

Goat Game



Szerző:

Szalma Szilárd

Vezető tanár:

Dr.Szántó Zoltán

Szoftver Rendszerek Tervezése

Sapientia EMTE,2019

Tartalomjegyzék:

1.Bevezető.....	3
1.1Előszó.....	3
1.2 Célkitűzések és rövid leírás.....	3
2.Követelmény Specifikáció.....	4
2.1 Rendszer követelmények.....	4
2.2Felhasználói követelmények.....	5
3.Tervezés.....	6
4.Implementálás.....	7
5.Összegzés.....	10
6.Jövőbeli tervek.....	11
7.Bibliográfia.....	11
8.Függelék.....	11

1.Bevezető:

1.1Előszó

A videójáték olyan elektronikus játék, amely a felhasználói felülettel való folyamatos kölcsönhatást foglalja magában, valós idejű vizuális visszajelzést generál a két- vagy háromdimenziós megjelenítő eszközön. Ilyen a TV képernyő vagy számítógépes monitor.

A videójátékok is különböző műfajokba sorolhatóak számos különböző jellemvonás alapján, mint a játékmenet, a célok fajtái stb. A játékok fejlődésével a műfajok is változtak az idők folyamán, ahogy új játéktílusok jöttek létre különböző tartalmakkal. A technikai lehetőségek bővülésével az évek során a videójátékok készítésének költségei is egyre emelkedtek, egyre életszerűbb grafikus látvánnyal, a történetmesélés és a játékmenet komplexitásával egyre kitolva az addig hagyományossá vált műfajok határait.

FPS:(First Person Shooter) belső nézetű lövöldözős játék, az irányított figura szemszögéből játszunk.

TPS:(Third Person Shooter) külső nézetű lövöldözős játék, fedezékből haladva játszunk.

Oldalnézetes akciójáték: klasszikus 2D-s megjelenítés.

1.2 Célkitűzések és rövid leírás

Projektem célja az volt, hogy felelevenítsem régen tanult C# ismereteimet, ezeket továbbfejlesszem, és megismerkedjek a játékfejlesztéssel, a szoftver rendszerek tervezése nevű tantárgyon belül.

A Goat Game nevű 2D-s játék célja, hogy a felhasználó, egy kecske bőrébe bújva minél több ellenséges objektumot elpusztítson, és ezzel pontokat szerezzen.

A játék szintjétől függően az ellenséges objektumok hol lassabban, hol gyorsabban közelítenek, a kecske fel-le mozgásával pedig likvidálni kell őket pontok fejében.

A háttér úgy van megvalósítva, mintha a karakterünk a levegőben lebegne, és az ellenséges objektumok feléje repülnének.

Standard játékelmény nyújtása során elég lett volna úrhajókat behelyezni, azonban a lebegő kecske, és a felé közeledő repülő objektumok nagyobb játékelményt válthatnak ki.

2.Követelmény Specifikáció

2.1 Rendszer követelmények:

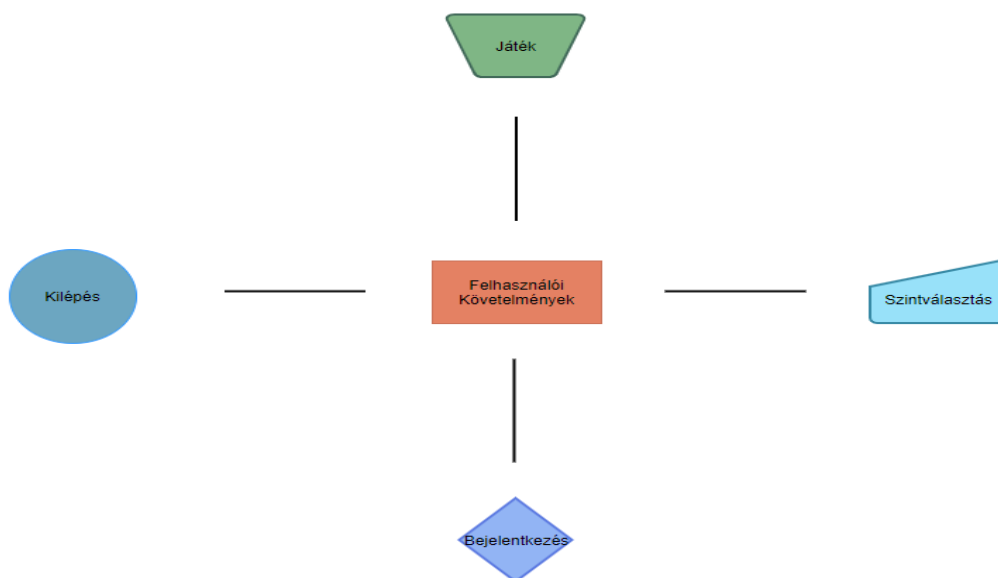
A Goat Game nevű játék Visual Studio programmal lett megalkotva,C# nyelvet használva.

Teház ahhoz,hogy játszani tudjunk a játékkal,szükséges letölteni a visual studio egyik verzióját,lehetőleg a leg újabbat.

- **Processzor:** 1,8 gigahertz(GHz),optimálisan ennél nagyobb;
- **Memória:** 2 gigabájt(GB),optimálisan 8 gigabájt;
- **Tárhely:** 800 megabájt(MB)- 210 gigabájt(GB) a telepített szolgáltatásoktól függően;
- **Operációs rendszer:** Windows7,Windows8,Windows 10;

2.2Felhasználói követelmények:

.

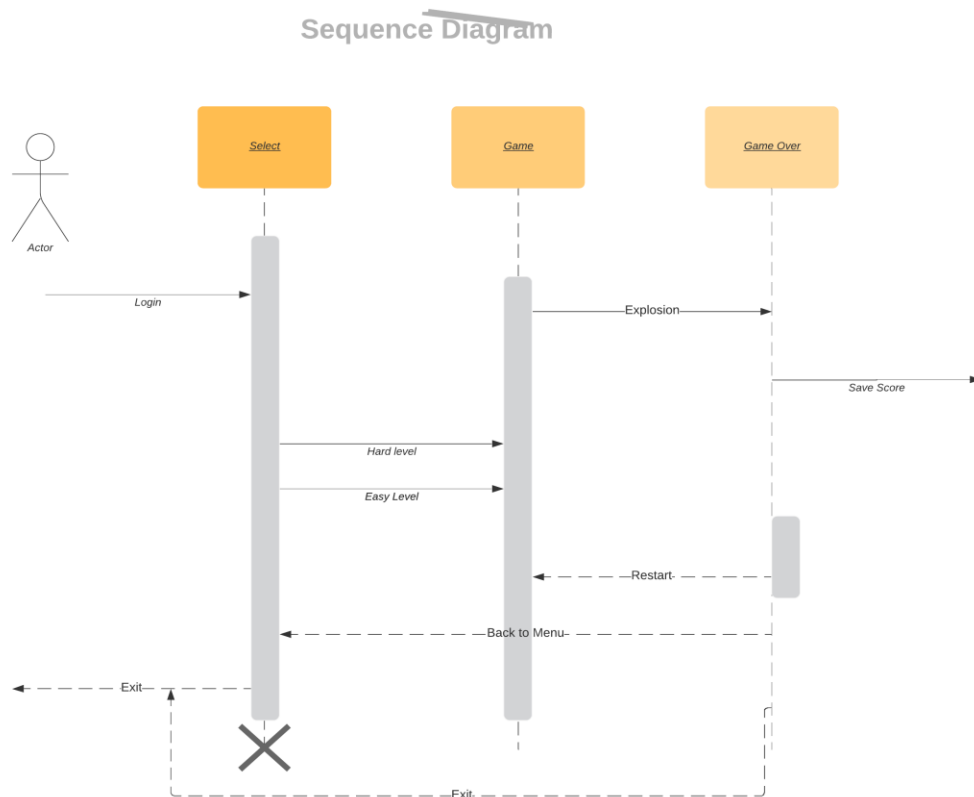


Bejelentkezés:A játékos első belépésre a login ablakban találja magát.Itt kell egy adott felhasználónévvel,illetve jelszóval belépni,amit előzetesen a szolgáltatótól kaphat meg.

Szintválasztás:Bejelentkezés után a szintválasztó select ablakba juthatunk ahol kiválaszthatjuk a nekünk megfelelő nehézséget.Ugyanebben az ablakban találhatóak a játék instrukciói:A kecske a fel-le nyilakkal mozgatható,a Z billentyű lenyomásával pedig lőhet.

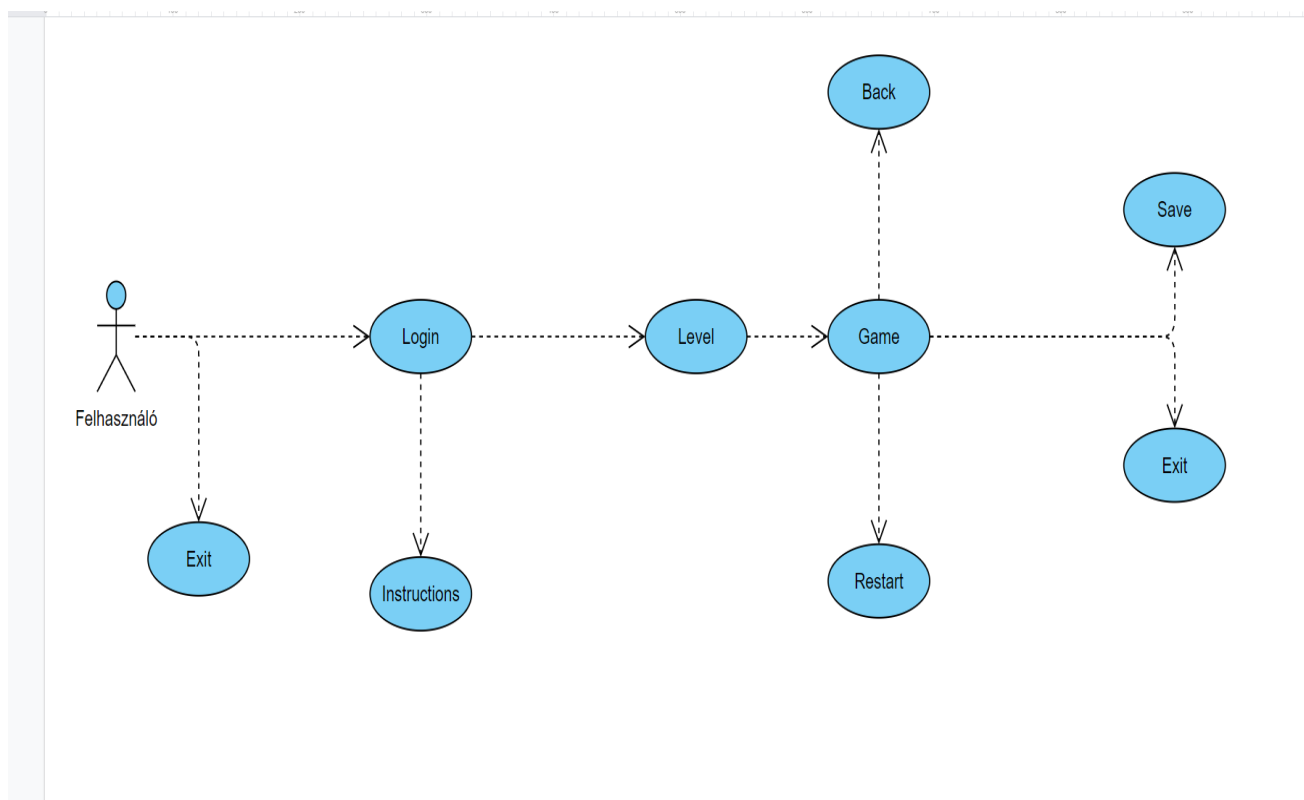
Játék:Az adott nehézség kiválasztása után bejutunk a játék felületre,ami azonnal indul is.A játékban található pontszámláló 1-esével nő,minden elpusztított objektum után.

Kilépés: Ha a karakter ütközik az objektummal,vége a játéknak.A felhasználó az exit paranccsal kiléphet,de ezen kívül,a restart-al előlről kezdhet játszani,a back to menu paranccsal visszaléphet a főmenübe,és a save score paranccsal pedig lementheti eddigi pontszámát.



3.Tervezés

Use Case diagram(1)



A tervezési folyamat elején a felhasználó által irányított karakter egy repülőgép volt, amivel az ellenséges repülőgép objektumokat kellett likvidálni. Háttérnek pedig egy csillagos sötét égboltot, Star Wars-os beütést alkalmazva.

Hamar kiderült, hogy a játék nagyon egyszerű, könnyen megvalósítható, ezen okokból kifolyólag pedig unalmassá válhat. Ennek okán lett a játék központi karaktere egy lebegő kecske, aki a hátsó részéből kilövő golyókkal semmisíti meg az ellenséges objektumokat, amiknek külalakján ismét kicsit változtattunk.

Fontos volt a fejlesztői környezet megválasztása is. A Visual Studio fejlesztői környezet viszonylag egyszerű, és jól lehet dolgozni vele. A környezet kiválasztása után lett eldöntve, hogy a játék C# nyelven lesz megírva.

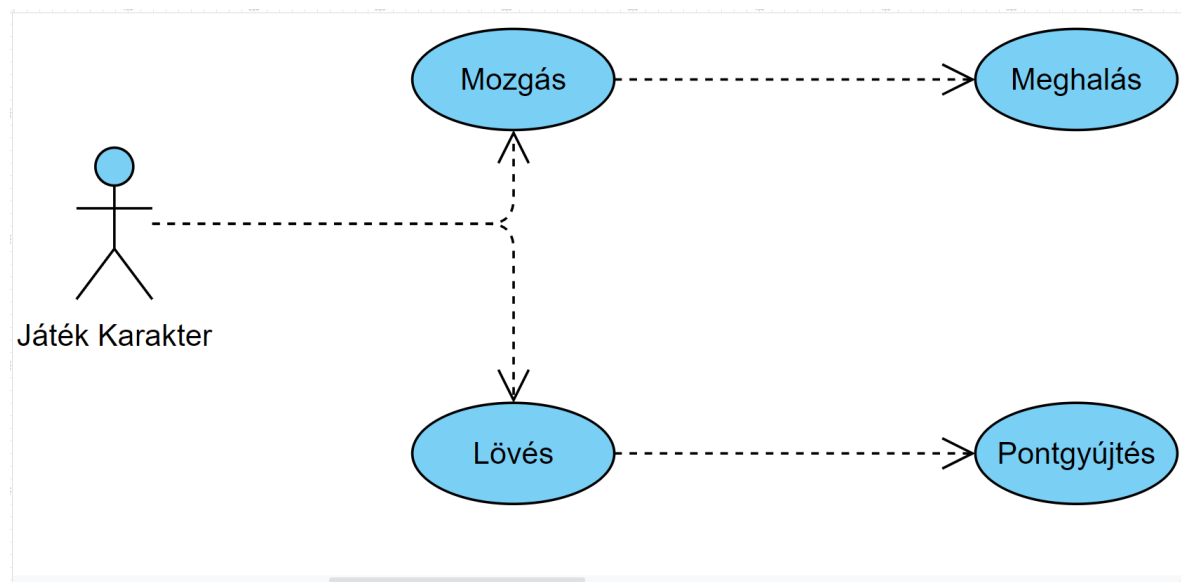
Kezdetben fontos volt egy alapszintű, demo játékot létrehozni, ami csak 1 felületből állt, 1 darab Main ablakkal, ahol a karakterünk lőtte az ellenséges objektumot és a pontszámláló ment felfele.

A szimpá játék megvalósítása után komplikálni kellett a feladatot, a játékhoz csatolva bejelentkezést, exit gombokat, lehetőséget az újrajátszásra, nehézségi szinteket felállítani és lehetőséget adni a pontszám lementésére.

A játékban egy egyszerű fel-le mozgatható karaktert láthatunk,ami 2D-s vizualizációt nyújt.Sok ötletelés után,megpróbáltuk használni a Unity fejlesztői környezetet,hogy a felhasználónak 3D-s élményt nyújtsunk,de ez az ötlet hamar el lett vetve,bonyolultsága miatt.

A játék egyszerűnek,könnyen elsajátíthatónak készült,ezért is ilyen egyszerű a fent látható Use Case diagram.Igy tehát a felhasználó csak a legegyszerűbb funkciókkal rendelkezik.Belépés a játék indításához,instrukciók a játék elsajátításához,szintválasztás,valamit játékújraindítás és a pontszám lementése.

Use Case diagram(2)



A fenti Use Case diagram a játékban szereplő karakter funkcióit mutatja be,amik a lehető legegyszerűbbek.

A karakter mozogni tud fel,illetve le és golyókat tud lőni,amivel minden egyes megölt objektum után pontot kap.Ha a karakter mozgás során ütközik az ellenséges objektummal,a karakter meghal és a játék véget ér.

4.Implementálás

Az implementáláshoz,az előbb említett Visual Studio fejlesztői környezetet használtuk,amivel a kódolást és a dizájnt egyaránt el lehetett végezni.

Login

```
private void Login1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Szalma
Szilard\Documents\Data.mdf;Integrated Security=True;Connect Timeout=30");

    SqlDataAdapter sda = new SqlDataAdapter("select count(*) from login where username
='" + write1.Text + "' and password='" + write2.Text + "'", conn);

    DataTable dt = new DataTable();

    sda.Fill(dt);

    if (dt.Rows[0][0].ToString() == "1")
    {
        this.Hide();

        select ff = new select();

        ff.Show ();
    }
    else
    {
        MessageBox.Show("please enter correct username and password", "alert",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

A fenti kódrészlet összeköttetést nyújt egy Sql adatbázissal, ami ugyancsak a Visual Studioból érhető el. Ott állíthatja be a szolgáltató a bejelentkezéshez szükséges adatokat

Select


```
private void SinglePlayer_Click(object sender, EventArgs e)
```

```
{
```

```
    this.Hide();
```

```
    Main ss = new Main();
```

```
    ss.Show();
```

```
}
```

```
private void TwoPlayer_Click(object sender, EventArgs e)
```

```
{
```

```
    this.Hide();
```

```
    Form2 oo = new Form2();
```

```
    oo.Show();
```

```
}
```

```
private void InstructionsGame_Click(object sender, EventArgs e)
```

```
{
```

```
    this.Hide();
```

```
    Instructions pp = new Instructions();
```

```
    pp.Show();
```

A fenti kódrészlet egyszerűen csak ablakok váltakozásához szükséges. Ez a select ablak ahol kiválaszthatjuk a szintet, illetve az instrukciókat egyaránt.

Save/Open File

```
private void OpenF_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        label1.Text = openFileDialog1.FileName;
        textBox1.Text = File.ReadAllText(label1.Text);
    }
}

private void SaveF_Click(object sender, EventArgs e)
{
    if (saveFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        File.WriteAllText(saveFileDialog1.FileName, textBox1.Text);
    }
}
```

Algoritmus a pontszám lementéséhez

5.Összegzés

Összegezve az eddigieket,sikerült egy egyszerű,de annál érdekesebb és viccesebb játékot létrehozni,ami mosolyt csalhat az emberek arcára.Sikerült feleleveníteni a C#-ban tanult ismereteket és bővíteni azokat,ugyanakkor sikerült elmélyíteni és jobban kitapasztalni a Visual Studio programot.

6.Jövőbeli tervek

- Two Player játékmód elkészítése
- Nehézségi szintek behozatala
- Boss behozatala a játékba
- Életerő implementálás
- Android app készítése
- Megtanulni használni a GitHub-ot

7.Bibliográfia

- 1.<https://www.youtube.com/watch?v=GhQdIIFyIQ8&vl=en>
- 2.<https://stackoverflow.com/>
- 3.<https://www.youtube.com/watch?v=tcmmCcMs8yU&t=532s>

8.Függelék

A diagramok a <https://online.visual-paradigm.com/diagrams/solutions/free-use-case-diagram-tool/> és <https://www.lucidchart.com/documents#/dashboard> online felületen készültek.

