**Fun Run Clone**

**(Proiect SPRC)**

Coordonator, Student,

Prof. Cristian Niculita Luca Sergiu

ANUL IV CTI

GRUPA 22C32 A

AN 2023-2024

Cuprins

[Introducere 3](#_Toc168515293)

[Ce este un game engine? 3](#_Toc168515294)

[Cele mai populare game engine-uri: 3](#_Toc168515295)

[De ce am ales Godot 4? 4](#_Toc168515296)

[Manualul jocului 5](#_Toc168515297)

[Lansarea Serverului 5](#_Toc168515298)

[Lansarea Aplicatiei 6](#_Toc168515299)

[Jucatul nivelului 8](#_Toc168515300)

[Resursele jocului 10](#_Toc168515301)

[Tehnologiile folosite 12](#_Toc168515302)

[Git: 12](#_Toc168515303)

[GitHub: 12](#_Toc168515304)

[C# (C Sharp): 12](#_Toc168515305)

[Godot 4: 12](#_Toc168515306)

[Vizualizarea codului sursa 13](#_Toc168515307)

[Componente c# folosite 15](#_Toc168515308)

[Finish Line.cs 15](#_Toc168515309)

[GameManager.cs 16](#_Toc168515310)

[MultiplayerController.cs 16](#_Toc168515311)

[Player.cs 17](#_Toc168515312)

[PlayerInfo.cs 18](#_Toc168515313)

[PowerUp.cs 18](#_Toc168515314)

[SceneManager.cs 19](#_Toc168515315)

[Arhitectura si Structura sistemului 20](#_Toc168515316)

[Bibliografie resurse si librarii 21](#_Toc168515317)

# Introducere

## Ce este un game engine?

Un game engine este ca o cutie de instrumente virtuală pentru dezvoltatorii de jocuri. În loc să construiască totul de la zero, un game engine oferă componente pre-construite și funcționalități esențiale, cum ar fi:

* **Randare grafică:** Afișarea imaginilor și animațiilor pe ecran.
* **Fizică:** Simularea mișcării și interacțiunilor obiectelor.
* **Sunet:** Gestionarea efectelor sonore și a muzicii.
* **Input:** Citirea comenzilor de la jucător (tastatură, mouse, controller).
* **Scripting:** Posibilitatea de a adăuga logică și comportament personalizat jocului.

Folosirea unui game engine accelerează dezvoltarea și permite creatorilor să se concentreze pe aspectele unice ale jocului lor.

## Cele mai populare game engine-uri:

Există o varietate de game engine-uri disponibile, fiecare cu propriile puncte forte și specializări. Printre cele mai populare se numără:

* **Unity:** Versatil și ușor de învățat, excelent pentru jocuri 2D și 3D de toate genurile.
* **Unreal Engine:** Cunoscut pentru grafica de înaltă calitate, ideal pentru jocuri AAA și simulări realiste.
* **Godot:** Open-source și gratuit, cu o comunitate activă și în creștere, potrivit pentru proiecte independente și experimentale.
* **GameMaker Studio 2:** Axat pe jocuri 2D, cu un limbaj de scripting vizual ușor de utilizat.

## De ce am ales Godot 4?

Godot 4 a fost ales pentru acest proiect "fun run clone" din mai multe motive:

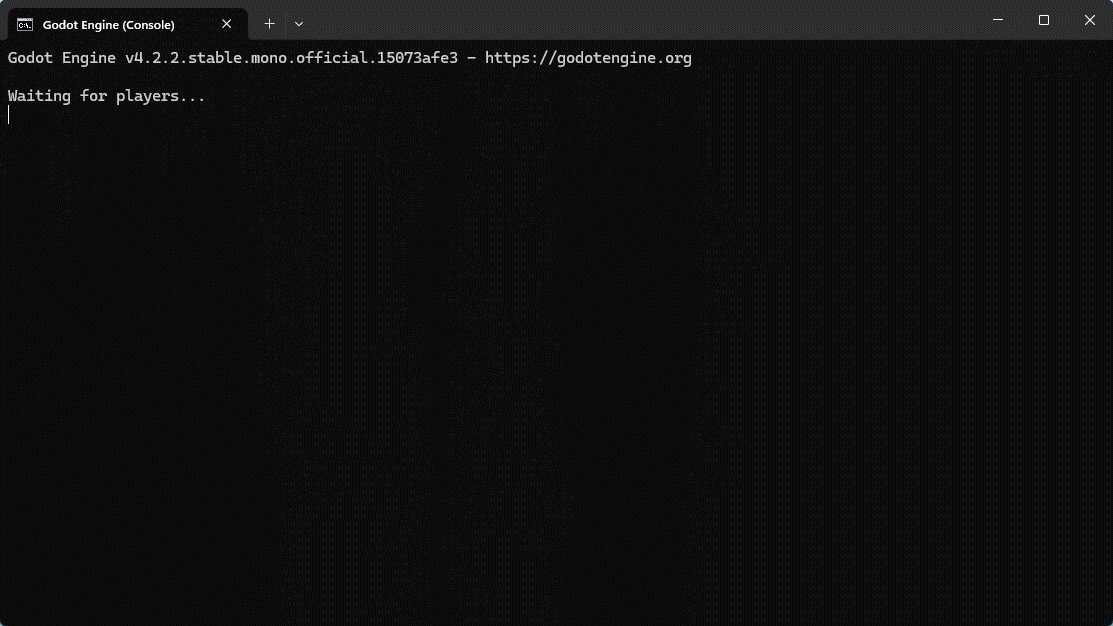
* **Open-source și gratuit:** Elimină barierele financiare și permite flexibilitate totală în dezvoltare.
* **Ușurință în utilizare:** Interfața intuitivă și sistemul de scene simplifică procesul de creare a jocului.
* **Performanță:** Motorul 2D puternic și optimizat poate gestiona cu ușurință grafica și fizica necesare.
* **Comunitate:** O comunitate activă și numeroase resurse online oferă suport și inspirație.
* **Portabilitate:** Jocurile create în Godot pot fi exportate pe o gamă largă de platforme (PC, mobil, web).

# Manualul jocului

În Fun Run Clone, tot ce ai de făcut este să alergi de la stanga la dreapta colectând power-up-ui și să sari peste obstacole încercând sa ajungi primul la linia de finish pentru a câstiga. Pare simplu, nu-i așa?

Succes!

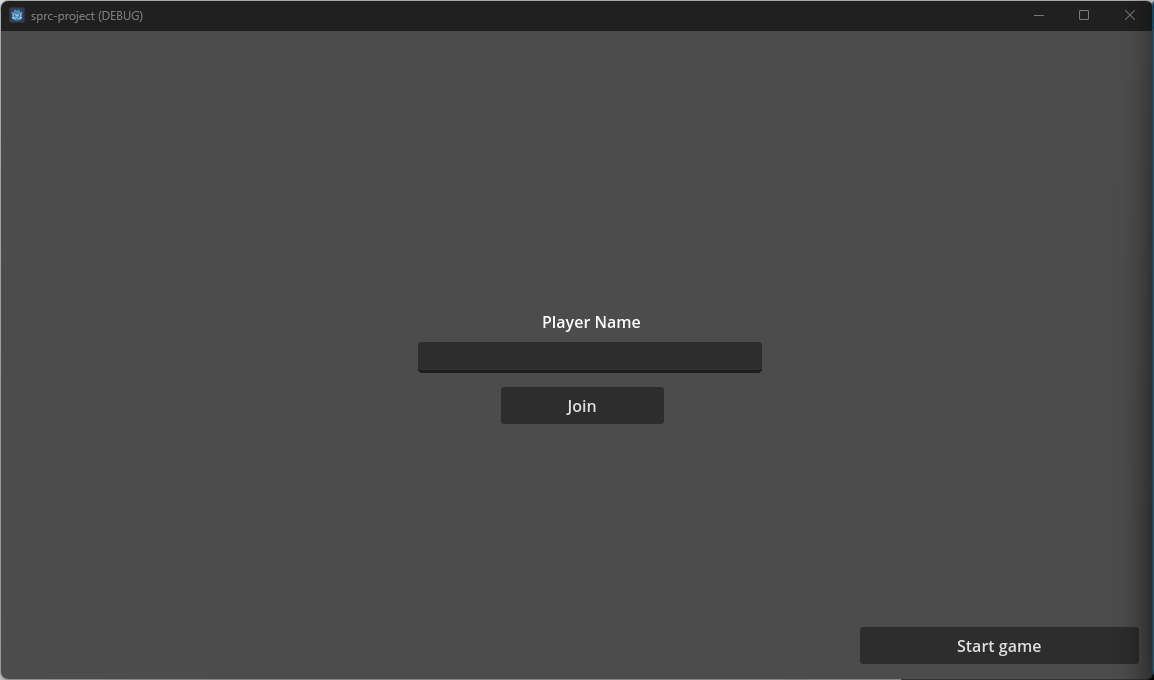
## Lansarea Serverului



Serverul poate fi lansat folosind fișierul Start\_Server.bat din subfolderul server. Acesta va rula executabilul pentru serverul dedicat odată cu alte comenzi pentru a se asigura că nu este afișată și interfata jocului ci doar consola serverului.

Odată lansat serverul, jucătorii se pot conecta la serverul dedicat și să joace.

## Lansarea Aplicației



Odata lansat jocul, jucătorul este întâmpinat de ecranul „Acasă” ce conține un câmp pentru introducerea numelui, butonul de Join (pentru a se conecta la server) și butonul „START GAME”.

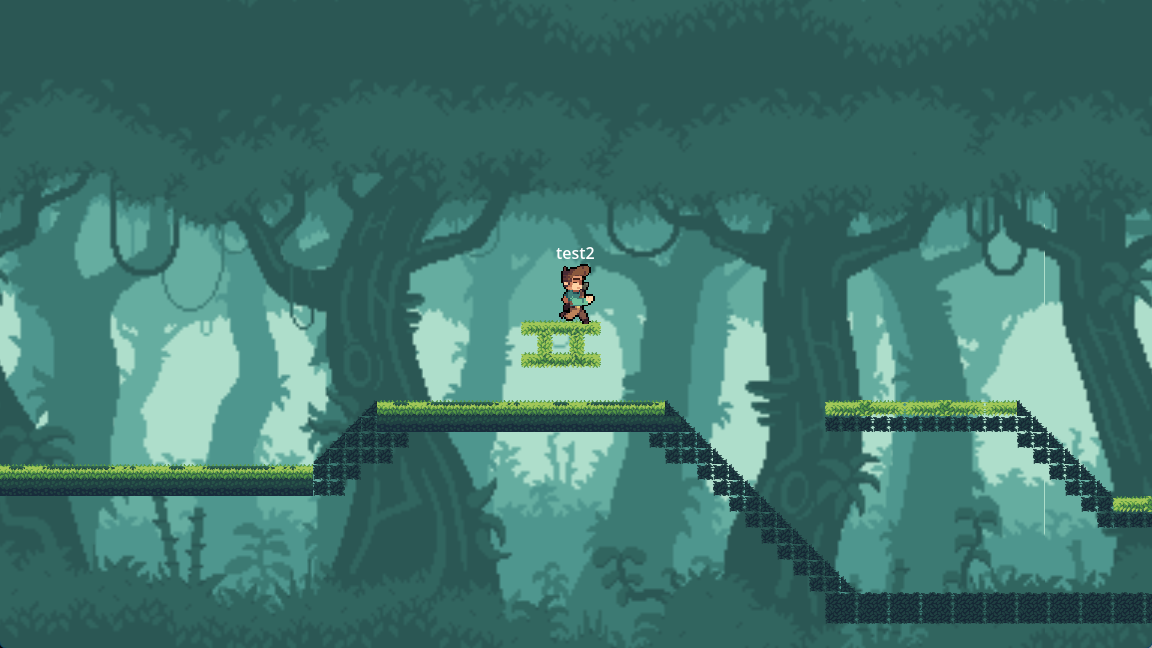
Odata apăsat butonul „START GAME” jocul va aștepta pană când încă un alt jucător se va conecta la server pentru a putea începe jocul.  


Odata ce al doilea jucător s-a conectat jocul va începe instant.

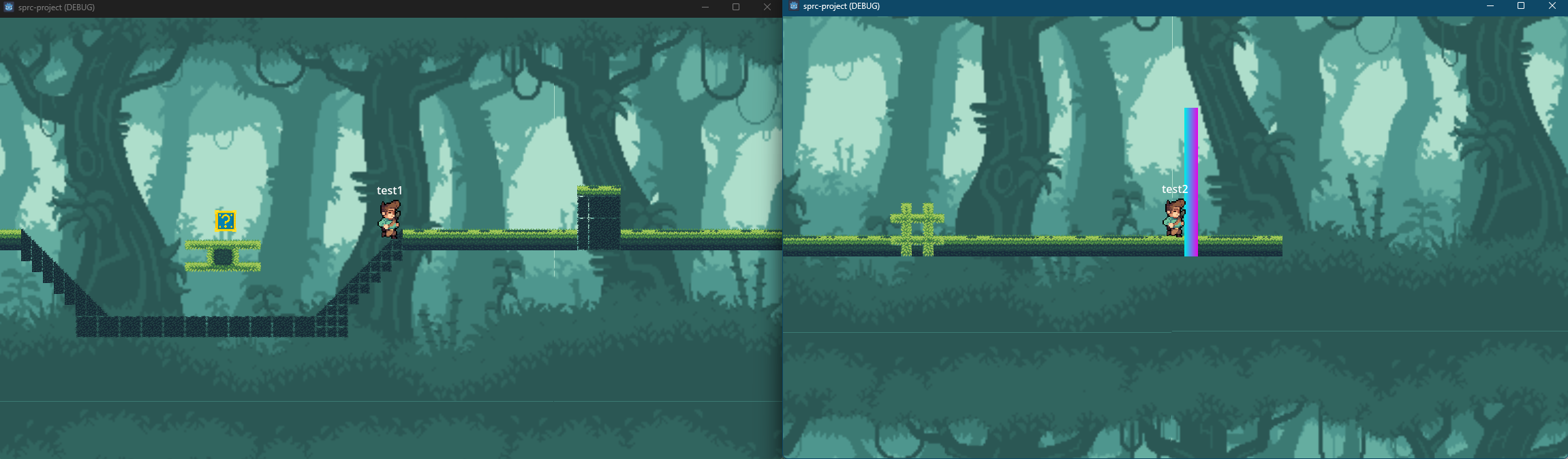
## Jucatul nivelului



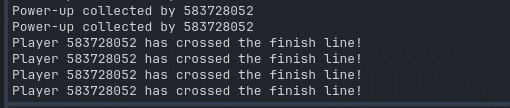
Fiecare nivel poate avea un tip diferit de platformă si așezare a power-up urilor.



Odată ce un jucător trece peste un power-up acesta v-a dispărea și nu va mai fi disponibil.



În momentul în care un jucător ajunge la finalul hării, la linia de finish acesta va alerta serverul că toți jucătorii ar trebui opriți doarece unul a câștigat.

Câștigătorul este anunțat în consolă printr-un log:  


# Resursele jocului

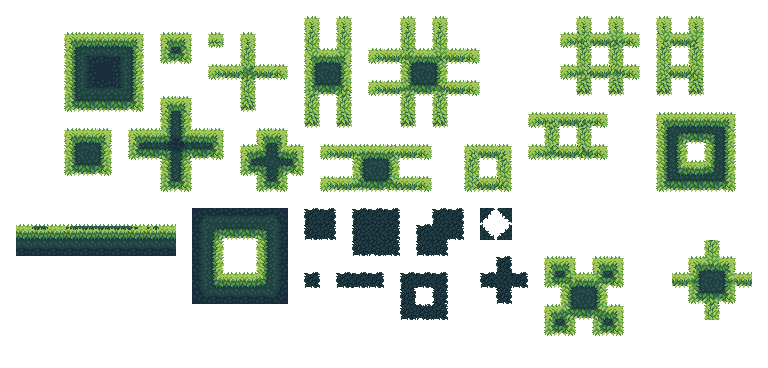
Pentru resursele jocului am folosit un set de indie game dev procurat de pe itch.io

De pe acest website am descărcat și folosit:

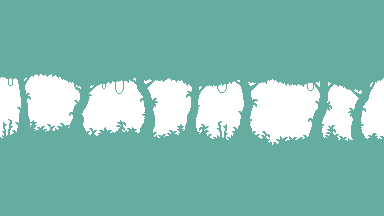
* Caracterul + animațiile acestuia:

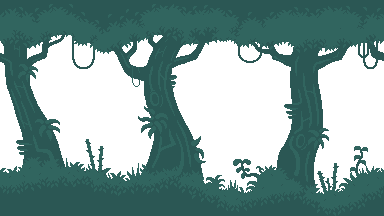


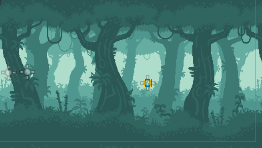
* Tile set ul pentru hartă:



* Background ul cu parallax:







# Tehnologiile folosite

Git: Git este un sistem de control al versiunilor (VCS) distribuit, esențial în dezvoltarea software modernă. Acesta permite urmărirea modificărilor aduse codului sursă, colaborarea între dezvoltatori și revenirea la versiuni anterioare în caz de erori. În proiectul nostru, Git a fost folosit pentru a gestiona și organiza eficient codul sursă al jocului, facilitând colaborarea între membrii echipei și asigurând o istorie clară a dezvoltării.

GitHub: GitHub este o platformă de găzduire pentru proiecte software care utilizează Git. Pe lângă funcționalitățile de bază ale Git, GitHub oferă și instrumente suplimentare pentru colaborare, cum ar fi urmărirea problemelor, revizuirea codului și gestionarea proiectelor. În cadrul proiectului nostru, GitHub a servit ca depozit central pentru codul sursă, permițând accesul facil al tuturor membrilor echipei și facilitând colaborarea la distanță.

C# (C Sharp):C# este un limbaj de programare modern, orientat pe obiect, dezvoltat de Microsoft. Acesta este cunoscut pentru sintaxa sa clară, versatilitatea și performanța ridicată. În proiectul nostru, C# a fost ales ca limbaj principal de programare datorită suportului său nativ în Godot 4 și a avantajelor sale în dezvoltarea jocurilor, cum ar fi gestionarea ușoară a obiectelor, colectarea automată a gunoiului și accesul la biblioteci puternice.

Godot 4: Godot 4 este un motor de joc open-source și gratuit, care oferă o gamă largă de instrumente și funcționalități pentru dezvoltarea jocurilor 2D și 3D. Acesta se remarcă prin interfața sa intuitivă, sistemul de scene flexibil și performanța optimizată. În proiectul nostru, Godot 4 a fost ales ca platformă principală de dezvoltare datorită ușurinței în utilizare, a comunității active și a suportului pentru C#, permițându-ne să creăm un joc "fun run clone" captivant și interactiv.

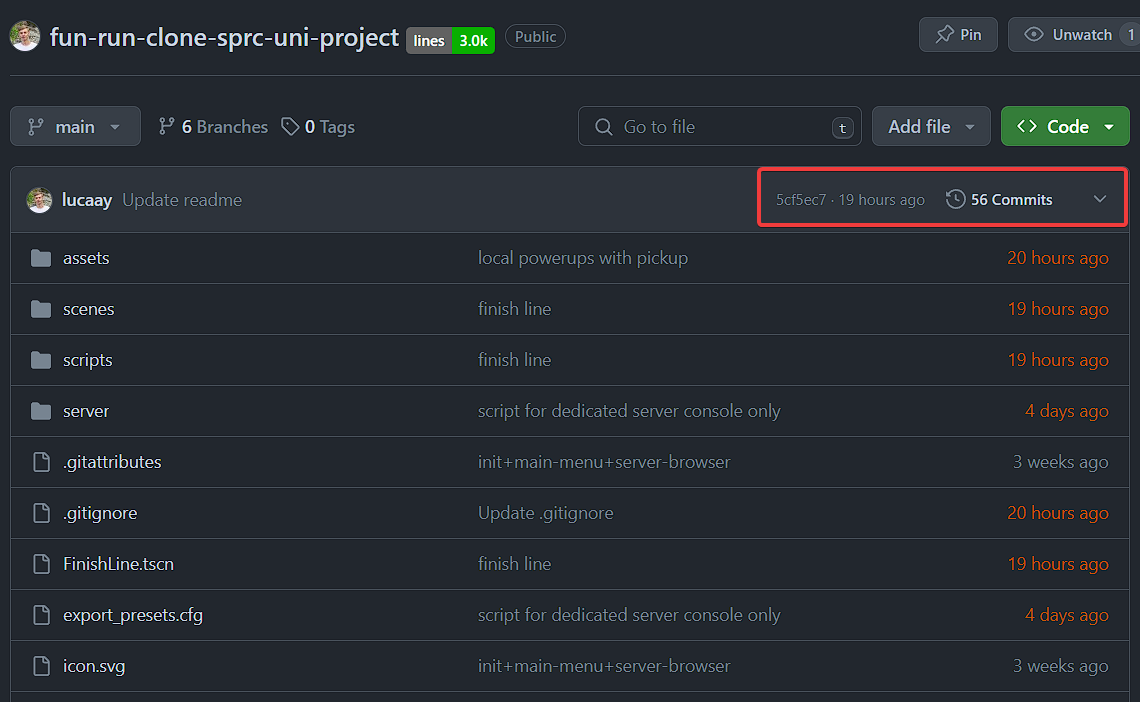
# Vizualizarea codului sursa

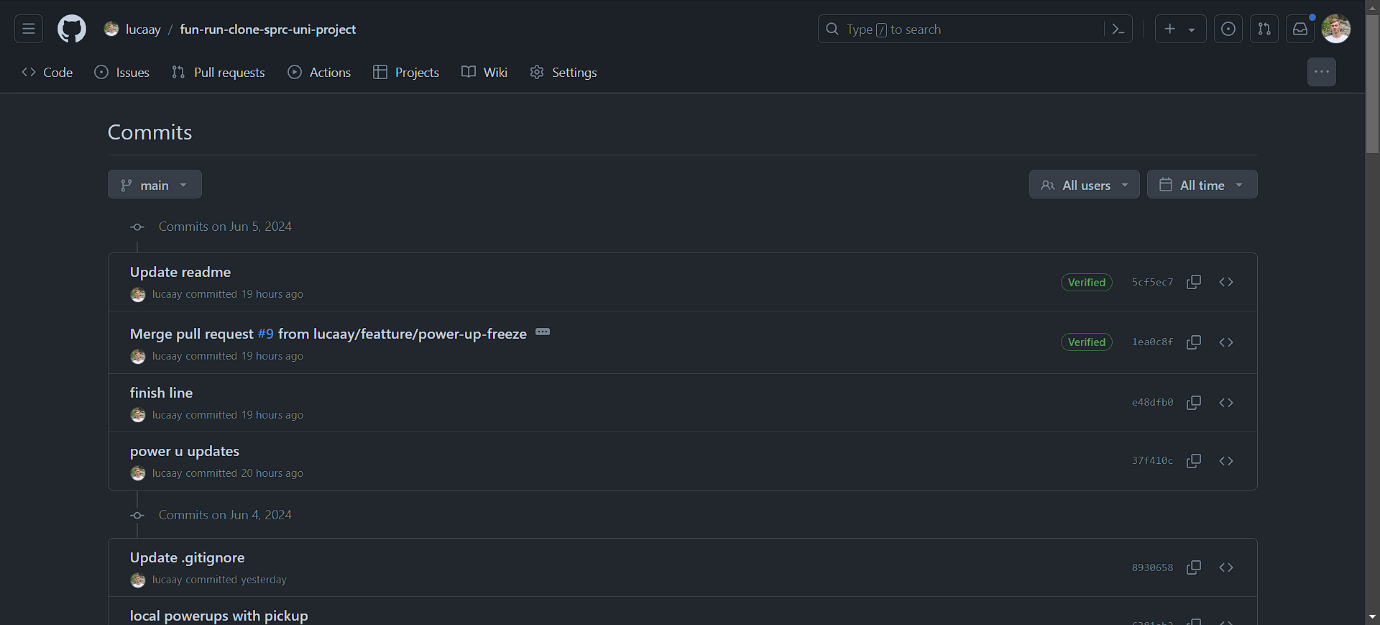
Codul sursa poate fi vizualizat pe linkul urmator:

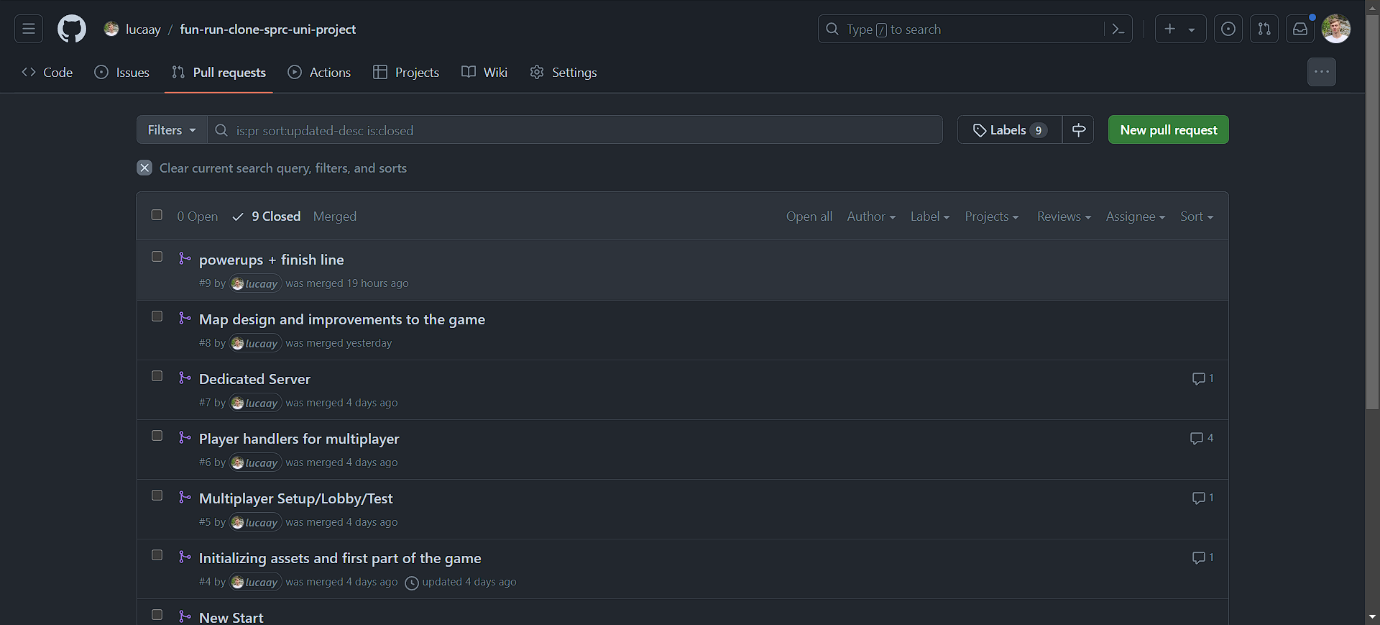
<https://github.com/lucaay/fun-run-clone-sprc-uni-project>

Aici pot fi vizualizate urmatoarele:

* Tot codul jocului
* Mai multe commituri ce arata modificarile realizate de-a lungul dezvoltarii jocului
* Codul impartit pe Pull Request uri reprezentand Etapele majore din dezvoltarea jocului.
* Cate un mesaj pentru fiecare commit ce explica modificarile realizate







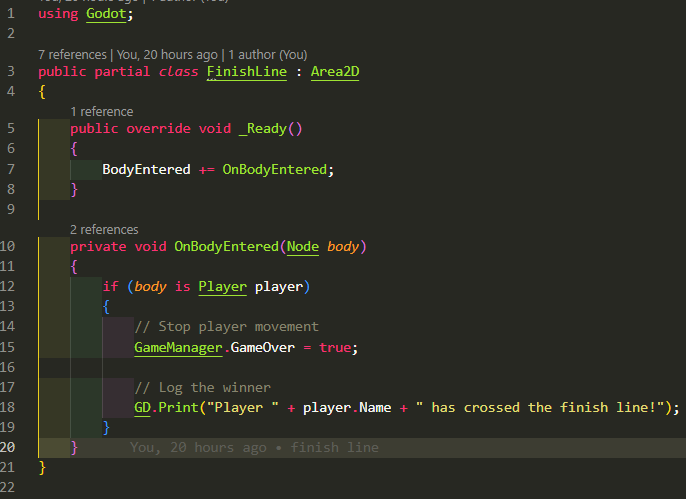
# Componente c# folosite

## Finish Line.cs

**Scop:** Detectează când un jucător atinge linia de sosire și declanșează sfârșitul jocului.

**Funcționare:**

* Așteaptă ca un obiect să intre în aria sa de coliziune.
* Verifică dacă obiectul este un jucător.
* Dacă da, semnalează sfârșitul jocului

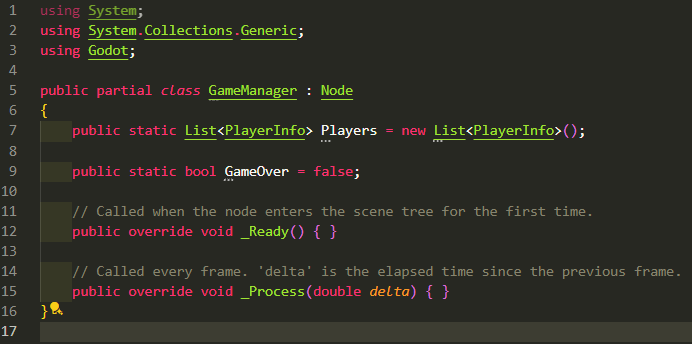


## GameManager.cs

**Scop:** Gestionează starea generală a jocului.

**Funcționare:**

* Păstrează o listă a jucătorilor conectați (Players).
* Menține o variabilă booleană pentru a indica dacă jocul s-a terminat (GameOver).



## MultiplayerController.cs

**Scop:** Gestionează conexiunea și comunicarea între jucători într-un mediu multiplayer.

**Funcționare:**

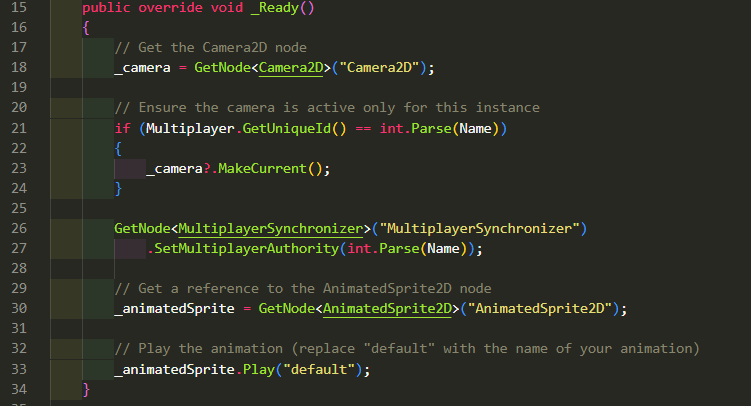
* Permite conectarea la un server existent dedicat.
* Trimite și primește informații între jucători, cum ar fi numele, poziția, acțiunile, etc.
* Gestionează evenimente precum conectarea, deconectarea sau începutul jocului.

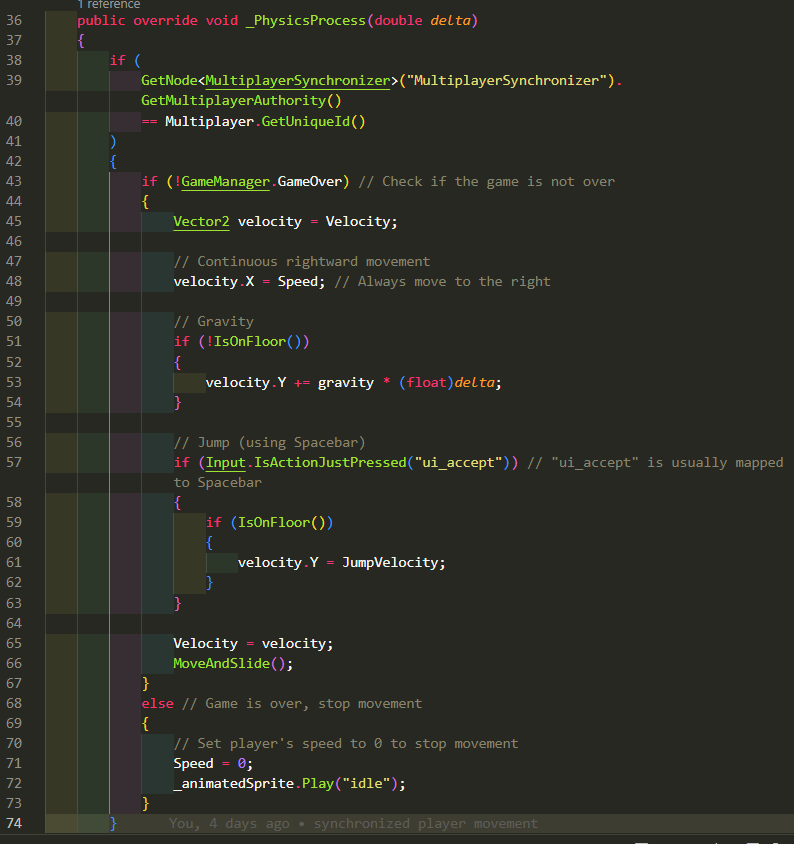
## Player.cs

**Scop:** Controlează mișcarea și acțiunile unui jucător.

**Funcționare:**

* Asigură mișcarea continuă a jucătorului spre dreapta.
* Permite jucătorului să sară.
* Gestionează coliziunile cu solul și alte obiecte.
* Sincronizează mișcarea și acțiunile jucătorului în multiplayer.



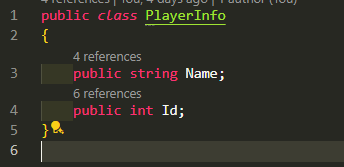


## PlayerInfo.cs

**Scop:** Stochează informații despre un jucător.

**Funcționare:**

* Conține proprietăți pentru numele și ID-ul jucătorului.
* Este folosită pentru a transmite informații despre jucători între server și clienți.

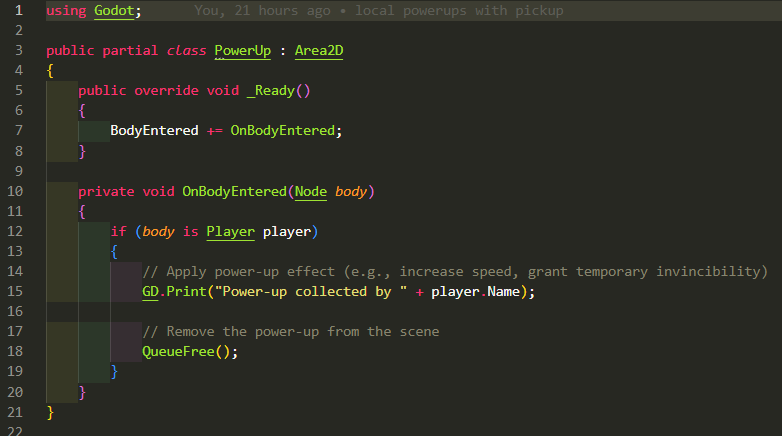


## PowerUp.cs

**Scop:** Definește comportamentul unui power-up.

**Funcționare:**

* Așteaptă ca un obiect să intre în aria sa de coliziune.
* Verifică dacă obiectul este un jucător.
* Dacă da, aplică efectul power-up-ului asupra jucătorului și se auto-distruge.

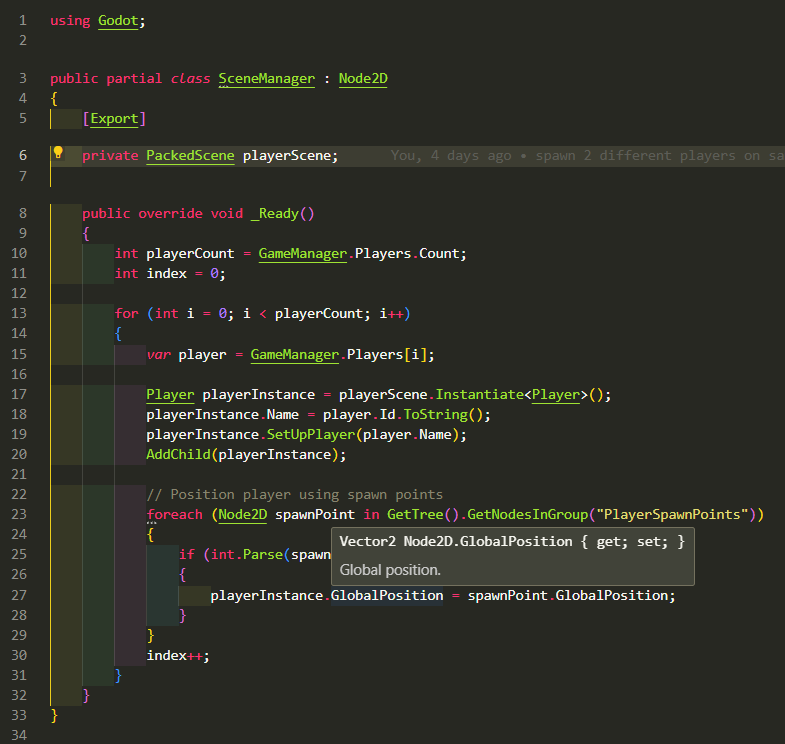


## SceneManager.cs

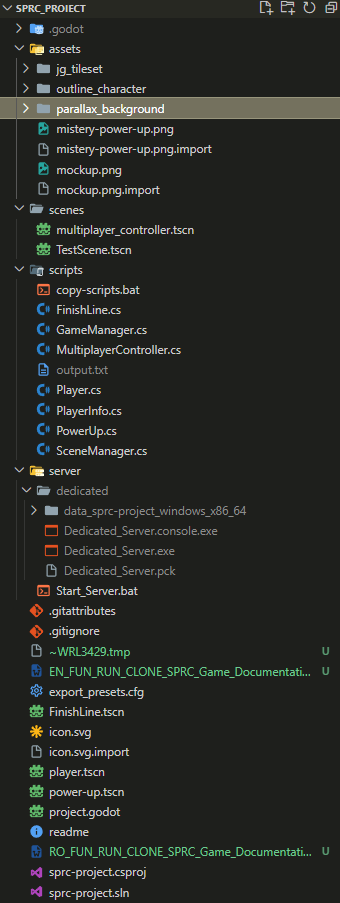
**Scop:** Gestionează încărcarea și plasarea elementelor în scenă, în special a jucătorilor.

**Funcționare:**

* Creează instanțe ale jucătorului pentru fiecare jucător conectat.
* Poziționează jucătorii pe punctele de spawn desemnate.



# Arhitectura si Structura sistemului



# Bibliografie resurse si librarii

1. <https://godotengine.org/>
2. <https://itch.io/game-assets>
3. IDE: <https://code.visualstudio.com/>
4. GIT Version Control: <https://github.com/lucaay/>
5. GIT Client: <https://desktop.github.com/>