

Product Design

Contents

Page	Title
3	Outline
	Elements
	Assets
4	Classes
4	DLL's
5	Houses
6	GameObject
7	HouseGenerator
8	RoofGenerator
9	BuildingGenerator
10	ChimneyGenerator
11	WindowGenerator
12	DoorGenerator
13	Houses
	Size
	Roof
	Building

Outline

The content we have decided to generate is that of houses. Using limited textures and a variety of filters our evidence shows that it is possible to generate thousands of unique houses. Using the .NET and XNA framework with C# we will write a program that, given a few parameters, will output many different image files, each containing a unique image of a house.

Elements

- Door
- Roof
- Building
- Chimney
- Windows

Assets

- White texture
- Brick textures

Classes

- Houses
- GameObject
- HouseGenerator (struct)
- RoofGenerator (struct)
- BuildingGenerator (struct)
- ChimneyGenerator (struct)
- WindowGenerator (struct)
- DoorGenerator (struct)

All classes, excluding Houses and HouseObject, returns a Texture2D. The idea is that the HouseGenerator calls all the below classes and uses the textures they return to build up a composite texture of the entire house, then return that to the Houses class, which then adds it to an array and calls it all again. Houses then eventually returns an array of Texture2D's.

The GameObject, however, will only contain a position, texture and rectangle.

DLL's

```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Content;
using Microsoft.Xna.Framework.GamerServices;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Media;
using Microsoft.Xna.Framework.Net;
using Microsoft.Xna.Framework.Storage;
```

The Microsoft.XNA reference will need to be added as well.

Houses

Parameters

- Map width
- Density
- Content manager
- Graphics device
- SpriteBatch

Variables

- An array of GameObject's for each house texture
- Constant width of a building
- Constant height of a floor

Load Method

This will load the appropriate textures and then generate all of the house textures, storing them in the appropriate array. Before this, however, it will calculate the number of houses it has to generate, then enter a loop state until it's finished generating all the necessary houses. Each loop will calculate the houses position and size, then generate the texture and store it in a new GameObject, with it's position and size (rectangle).

GameObject

Variables

- Position
- Collideable/Drawable Rectangle
- Texture

Methods

- Constructor
- Draw

Constructor Parameters

- Position
- Collideable/Drawable Rectangle
- Texture

Draw Parameters

- Scale
- SpriteBatch

Draw Method

- Draws the texture at it's given position and scale

HouseGenerator

Methods

- GenerateHouse
- GenerateBuilding
- GenerateRoof
- GenerateChimney
- GenerateWindows
- GenerateDoor

GenerateHouse Parameters

- Size
- SpriteBatch
- Graphics Device

GenerateBuilding Parameters

- Size

GenerateRoof Parameters

- Size

GenerateChimney Parameters

- Size

GenerateWindows Parameters

- Size

GenerateDoor Parameters

- Size

RoofGenerator

Methods

- GenerateRoof
- CalculatePeek
- GenerateTexture

GenerateRoof Parameters

- Size

CalculatePeek

- Size

GenerateTextures

- Peek

BuildingGenerator

Methods

- GenerateBuilding
- CalculateHeight
- CalculateWidth

GenerateBuilding Parameters

- Size

CalculateHeight Parameters

- Size

CalculateWidth Parameters

- Size

ChimneyGenerator

Methods

- GenerateRoof
- CalculateHeight
- CalculateWidth

GenerateRoof Parameters

- Size

CalculateHeight Parameters

- Size

CalculateWidth Parameters

- Size

WindowGenerator

Methods

- GenerateWindows
- CalculatePositions

GenerateWindows Parameters

- Size

CalculatePositions Parameters

- Size

DoorGenerator

Methods

- GenerateDoor

GenerateDoor Parameters

- Size

Houses

Size

The size contains two integers values. The first is multiplied against the building width to give the actual width; the second is multiplied against the floor height to get the actual height.

Roof

The roof is always one floor high and it's size is multiplied by the width to give you it's width.

Building

The building is broken up into units. Each unit's width and height are stored in the HouseGenerator struct as a constant. The size of each house is multiplied against the these constants to produce the actual size of the building.