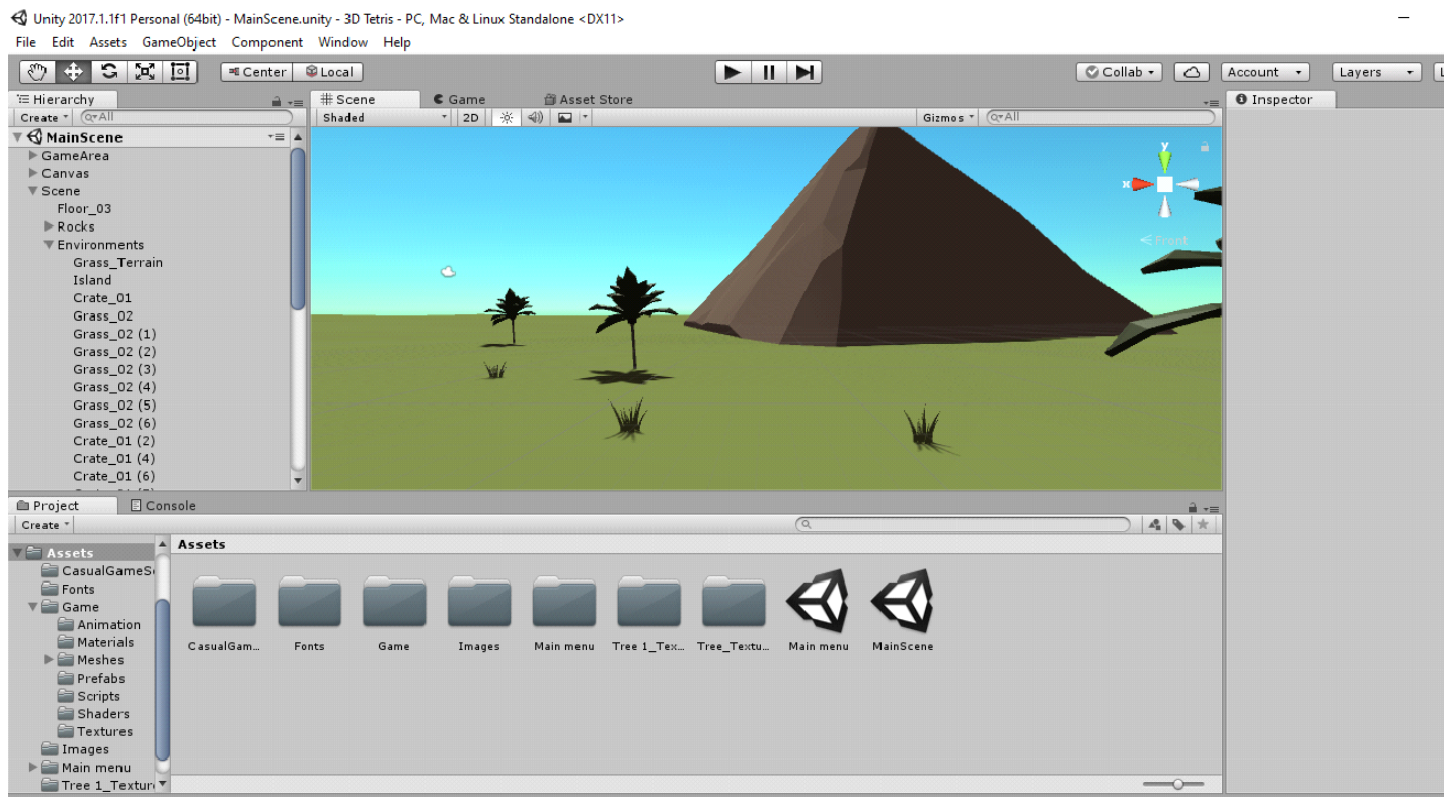


Figura 6-Crearea hartii

4.2. Codul sursa

Functia de start

```
void Start() {
    blocks = General.generateBlockTemplate();

    GameObject camerax;

    camerax = GameObject.FindGameObjectWithTag("MainCamera");
    cameraScript = (CameraMovement)camerax.GetComponent(typeof(CameraMovement));
    canvas = GameObject.Find("Canvas");
    cameraPivot = GameObject.Find("CameraPivot");

    particle = GameObject.Find("Particle");
    particle_system = (ParticleSystem)particle.GetComponent(typeof(ParticleSystem));
    particle_system_start_position = particle.transform.position;
    particle.SetActive(false);

    nextBlockId = Random.Range(0, blocks.Length);
    nextBlockColourId = Random.Range(0, colours.Length);
    addNextBlock();
    Score.init();
}
```

Crearea urmatorului block

```
void addNextBlock() {
    currentBlockObject = createBlock(this.gameObject, blocks[nextBlockId],
nextBlockColourId);
    currentScript = (BlockBase)currentBlockObject.GetComponent(typeof(BlockBase));

    currentScript.x = Random.Range(currentScript.xMin, currentScript.xMax + 1);

    if (!isMovePossible(currentScript.block.block, 0, 0, 1)) {
        if (Random.Range(0, 2) == 1) {
            currentScript.z += 1;
        }
    }
    currentScript.y = General.height;

    currentScript.fixPositionZ();
    currentScript.fixPositionX();
    currentScript.fixPositionY();

    createHintBoxes();

    currentTimeForEachDrop = General.timeForEachDrop;
    timeForNextCheck = currentTimeForEachDrop;
    timeForMovingAni = -1;
}
```

```

    isMoving = true;
    if (currentNextBlockObject != null) {
        Destroy(currentNextBlockObject);
    }

    nextBlockId = Random.Range(0, blocks.Length);
    nextBlockColourId = Random.Range(0, colours.Length);
    currentNextBlockObject = createBlock(this.gameObject, blocks[nextBlockId],
nextBlockColourId);
    currentNextBlockObject.transform.parent = NextBlock.transform;
    currentNextBlockObject.transform.localPosition = new Vector3(0, 0, 0);
}

```

Camera

```

void Update(){

    if (isGameOver) {
        gameOver_center = Vector3.SmoothDamp(gameOver_center,
gameOverText.transform.position + new Vector3(2, 0, -1), ref velocity2, 0.5f);
        transform.position = Vector3.SmoothDamp(transform.position,
camera_gameover_position, ref velocity, 0.5f);
        transform.LookAt (gameOver_center);
        return;
    }

    if (mainscript.GameOver ()) {
        gameOverText.SetActive (true);
        isGameOver = true;
        scoreTextF.SetActive(false);
        scoreTextB.SetActive(false);
        gameOverScoreText.SetActive(true);
        gameOverScoreText.GetComponent<TextMesh>().text =
scoreTextF.GetComponent<TextMesh>().text;
        return;
    }

    if (transform.position.z < 0) {
        front = 1;
    } else {
        front = -1;
    }

    if (!ismovingback && (Input.GetKeyDown (KeyCode.LeftShift) ||
Input.GetKeyDown(KeyCode.RightShift))) {
        ismovingback = true;
        front = -front;
        if (front == 1) {
            targetPosition = camera_start_position;
            scoreTextF.SetActive(true);
            scoreTextB.SetActive(false);
        } else {
            targetPosition = camera_back_position;
            scoreTextB.SetActive(true);
            scoreTextF.SetActive(false);
        }
    }
}

```

```

        if (transform.position == targetPosition) {
            ismovingback = false;
        }

        if (ismovingback) {
            Quaternion QT = Quaternion.Euler(0, 0, 0);
            this.CameraPivot.rotation = QT;
            transform.position = Vector3.SmoothDamp(transform.position,
targetPosition, ref velocity,0.1f);
            transform.LookAt (target_center);
            return;
        }

        if (Input.GetAxis("Mouse X") != 0 || Input.GetAxis("Mouse Y") != 0){
            cameraRotation.x += Input.GetAxis("Mouse X") * MouseSpeed;
            cameraRotation.y += Input.GetAxis("Mouse Y") * MouseSpeed;

            if (cameraRotation.y < -15f)
                cameraRotation.y = -15f;
            else if (cameraRotation.y > 50f)
                cameraRotation.y = 50f;
            if (cameraRotation.x < -40f)
                cameraRotation.x = -40f;
            else if (cameraRotation.x > 40f)
                cameraRotation.x = 40f;

            Quaternion QT = Quaternion.Euler((front)*cameraRotation.y,
-cameraRotation.x, 0);
            this.CameraPivot.rotation =
Quaternion.Lerp(this.CameraPivot.rotation, QT, Time.deltaTime * OrbitDampening);
        }

        transform.LookAt(target_center);
    }
}

```

4.3 Probleme intampinate

Neavand experienta cu acest mediu de dezvoltare a fost un pic dificil la inceput, insa se gasesc multe tutorial si surse de informare pe internet.

A fost dificila in mare parte zona de implementare, mai ales acolo unde era necesara o atentie mai sporita, cum ar fi rotirea si miscarea block-ului.

5.Concluzii

Crearea acestui proiect m-a ajutat sa invat lucruri noi, care pe viitor sigur o sa imi fie de ajutor. Ca un update pentru joc, pe viitor s-ar putea crea mai multe nivele, cu harti noi, trecand in urmatorul nivel pe baza unui scor minim sau alte criterii. De asemenea, se poate implementa disparitia unei linii completata in intregime, astfel incat s-ar putea mari timpul de joc si scorul.

6.Bibliografie

[1] [HYPERLINK](#)

"%5b1%5d%20https://www.youtube.com/watch?v=zc8ac_qUXQY&feature=share" <https://www.skillshare.com/classes/Unity-Game-Tutorial-Tetris-3D/579971593>

[2] <https://github.com/marvpaul/Ultimate-Tetris-3D-Unity>

[3] <https://www.youtube.com/watch?v=TNAKv1dkYyQ>

[4] <http://unity3d.com>

[5] <http://www.unifycommunity.com>