The island generator

Max Kersten - *3020282*

Functie generator

De generator dient een overtuigend eiland te genereren. Het eiland moet verschillende layers bevatten. Deze lagen delen het eiland op in verschillende gebieden. Deze gebieden kunnen bestaan uit; zand, gras, steen en of sneeuw. Op ieder van deze layers kunnen er verschillende resources spawnen. Met deze resources kan de speler een interactie aangaan. Verder dient de generator ook het eiland “aan te kleden”. De generator spawnt op iedere laag een ander soort foliage. Als voorbeeld; Op zand spawnt de generator allemaal verschillende soorten schelpen. En in de zee spawnt het allemaal verschillende types zeewier. Met deze objecten kan de speler geen interactie aangaan.

Inspiratie

Ik heb voor deze generatore niet echt een inspiratiebron. Maar als ik een paar games zou mogen uitkiezen die te maken hebben met het genereren van een terrein met resources, denk ik al snel aan Valheim, No Man’s Sky en Elite Dangerous. In deze spellen worden er objecten gespawned op een procedural gegenereed terrein waarmee de speler een interactie mee kan aangaan.

De generator

Terrein mesh

Als eerste genereren we een noise-map. Deze map wordt gegenereerd op basis van een aantal gegeven parameters.

Noise.GenerateNoiseMap(*int* ***width,*** *int* ***height,*** *int* ***seed,*** *float* ***scale,*** *int* ***octaves,*** *float* ***persistence,*** *float* ***lacunarity,*** *Vector2* ***offset)***

De belangrijkste parameters

Width en Height: Deze parameters bepalen de resolutie van het terrein en de mesh die hierop vormt. een Width van 100 maakt het terrein 100 cellen breedt en een Height van 100 maakt het terrein 100 cellen hoog.

Seed: Deze parameter zet de “random.System functie” Op basis van deze gegeven random seed zal de generator een random map genereren.

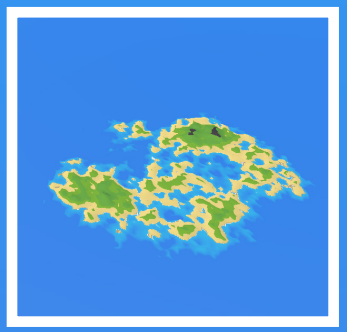
Octaves: Deze parameter gaat over hoeveel “noise lagen” er in de map zijn.

Lacunarity: Een vermenigvuldiger die bepaalt hoe snel de frequentie toeneemt voor elk opeenvolgend octaaf. Dit maakt het terrein “ruiger”.

Percistance: Een perameter die bepaald hoe zwaar de Lacunarity een effect moet hebben op de Octaven.

Seed: 63734928
Octaves: 1
Persistance: 0
Lacunarity: 1

Seed: 63734928
Octaves: 2
Persistance: 0.5
Lacunarity: 2.35

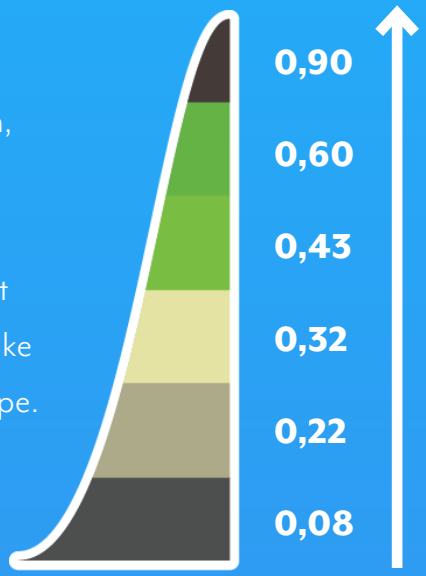


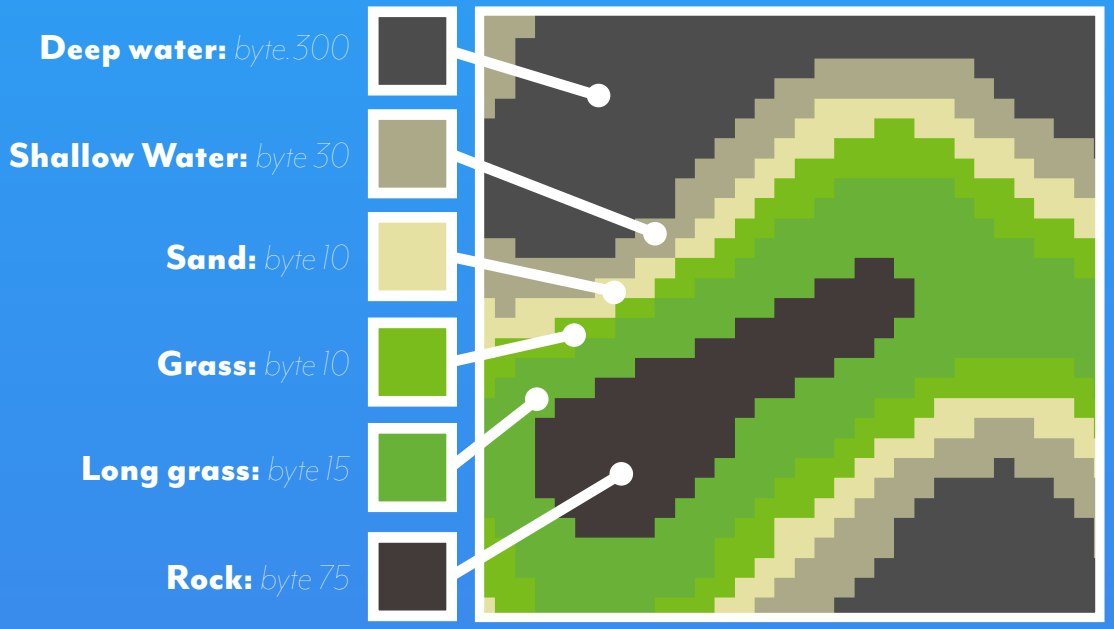
Seed: 63734928
Octaves: 5
Persistance: 0.5
Lacunarity: 5.0

De generator

Terrein mesh

Als we eenmaal een multidimensionale array van floats hebben, gaat het systeem hier doorheen itereren. Tegelijk maken we een nieuw multidimensionaal array aan. Dit array bestaat uit een enum Celltype. Dit Celltype is afhankelijk van de value die in het geïtereerde float element zit. Het systeem kijkt vervolgens in welke layer deze value valt. En geeft de Celltype[x , y] het correcte type.





Cost field

Iedere CellType heeft zijn eigen kosten. Deze kosten bepaalt hoe een Unit over het grid zal lopen.

De generator

Integration field

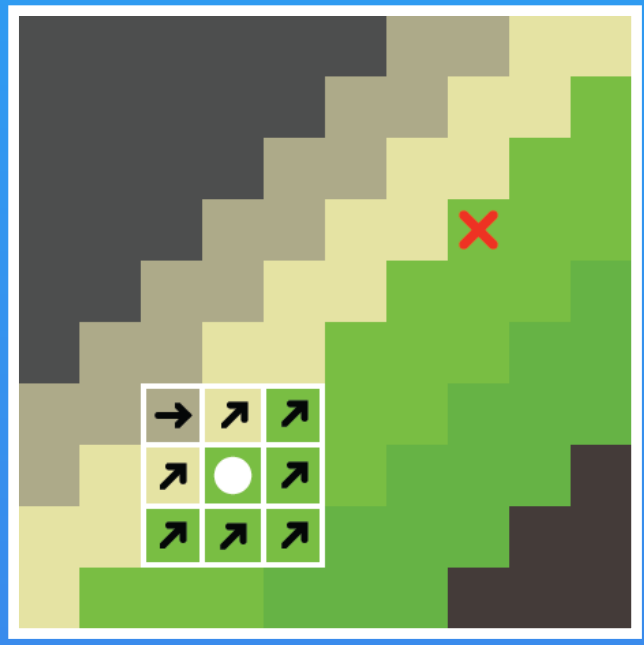
Om kosten te besparen tijdens runtime zal ieder object of groep van Units in de wereld een eigen integration field krijgen. Zo is er maar èèn integration field nodig als meerdere Units een zelfde object of positie als target hebben. Dit field wordt via multithreading gegenereerd. Dit zorgt voor minimale impact op performance voor de speler.

Flow field

Als een Unit over het integration field loopt creëert hij voor zichzelf een klein lokaal flow field. Het unit kijkt in alle neighbour-Cellen om te bepalen wat de beste en goedkoopste richting is die de Unit moet nemen om bij zijn doel te komen.

Intergration field

Flow field



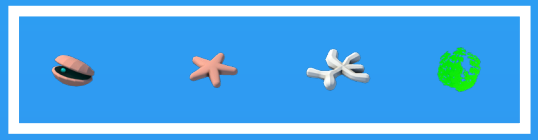
De generator

Resource layers

Een Resource-Layer is een element wat informatie bevat over wat het systeem op een layer van een bepaalt type moet plaatsen. Hij houd bij hoeveel er van een bepaalt type resource is geplaatst en hoeveel hij er uiteindelijk moet plaatsen. Verder heeft dit object ook informatie over de foliage van de layer. Deze informatie bestaat uit; Het percentage of 'kans' dat je een object moet plaatsen op een Cell, hoeveel objecten je dan op de Cell mag plaatsen en hoe de geplaatse objecten staan op de Cell. Met de foliage-Objecten kan de speler GEEN interactie aangaan.

CellType: Sand

List GameObjects foliageObjects



Spawn percentage: 32%

How many spawns per Cell

Resource spawn options 0

ResourceType: Wood

List GameObjects Resources

Maximum amount: 5

Minimum amount: 10

List InstantiatedObjects

Resource spawn options 1

ResourceType: Stone

List GameObjects Resources

Maximum amount: 10

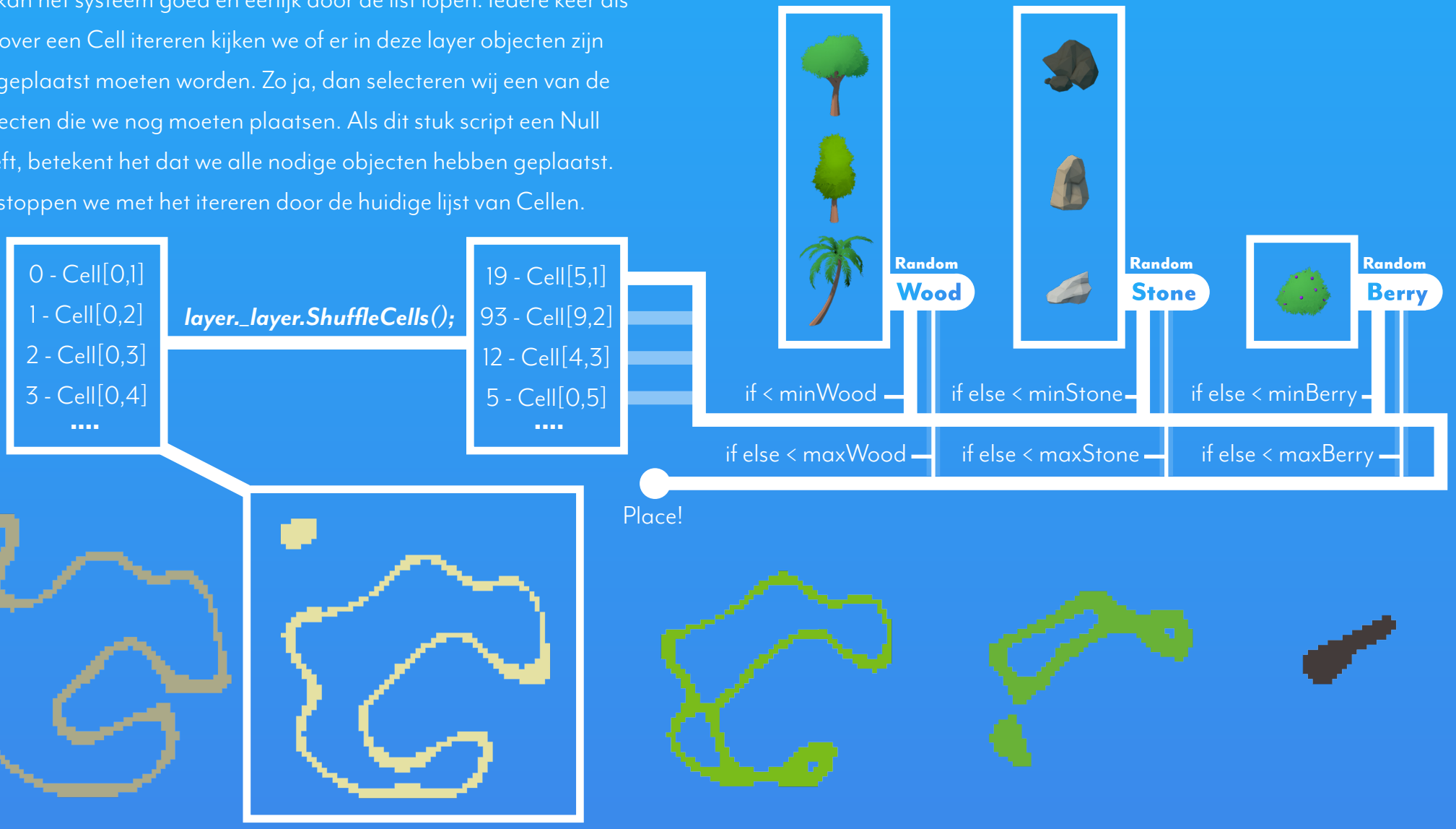
Minimum amount: 25

List InstantiatedObjects

De generator

Resources plaatsen

De generator verzamelt eerst ieder type Cell in een eigen lijst van cellen. Daarna schud het systeem de lijst waarmee hij gaat werken. Zo kan het systeem goed en eerlijk door de list lopen. Iedere keer als we over een Cell itereren kijken we of er in deze layer objecten zijn de geplaatst moeten worden. Zo ja, dan selecteren wij een van de objecten die we nog moeten plaatsen. Als dit stuk script een Null geeft, betekent het dat we alle nodige objecten hebben geplaatst. En stoppen we met het itereren door de huidige lijst van Cellen.



De generator

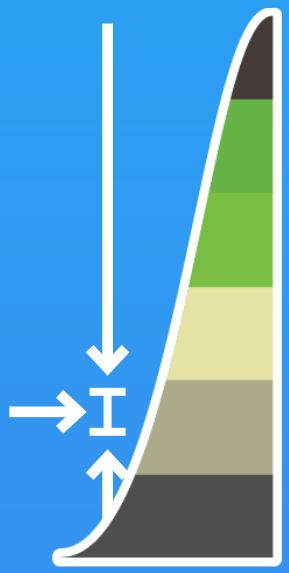
Spawn positie van de Units

Om een spawn-positie voor onze eerste units te bepalen, verzamelen we eerst alle Cellen die binnen een bepaalde range vallen. Als het systeem er een heeft uitgekozen spawnen we hier ons start object op. Verder moeten we er ook voor zorgen dat het start-object richting de goede kant staat geroteerd. We willen ten alle tijden met onze 'voorkant' richting de kust staan. Het systeem kijkt naar de normal van de Cell, en berekend te rotatie die het object dient te hebben.

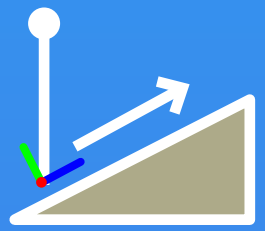
```
1 reference
private List<Cell> CollectPossibleSpawnCells(GameObject parentTransform)
{
    List<Cell> correctType = new List<Cell>();
    List<Cell> allCells = GridController.Instance.GenesisField._allCells;

    foreach (Cell cell in allCells)
    {
        if (cell._type == CellType.shallowWater)
        {
            Vector3 rayOffset = new Vector3(0, 10f, 0);
            Vector3 rayDirection = parentTransform.transform.TransformDirection(Vector3.down);
            Vector3 rayPosition = cell._worldPosition + rayOffset;

            RaycastHit ray01;
            if (Physics.Raycast(rayPosition, rayDirection, out ray01, Mathf.Infinity, _layerMask))
            {
                if (ray01.point.y > .5f)
                {
                    correctType.Add(cell);
                }
            }
        }
    }
    return correctType;
}
```



```
//Rotate spawn-parent to coast
Vector3 left = Vector3.Cross(ray.normal, Vector3.up);
Vector3 slope = Vector3.Cross(ray.normal, left);
parentTransform.transform.LookAt(parentTransform.transform.position + slope, Vector3.up);
```



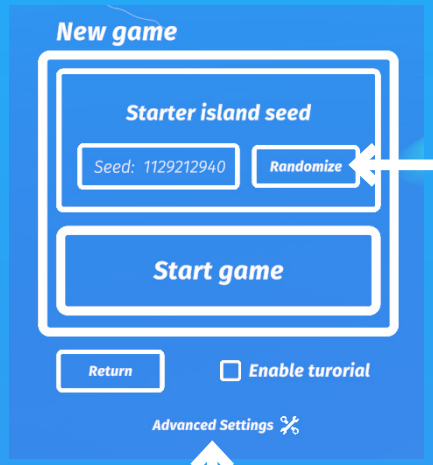
De generator in actie

Hoe te gebruiken

De generator

Stap 01

Klik op “new game”.
Dit opent een volgend menu.



Stap 02

Klik op randomize.
Dit zal een nieuwe seed en eiland genereren

Extra

Klik op Advanced Settings om dit menu te activeren

Advanced Settings

In dit menu kan je bepalen hoeveel er van een bepaalde soort “resource” er wordt geplaatst op een layer.

Re-generate

Na iedere aanpassing moet je het eiland opnieuw genereren. Druk op de “Re-generate” knop om het eiland te hergenereren

Hoe te gebruiken

Camera controls in game

Zooming
Gebruik om in en uit te zoomen het muiswiel.

Camera movement
Om de camera te bewegen dien je de rechtermuisknop ingedrukt te houden. Beweeg de muis om de camera te bewegen.

Camera rotation
Om de camera te draaien, moet je ALT en de rechtermuisknop ingedrukt houden. Beweeg de muis van links naar rechts om de camera te roteren.

Camere pitch
Om de pitch van de camera te veranderen, dien je ALT + de linkermuisknop in gedrukt te houden. Beweeg nu de muis van boven naar beneden om de pitch van de camera te veranderen.

Unit interactie in de game

Unit selection
Hover met je muis boven een Unit. Klik de unit met je rechtermuisknop om deze te selecteren.

Unit un-selection
Klik met je rechtermuisknop op een gebied waar geen objecten staan.

Unit movement
Om een unit te verplaatsen dien je hem eerst te selecteren. Eenmaal geselecteerd, houd shift ingedrukt. Klik en drag met de rechtermuisknop om een positie en rotatie aan de unit te geven. De unit zal deze actie uitvoeren als je alle knoppen loslaat.

Unit assignment
Om een unit een opdracht te geven, dien je de correcte unit voor de taak te hebben geselecteerd. Met de unit geselecteerd, druk op een object waar de unit een interactie mee moet aangaan. Bijvoobeeld het ‘Harvesten’ van hout van een boom.

Movement

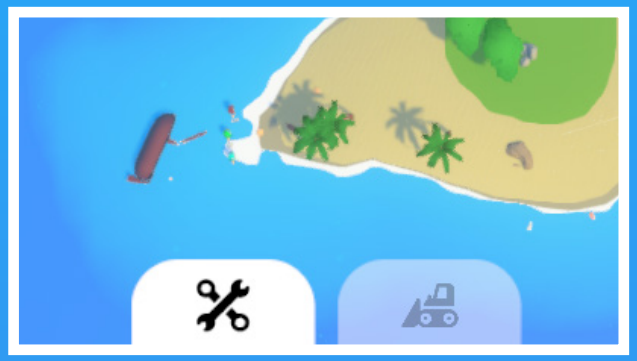
Task

Hoe te gebruiken

De bouw functie

Stap 01

Klik op deze knop om het
bouw menu te openen.



Stap 02

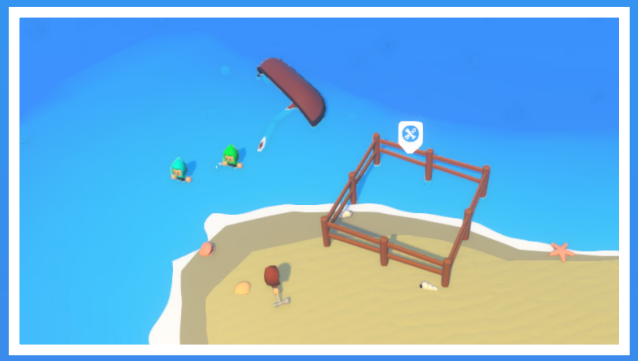
Kies een gebouw uit.
Blader door de tabjes!

Stap 03

Plaats het gekozen gebouw.
Met het scrollwiel + ALT kan je de
rotatie van het gebouw aanpassen.

Stap 04

Eenmaal geplaatst kan je een builder de
taak geven om het gebouw te bouwen.
Zorg er wel voor dat hij over de benodigde
resources beschikt.



Annuleren

Niet blij met de gekozen locatie?
Gebruik de Demolish funtie om een bouw-
opdracht te annulleren

Analyse

Je level generator moet minimaal:

De layout van een level genereren;

Het project voldoet hier aan omdat: Er kan een eiland worden gegenereerd en de gebruiker is instaat om zelf andere eilanden te genereren met meer of minder resources.

De content in het level plaatsen aan de hand van een procedure.

Het project voldoet hier aan omdat: Tijdens het genereren van een eiland gebruik ik een procedure om op een goed verdeelde manier objecten te plaatsen. Ook zijn er voorwaarden die bepalen of een object op de juiste plek geplaatst wordt. Ook wordt aan de hand van informatie uit het gegenereerde eiland een correcte begin positie bepaalt voor de Units die de speler van controleren.

De content die een level minimaal moet bevatten is:

Een begin positie;

Als de gebruiker op “start game” klikt, zoekt het systeem een correcte positie om de Units te plaatsen waarmee de speler het spel kan beginnen. Deze positie is afhankelijk van een aantal gegeven voorwaarden.

Objective (uitgang, missie, etc.);

Omdat dit spel nog in development is, is er nog geen echt functioneel doel. Maar het systeem is nu zo opgezet dat dit goed toegevoegd kan worden aan het spel. Het uiteindelijke doel van het eerste eiland is om de resources te verzamelen die de speler kan ‘harvesten’. Met deze verzamelde resources moet de speler een ‘settlement’ op bouwen. Als de speler zijn ‘settlement’ kan onderhouden, moet hij een schip bouwen en kan hij naar andere eilanden varen en nieuwe dingen ontdekken.

2 soorten enemies;

Het start-eiland genereerd nog geen enemies. Het plan is om op andere eilanden enemies te genereren. Het systeem is nu zo opgezet dat het zelfde ‘Units-systeem’ wat de speler gebruikt, ook gebruikt kan worden door een AI.

1 soort obstacle;

Ik vind deze eis wat moeilijk om te plaatsen.
Er worden obstakels geplaatst door de Generator, maar deze kunnen tegelijkertijd worden gezien als ‘loot’. Aangezien de speler de obstakels kan harvesten voor resources. Maar om wat specifiek te zijn, als er een object wordt geplaatst, wordt het type van de Cell onder het object als ‘Unpassable’ gemarkeerd. De Units kunnen nu niet meer over deze Cell lopen en dienen om te lopen als er een unpassable Cell in hun weg ligt.

1 soort loot;

Momenteel zijn er 3 resource soorten harvestable door de speler. Deze zijn; Wood, Stone en Berries. Als de speler wilt kan hij ook een wheat-field plaatsen en wheat farmen. Met deze wheat kan de speler in een ‘Workplace’ de Wheat omzetten in Bread.

Momenteel zijn Berries en Bread de enige vorm van voedsel voor de Units. Als de Units energie te kort komen, kan de speler een huis bouwen. Als de speler een huis assigned aan een Unit, kan de unit het huis in gaan om uit te rusten.

Resources

Sebastian Lague

Landmass generator:
https://www.youtube.com/watch?v=wbpMiKiSKm8&list=PLFt_AvWsXl0eBW2EiBtl_sxmDtSgZBxB3