**Báthory István Elméleti Líceum**

**Kolozsvár**

**Matematika-informatika**

**intenzív informatika**

**Pacman**

**Felkészítő tanár: Készítette:**

**Szász Tünde Kaszta Emmánuel 12.B**

**2023**

**Tartalomjegyzék**

1. A feladat megfogalmazása 3. oldal

2. Hard és software elemek 5. oldal

3. Programozási dokumentáció 6. oldal

4. Felhasználói dokumentáció 15. oldal

5. Továbbfejlesztési lehetőségek 20. oldal

1. **A feladat megfogalmazása**

A Pacman egy klasszikus videojáték, amelyben a játékosnak egy sárga kör alakú karaktert kell irányítania, amelyet Pacmannak neveznek. A játék célja, hogy a Pacman átjárjon egy labirintusban, gyűjtsön össze minél több pontot, és elkerülje a szellemeket.



A labirintusban különböző pontok találhatók, amelyeket a Pacman felvehet. A pontokat általában pöttyként vagy gyümölcsként ábrázolják. Amikor a Pacman felvesz egy pontot, a pontszáma növekszik. Az a cél, hogy a játékos minél több pontot gyűjtsön össze.

Azonban a Pacmannak óvakodnia kell a szellemektől, amelyek állandó mozgásban vannak a labirintusban. Ha a Pacman és egy szellem találkozik, a Pacman elveszti az életét, és a játékos pontszáma csökken. A játékosnak meg kell próbálnia elkerülni a szellemeket és ügyesen navigálni a labirintusban.

A játék célja az összes pont összegyűjtése a labirintusban anélkül, hogy a Pacmant elkapnák a szellemek. Ha a játékos sikeresen összegyűjti az összes pontot, akkor nyert.



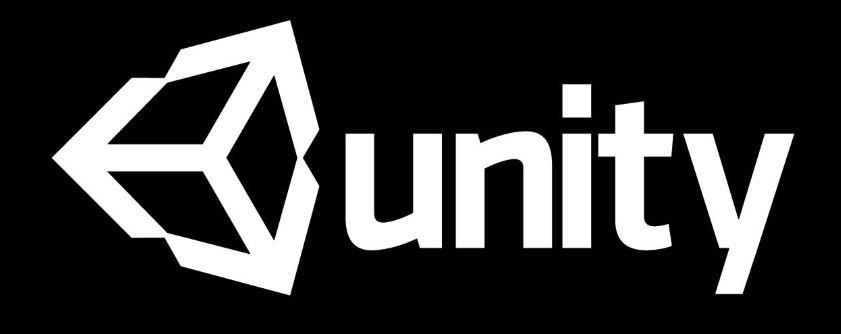
1. **Hard és software elemek**

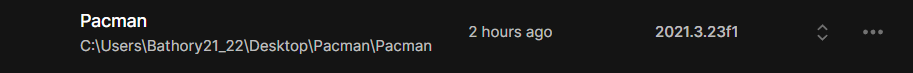
**Hardverelemek:**

* Processzor: Legalább 1 GHz-es sebességű processzor.
* Memória (RAM): Legalább 1 GB RAM.
* Grafikus kártya: Legalább 128 MB dedikált grafikus memóriával rendelkező grafikus kártya.
* Kijelző: Legalább 800x600 képpont felbontású kijelző.

**Szoftverelemek:**

* Operációs rendszer: Windows 7 vagy újabb, macOS, vagy Linux rendszer.
* CPU: 64x
* Játék Futtató szoftver: **Unity 2021.3.23f1** verzió vagy újabb

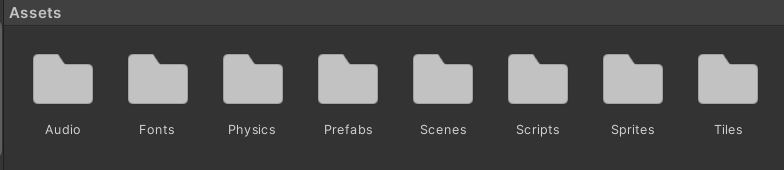




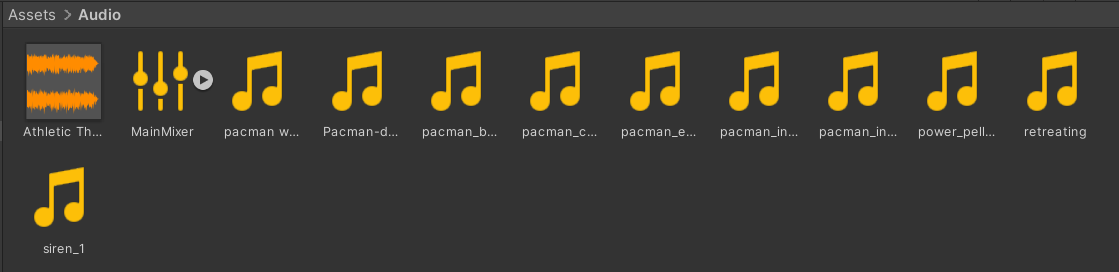
**3. Programozási dokumentáció**

A program a **C# (C-Sharp)** programozási nyelvben volt megírva, **Unity**-t használva, tehát objektum orientáltan. A játék a **Packages** alapértelmezett, és az **Assets** mappát használja.

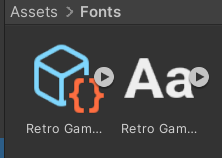
Az **Assets** mappa **8 csomagot** tartalmaz:



* Az **Audio** csomag a zenéket, soundeffectket tartalmazza, a játék során játszódnak le



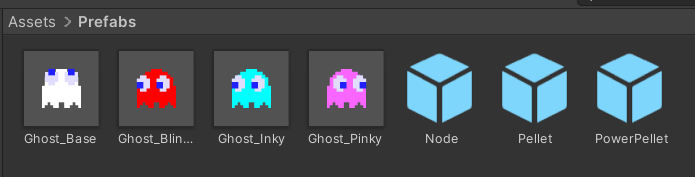
* A **Fonts** csomag a betűstílust tartalmazza, ami megjelenik az UI-on a játék során



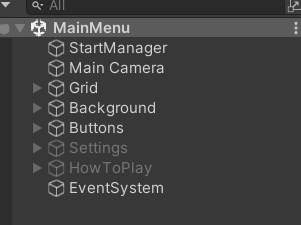
* A **Physics** csomag a karaktereket-t befojásoló fizikai szabályokat tartalmazzák

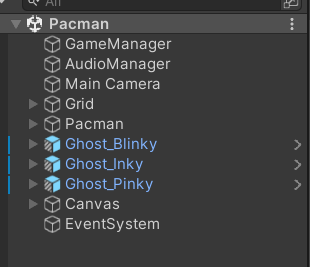


* A **Prefabs** (rövidítve Prefabrications) az Unity játékfejlesztő motorban egy olyan funkció, amely lehetővé teszi előre definiált objektumok létrehozását és újrahasznosítását. A Prefab egy sablon vagy minta, amelyből könnyen létrehozhatók az azonos tulajdonságokkal rendelkező objektumok. Ebben az esetben a **Ghost** karakterek, **Pellet** és **PowerPellet** sablonja

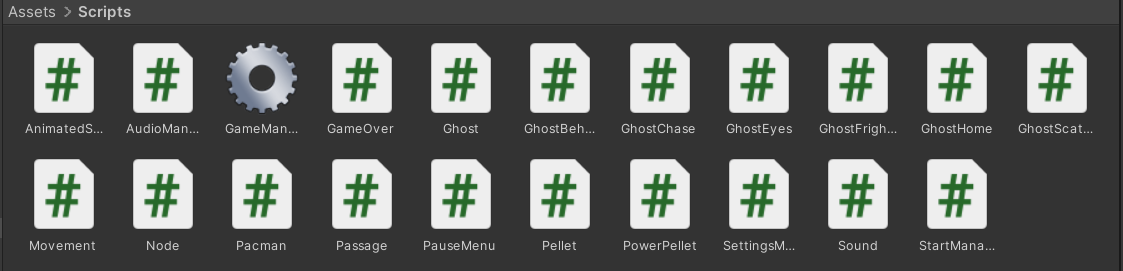


* A **Scenes** csomag tartalmazza a játékban használt 2 felületet, a MainMenu-t és a Game ablakot (“Pacman”)-t

****



* A **Scripts** csomag tartalmazza a játék során felhasznált összes C# osztályt

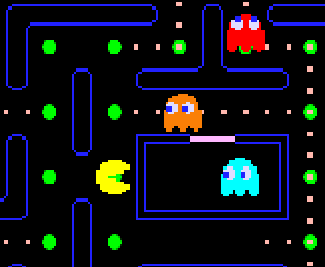


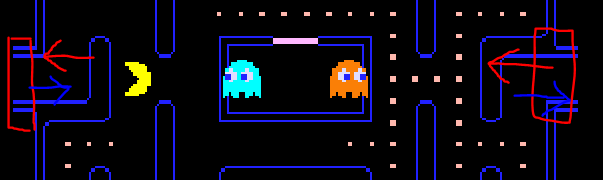
1. Az **AudioManager** és **Sound** osztályok kezelik a zenék indítását, megállítását és egyéb dolgokat. Ezeket a scripteket a “AudioManager” GameObject használja.
2. A **Node, Passage, Pellet** és **PowerPellet** osztály a pályán lévő elemek rendszerezésére van.

A **Node** *(az 1.ábrán látható, zöld pont)* segítségével tudjuk hol, milyen irányba fordulhatnak az entity-k;

A **Passage** osztály átteleportálja az entity-ket a pályán megjelenő két kivezető cső ellentétes oldalára; *(Teleportálás vázolva a 2.ábrán)*

A **Pellet és PowerPellet** osztály viszont számon tartja a megevett bogyók számát és ha Pacman megeheti-e a Ghostokat;





*2. ábra* *1.ábra*

1. A **Ghost, GhostBehavior, GhostChase, GhostEyes, GhostFrightened, GhostHome, GhostScatter** a szellemek állapotát kezelik.

A **Ghost** és a **GhostEyes** kezeli a szellemek animációját és kinézetét;

A **GhostsBehavior** összesíti a szellemek állapotváltozásait;

A **GhostChase** osztály arra a módra van, amikor a szellemek elkezdik követni Pacmant. Ez a követés egy kereső-algoritmust használ; *(3.ábra)*

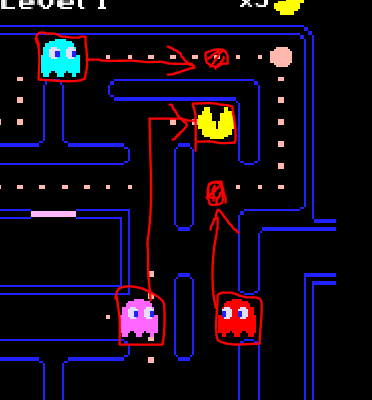
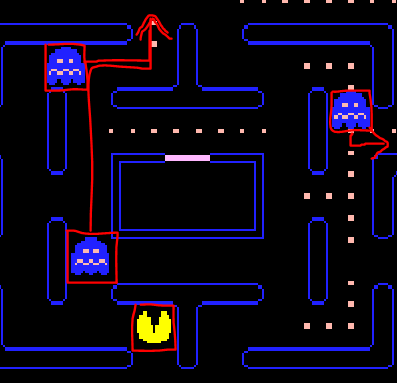
A **GhostScatter** arra a szellem-állapotra van, amikor a szellemek felhagynak kis időre Pacman követésével, és elindulnak kiszámíthatatlan irányokba egy adott időre;

A **GhostFrightened** osztály a **GhostChase** ellentéte, a szellemek menekülnek Pacman elől amíg PowerPellet hatása le nem jár; *(4.ábra)*

Legvégül a **GhostHome** kezeli a szellemek kezdeti pozícióját amikor a szellemek otthon, a kis kockákban vannak vagy miután megette őket Pacman, vagy a játék kezdete elején; *(5.ábra)*

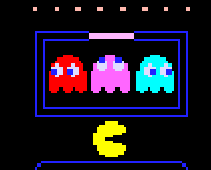
A kereső algoritmus amit a szellemek használnak, megnézi minden Node-nál a szomszédai közül melyik van közelebb/távolabb Pacman pozíciójához, és legvégül arra mennek tovább;

Szellemként a keresett pozíció változik. **Ghost\_Blinky** pacman pozícióját követi, viszont **Ghost\_Inky** pacman elotti, (+(Pacman.direction + 4.5)), **Ghost\_Pinky** pedig pacman háta mögötti (-(Pacman.direction + 4.5)) pozíciót veszi célba. *(3.ábra)*



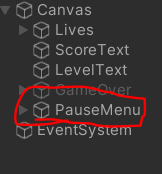
*4.ábra*

*3.ábra*



*5.ábra*

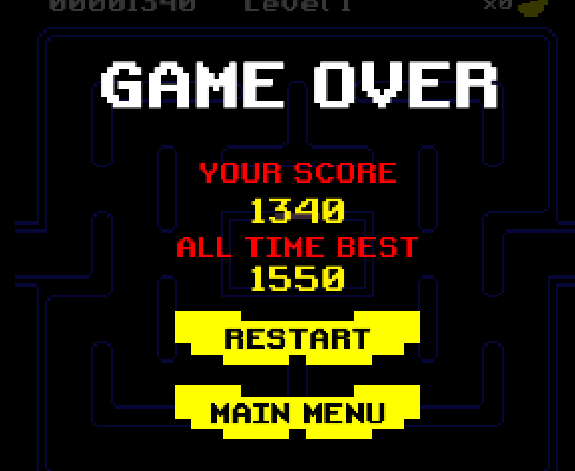
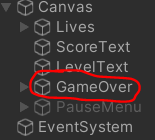
1. Az **AnimatedSprite** és **Movement** osztályok segítségével mozognak az entity-k, módosíthatjuk Pacman pozícióját, majd vetítjük ki a képernyőre, meganimálva a mozgásukat
2. A **PauseMenu** osztály megjeleníti a PauseMenu UI-t. Ennek segítségével szünetelhetjük a játékot menet közben, akár ki is léphetünk, vagy scene-t válthatunk „Pacman”-ről „MainMenu”-re. 3 gombot tartalmaz a PauseMenuUI, egyik „RESUME”, másik „MAIN MENU” és legutolsó „EXIT GAME”.*(6,7. ábra)*



*6.ábra*

1. A **GameOver** osztály megjeleníti a GameOver UI ablakot. Itt a gombokért, a GameOver sound lejátszásáért és a map feltakarításáért felelős ez a script.

A képernyőre kiírja a mostani játék alatt összegyűjtött pontokat és a eddigi legnagyobb pontszámot, amit elért a játékos. *(8,9.ábra)*



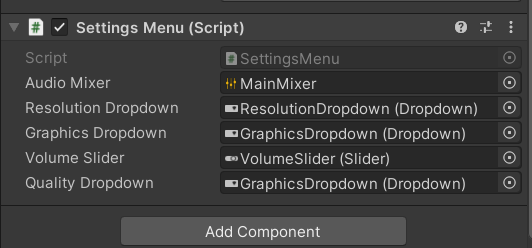
*9.ábra*

*8.ábra*

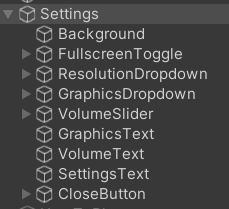
1. A **Pacman** osztály felelős pacman navigációs irányításának frissítésére. Ez akkor fog megtörténni, mikor a program érzékel egy KeyInputot. Ha a megadott WASD vagy ↑ ↓ → ← gombokat le vannak nyomva akkor frissül pacman irányvektora ennek függvényében.

Ez az osztály felelős pacman meghalási animációjának elindításáért, és pacman pozíciójának, public void ResetState() - ért.

1. A **SettingsManager** osztály végez mindent ami a Settings UI-al kapcsolatos. A gombok megjelenítése, ezek működése, a játék frissítése a beállítások függvényében. Legfontosabb dolga hogy elmenti a felhasználó által módosított beállításokat, majd ezeket a játék összes ezentúli betöltése után felhasznál. *(10, 11.ábra)*

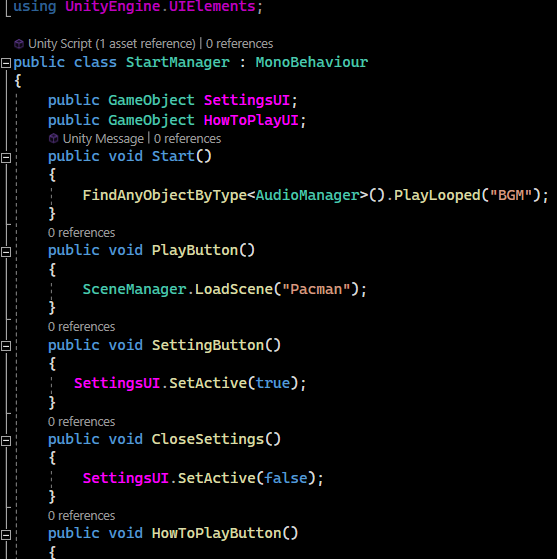


*10.ábra*



*11.ábra*

1. Az első scene-t, a MainMenu-t ami teljesen összefűzi az a **StartManager** osztály. Itt a különböző UI-ok megjelenítésére szükséges funkciók vannak mint például: public void PlayButton(), public void SettingButton(), public void CloseSettings(), public void HowToPlayButton(), és egyebek. *(12.ábra)*



*12.ábra*

1. Legutolsó sorban, a másik scene-t összekötő osztálya a **GameManager**

Ez az osztály frissíti, hívja meg a többi osztály függvényeit amelyeket összerakva egy összeálló, kohézív játékfelületet kapunk*.(13.ábra)*

Főbb függvényei amelyet felhasznál az a: public void NewGame(), private void NewRound(), private void ResetState(), private void GameOver(), public void MainMenu() és sok egyébb. Olyan függvényeket is implementál, amelyek minden új körben megváltoztatják a szellemek színét egy random 3 színre a 4 alap szín közül (lásd: public void getRandomColors()).

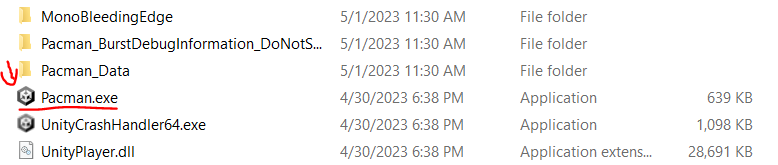
Ebben a osztályban változtatja a játék nehézségét minden kör lejátszása után, de viszont ha egy új játék kezdődik, visszaállít mindent az alap kezdőértékre.



*13.ábra*

1. **Felhasználói dokumentáció**

A játékot a Pacman.exe állomány segítségével lehet elindítani.



Ezután a játék elindul. Az első scene amit lerenderel a játék az a **MainMenu scene** lesz. Ezen az UI felületen **4 gomb** található.



Ahogy betölt a **MainMenu**, elindul a vele párosított „BGM” háttérzene.

Az első interaktálható gomb az a **START GAME** gomb. Ezt megnyomva a játék be fogja tölteni a 2. scene-t, a **Pacman scene**-t. Itt a **GameManager** helyreállítja az entity-ket, a kezdő zenét elindítja, majd kezd egy public void NewGame() -et.

A második gomb az a **SETTINGS** gomb. Ezt megnyomva a **StartManager** osztályunk betölti a **SettingsUI**-t.



Ezen a felületen újabb interaktálható gombok jelennek meg. Ezek segítségével módosítható:

* Játék ablakának mérete (első gomb, ebben az esetben a képernyő méret beállítása: 1600 x 900)
* Játék teljes képernyő beállítási gombja (2. gomb, a „FULLSCREEN” szöveg előtt, a pacman icon jelzi ha be van kapcsolva)
* Játék felbontása (3. gomb, a „GRAPHICS” szöveg után, ebben az esetben „VERY HIGH” felbontás)
* Játék zenéjének hangereje (pacman pozíciója jelzi a hangerő magasságát)

Ezeken az ablakon végrehajtott módosítások elmentődnek, amint a felhasználó kilép a SettingsUI-ból, a **PIROS X** – re kattintva a bal felső sarokban.

A harmadik interaktálható gomb a **MainMenu**-ben a **HOW TO PLAY** gomb.

Erre rákattintva bejön a **HowToPlayUI** ahol megjeleníti a játék a használati utasítást. Ez egy interaktálható Panel, ahol lehet lefelé görgetni/egérrel scrollolni.









Ebből az UI-ból a PIROS X használatával lehet kilépni, visszakerülve a MainMenu-be.

Az legalsó gombbal a **MainMenu**-ben, az **EXIT**-el a játékos bezárhatja a játékot.

Ha a **MainMenu**-ben a felhasználó megnyomja az első gombot, a **START GAME**-t akkor a 2. scene fog betölteni. Itt fog lefolyni a játék menete.

A Játek menete közben lehetőség van bármikor megállítani a játékot az **ESC** billentyűzet gombbal. Ilyenkor megjelenik a **PauseUI** (lásd 6.ábra)

A „**RESUME**”-ra kattintva, vagy az **ESC** billentyűzet gomb megnyomása után folytatódik a játék. A „**MAIN** **MENU**” gombra kattintva a játék átvált a **MainMenu** scene-re. Az „**EXIT GAME**” gomb pedig lezárja az alkalmazást.

Legvégül, mivel minden pálya kivitele után a játék nehezedik, egy idő után, miután a játékos meghal, megjelenik a **GameOverUI**-t (lásd 8.ábra)

Itt 2 lehetősége van a játékosnak, visszaváltani a **MainMenu** scene-re a „**MAIN** **MENU**” gombbal, vagy újrakezdeni a játékot a „**RESTART**” gombbal.

1. **Fejlesztési lehetőségek**

A játékban sok fejlesztésnek való hely van:

* Több pálya hozzáadása a játékhoz
* Nehézségi fokozatok beállítása
* A menu kijelzőt dinamikusabbá tenni
* Több szellem variációt, esetleg a felhasználó által hozzáadott ellenségeket
* A felhasználó által készített pályák mentése és játszhatósága