

Documentatie

Lupu Andrei
Moise Andrei Marian
Buhnia Tudor Alexandru
Voinea Ionut Florin

Ideea Proiectului

Ne-am gandit sa facem un joc 2D în care sa modifici niște date de intrare pentru a rezulta output-ul necesar.

Input-urile la noi sunt matrici de 3×3 pe care le putem modifica prin tăierea și rotirea obiectului care contine matricea.

Fiecare modificare este reprezentată de cate un obiect separat care are un preț. În funcție de complexitatea unui obiect acesta poate valora mai mult, astfel am creat o economie bazată pe obiectele rezultate din matricea 3×3 .

Aspecte Tehnice

Jocul este facut in Unity, folosindu-ne de C# ca limbaj de programare pentru majoritatea scripturilor. Am ales Unity pentru ca este unul dintre cele mai accesibile game engine-uri și pentru ca este foarte potrivit pentru jocurile 2D, de obicei este mai folosit pentru platformere, însă noi am vrut să facem ceva diferit cu un top down factory building similar cu jocuri precum Factorio si Shapez.

C# este de asemenea un limbaj foarte similar cu C++ ceea ce nu ne-a ridicat mai multe probleme ca de obicei

Block

Blocul este un obiect care tine în memorie o matrice de 3x3 booleene și randeaza la rulare o matrice formată din 2 sprite-uri, unul negru pentru True/1 și unul gri ușor transparent pentru False/0.

Obiectul are de asemenea funcții pentru a-i modifica și citi matricea pentru ca acesta să poată interacționa cu obiectele din jur.

Dificultati:

- Am avut un bug în care la crearea unui obiect prin funcția de modificare se randa de 2 ori obiectul, odata din functia de Start() și incaodata din funcția de Update(). Pentru a opri redarea dublată am folosit o variabila auxiliara care sa fie adevărată numai cat timp se modifică sprite-urile, astfel nu putem sa avem același proces de 2 ori.
- O alta problema pe care am avut-o a fost la orientarea casutelor si poziționarea lor în jurul originii relative a obiectului, este insa doar o problema vizuala si nu am rezolvat-o inca.

Cutter

Cutter-ul este un obiect care citește Block-ul pe care l-a primit și îl “taie” cu sablonul pe care-l primește. Tăierea funcționează precum o ștampilă, obiectele trimise mai departe având găuri în ele. Pe scurt modifica obiectul cu False unde a avut de taiat și în rest le lasa normal.

Pentru acest obiect am folosit o adaptare a BFS pe matrice, similară cu algoritmul lui lee pentru a găsi componentele conexe, Cutter-ul având opțiunea sa lase o matrice cu găuri în ea sau sa creeze 2 obiecte diferite, fiecare obiect având toate căsuțele unite.

Cutter-ul are de asemenea un boolean care sa îi schimbe modul de funcționare, unul dintre ele este cel de despărțire a blocului în mai multe bloculete, iar celălalt doar modifica valorile din obiect fără a mai face alte modificări.

Dificultati:

- Atribuirea corecta a datelor obiectelor ce ies din el. Avand aici problema care a ridicat bug-ul de la Block cu Update si Start.
- Creerea unui obiect nou, conform unei componente conexe în cazul în care cutterul scoate 2 obiecte.

Rotator

Rotatorul este un obiect plasabil în zona de lucru, ce are ca scop rotirea block-urilor la 90 de grade. Acesta poate sa le rotească fie în sensul acelor de ceasornic fie invers, avand in boolean care modifica funcționarea lui.

Acesta este cel mai simplu obiect din proiectul nostru avand în mare parte doar o rotire de matrice a Block-ului primit și crearea unui Block nou cu matricea modificata

Dificultati:

- În urma creării și testării obiectului am observat că ordinea în care se randeaza obiectele era defapt greșită acestea fiind randate din colțul din stânga jos în dreapta sus, ceea ce a creat un alt bug pentru randarea Block-ului.

Conveyor Belt

Conveyor Beltul este un obiect ce are ca scop deplasarea obiectelor de la un punct la altul (de la Spawner la Cutter, Cutter la Rotator, Rotator la Storage Unit, etc.). Cand conveyor beltul este alcatuit din mai multe segmente orizontale sau verticale, se calculeaza suma totala a lungimii segmentelor ce trebuie parcurse. Viteza obiectului se mentine constanta afland ce procentaj reprezinta segmentul curent din lungimea totala.

Dificultati: Gasirea unei viteze constante atunci cand banda rulanta nu este reprezentata doar de o linie dreapta, aceasta continand si curbe. Gasirea unui mod eficient de a permite jucatorului sa poata plasa conveyor beltul asa cum isi doreste folosind drag and drop(inca lucram la asta).

Spawner

Spawner-ul este un obiect care creeaza Block-uri, acesta are in memorie o matrice pe care o aplica Block-ului creat. Mai are de asemenea o variabila care îi da un interval la care sa le creeze. Fiecare spawner este conectat la un conveyor belt, obiectele create fiind preluate de acesta.

Cu spawner-ul nu am avut mari probleme fiind doar un obiect care creeaza alte obiecte, ceea ce in unity se face destul de ușor.

Storage Unit (work in progress)

Storage Unit-ul este un obiect care genereaza o matrice aleatoare de booleanuri, scopul jocului fiind sa obținem obiectele cerute de storage folosind cuttere și rotatoare. În prezent fiecare storage cere 10 obiecte. Vom modifica cerințele pe viitor în funcție de nivel. De asemenea fiecare obiect obținut va aduce o suma de bani.

Ce mai avem de făcut?

1. Grafica pentru majoritatea obiectelor și elemente UI.
2. De terminat Storage Unit care sa verifice tipul obiectelor ce ajung în el și dacă se îndeplinește taskul cerut.
3. Legarea Cutter-ului de propriul UI-PopUp.
4. Toate elementele din toolbar/inventar să poată fi plasate în zona de lucru.
5. Implementarea unui algoritm de leveling/misiuni.
6. Gasirea unui mod in care jucatorul poate plasa Conveyor Beltul pe grid in functie de cum isi doreste.
7. Fixarea unui bug în care unul din obiectele scoase din Cutter se întoarce în Conveyor Belt