

BIRKBECK, UNIVERSITY OF LONDON
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

MSc Computer Science

Project Proposal

Game With Procedurally Generated Content in Unity 3D

Supervisor: Keith Mannock

Author: Christian Radev

This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.

April 2017

Abstract

This project proposal outlines my background research in the area of graphics, computer games and algorithmic generation of 3D worlds and game objects. The purpose of which is to create a game with primarily procedurally generated content such as fully featured and realistic planets, solar system, characters, textures and more. The latest developments in the field clearly showed the impressive capabilities of procedural generation and its many benefits as compared to ordinary game development. The possibilities it offers are endless and requires much less resources to produce big projects with incredible results, than any other approach. My work so far shows how efficient and productive certain algorithms can be in generating height maps which are used to deform spherical mesh to resemble planet surface. This approach has endless variations of planetary detail and can be applied to generating the rest of the game content that is upcoming. It clearly shows that with minimal resources which are myself as a programmer, game engine (Unity 3D) and algorithms one can create a lot of game content which would normally take a big team of artists and game designers. By the end of this project it is expected that a fully featured procedurally generated game will be accomplished. The game itself is first-person role-playing in a fantasy theme, with the goal of exploring planets, gaining experience and advancing your character through quests and interaction with non-player characters. The results will produce a fun playable game with the option of being constantly expanded, simply by code and minimal to none manual content creation.

Contents

| | |
|---------------------------------|-----|
| Abstract | i |
| List of Figures | iii |
| List of Tables | iv |
| Abbreviations | v |
| Symbols | vi |
| | |
| 1 Introduction | 1 |
| 1.0.1 Benefits of PCG | 1 |
| 1.0.2 Project Goals | 2 |
| 1.0.3 Organisation | 2 |
| 1.1 Games Using PCG | 2 |
| 2 Background Review | 4 |
| | |
| A An Appendix | 5 |
| | |
| Bibliography | 7 |

List of Figures

| | | |
|-----|-------------------------------|---|
| 1.1 | Spore, planets | 3 |
| 1.2 | Spore, terrain | 3 |
| 1.3 | No Man Sky, planet | 3 |
| 1.4 | No Man Sky, terrain | 3 |

List of Tables

Abbreviations

PCG: Procedural Content Generation

NPC: Non - Player Character

Symbols

Chapter 1

Introduction

The definition for procedural content generation, abbreviated as PCG, is the practice of creating game content algorithmically with limited or indirect user input. It is a piece of code that creates new or modifies 3D objects and fills the scene with them, thus creating a world in which the player can exist in. Such content can be, levels, maps, textures, models, weather, weapons, quests, etc. PCG is a different approach as compared to the usual manual creation of content. In the latter, all the work is done by artists and game designers, meaning that the amount and diversity of content is limited by the number of people working and by their creativity. With algorithmic generation, on the other hand, all the repetitive tasks are taken care of by the computer and random diverse content is produced dynamically. One criticism of PCG is that the content can appear very repetitive as well with not having direct artistic control over the output, however this problem can be overcome by combining manual and procedural creation and carefully choosing which aspects of the game will be generated and which will be created by designers.

1.0.1 Benefits of PCG

The benefits of procedural generation are that much more content can be created in much shorter time, the content can be infinitely diverse as each generation can produce results with different seed for variables and it only takes computing power rather than people and creativity. There are well defined algorithms which are being used in the generation of any content available for a game, as it will be discussed later on. In terms of price PCG offers a much cheaper and faster solution, that is why many companies have started employing this method either entirely or in conjunction with manual design.

1.0.2 Project Goals

The purpose of this project is to create a fantasy first-person role-playing game in the game engine Unity 3D, where all or most of the content is generated by code and not produced manually. This engine is free and offers a lot of features and flexibility for game development, where the main programming language that will be used is C#. The game will feature a space environment with different planets in a solar system. Each planet will be as detailed and realistic as possible, with all elements that a planet includes such as diverse landscapes, water, weather, atmosphere, clouds, etc. The player will explore and interact with the environment as well with non-player characters (NPCs), with the aim of completing quests and advancing your character with new abilities and items.

1.0.3 Organisation

This proposal will present my research in the field of PCG and the implementation results obtained to date. It will discuss some of the methods that will be used in developing all aspect of the game discussed previously and what steps I am planning to take in order to achieve it. At this point in time the chapters are organised in the following way:

- Chapter 2: Background Review and Specifications
- Chapter 3: Theory
- Chapter 3: Implementation
- Chapter 4: Current Results
- Chapter 5: Conclusions

1.1 Games Using PCG

Games that included some sort of procedural generation have existed for long time such as The Elder Scrolls II: Daggerfall released by Bethesda Softworks in 1996, which takes place in mostly generated world. One of the most notable games are Spore released by Maxis, Electronic Arts in 2008 where a planet generator is built into their game engine, and allows creation of infinite diverse planets, which the player can visit and even edit.



FIGURE 1.1: Spore, planets



FIGURE 1.2: Spore, terrain

And by far the best example and most recent game is No Man Sky released by Hello Games in 2016. The game is fully procedurally generated and features an open universe with 18 quintillion diverse planets, in solar systems and galaxies. The game include all sort of planet details such as ecosystem, flora, fauna and behavior patterns, artificial structures, spaceships and more. This game is incredible in terms of its development and features and the fact that everything is generated by code, makes it the prime motivation behind my project to achieve similar results.



FIGURE 1.3: No Man Sky, planet



FIGURE 1.4: No Man Sky, terrain

Chapter 2

Background Review

Appendix A

An Appendix

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus at pulvinar nisi. Phasellus hendrerit, diam placerat interdum iaculis, mauris justo cursus risus, in viverra purus eros at ligula. Ut metus justo, consequat a tristique posuere, laoreet nec nibh. Etiam et scelerisque mauris. Phasellus vel massa magna. Ut non neque id tortor pharetra bibendum vitae sit amet nisi. Duis nec quam quam, sed euismod justo. Pellentesque eu tellus vitae ante tempus malesuada. Nunc accumsan, quam in congue consequat, lectus lectus dapibus erat, id aliquet urna neque at massa. Nulla facilisi. Morbi ullamcorper eleifend posuere. Donec libero leo, faucibus nec bibendum at, mattis et urna. Proin consectetur, nunc ut imperdiet lobortis, magna neque tincidunt lectus, id iaculis nisi justo id nibh. Pellentesque vel sem in erat vulputate faucibus molestie ut lorem.

Quisque tristique urna in lorem laoreet at laoreet quam congue. Donec dolor turpis, blandit non imperdiet aliquet, blandit et felis. In lorem nisi, pretium sit amet vestibulum sed, tempus et sem. Proin non ante turpis. Nulla imperdiet fringilla convallis. Vivamus vel bibendum nisl. Pellentesque justo lectus, molestie vel luctus sed, lobortis in libero. Nulla facilisi. Aliquam erat volutpat. Suspendisse vitae nunc nunc. Sed aliquet est suscipit sapien rhoncus non adipiscing nibh consequat. Aliquam metus urna, faucibus eu vulputate non, luctus eu justo.

Donec urna leo, vulputate vitae porta eu, vehicula blandit libero. Phasellus eget massa et leo condimentum mollis. Nullam molestie, justo at pellentesque vulputate, sapien velit ornare diam, nec gravida lacus augue non diam. Integer mattis lacus id libero ultrices sit amet mollis neque molestie. Integer ut leo eget mi volutpat congue. Vivamus sodales, turpis id venenatis placerat, tellus purus adipiscing magna, eu aliquam nibh dolor id nibh. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed cursus convallis quam nec vehicula. Sed vulputate neque eget odio fringilla ac sodales urna feugiat.

Phasellus nisi quam, volutpat non ullamcorper eget, congue fringilla leo. Cras et erat et nibh placerat commodo id ornare est. Nulla facilisi. Aenean pulvinar scelerisque eros eget interdum. Nunc pulvinar magna ut felis varius in hendrerit dolor accumsan. Nunc pellentesque magna quis magna bibendum non laoreet erat tincidunt. Nulla facilisi.

Duis eget massa sem, gravida interdum ipsum. Nulla nunc nisl, hendrerit sit amet commodo vel, varius id tellus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ac dolor est. Suspendisse ultrices tincidunt metus eget accumsan. Nullam facilisis, justo vitae convallis sollicitudin, eros augue malesuada metus, nec sagittis diam nibh ut sapien. Duis blandit lectus vitae lorem aliquam nec euismod nisi volutpat. Vestibulum ornare dictum tortor, at faucibus justo tempor non. Nulla facilisi. Cras non massa nunc, eget euismod purus. Nunc metus ipsum, euismod a consectetur vel, hendrerit nec nunc.

Bibliography