**Procedural World Generation**

**User Requirements Document**

**DIBRIS – Università di Genova. Scuola Politecnica, Corso di Ingegneria del Software 80154**

**DATA : 14/03/2018**

**VERSION: 1.0**

**Autori**

Stefano Sbarbaro

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versione** | **Data** | **Autore** | **Note** |
| 1.0 | 14/03/2018 | SS | Inizio |
| 1.1 | 26/03/2018 | SS | Precisazione del requirement sul seed |
| 1.2 | 10/04/2018 | SS | Ulteriore precisazione del requirement sul seed |
| 1.3 | 12/04/2018 | SS | Modifica riguardo le issues |

# 1​ Introduzione

## ​1.1​ Scopo del Documento

Questo documento riguarda il progetto ID-8 Procedural World Generation proposto da Untold Games s.r.l per il corso di Software Engineering(SE) presso l’Università di Genova. Il documento si promette di descrivere le funzionalità di un tool dedicato alla generazione procedurale di mappe virtuali utilizzate nell’ambito dei videogames.

## ​1.2​ Ambito Applicativo del Documento

Questo URD è fondamentale per la descrizione del progetto che vale come esame per il corso di SE.

## 1.3​ Definizioni e Acronimi

|  |  |
| --- | --- |
| **Acronimo-Nome** | **Definizione** |
| PWG | Procedural World Generation |
| SS | Stefano Sbarbaro |
| SE | Software Engineering |
| URD | User Requirements Document |

## ​1.5​ Overview del documento

Il documento è organizzato come segue: la prima parte del documento contiene una breve descrizione del problema che si vuole affrontare e una possibile soluzione; la seconda parte elenca le funzionalità aspettate dal sistema e la loro priorità di implementazione.

# ​2​ Descrizione Generale del Sistema

Procedural World Generation è un tool dedicato allo sviluppo e alla generazione in modo procedurale della piantina di una mappa, in particolare di una città, in tempo reale.

Il software si propone di essere in grado di creare contenuti in modo indipendente o più precisamente con un limitato supporto dell’intervento dell’umano; tutte le mappe prodotta dal tool dovranno essere sempre diverse tra loro se sarà differente il seed che le contraddistingue.

Partendo da questo prototipo solo in futuro verrà implementata la terza dimensione sfruttando proprio la piantina della mappa come la base dalla quale potranno essere modellati gli edifici e le diverse strutture che la compongono.

Il software dovrà integrarsi con l’engine grafico Unreal Engine 4 e rispettare alcuni parametri di gameplay, generandosi di conseguenza.

## 2.1​Contesto e Motivazioni

La generazione procedurale dei contenuti nel mondo dei videogames è sempre più utile, sia per efficienza che per ragioni di gameplay, quindi per motivi di velocità di realizzazione e gestione.

Uno dei maggiori goal dell'industria è quello di generare mappe che rispettino contemporaneamente diversi parametri di gameplay ed estetica;

la necessità di automatizzare i processi di creazione dei contenuti in tale maniera offre inoltre un grande vantaggio riguardo il risparmio sul tempo di computazione dato che tali contenuti non dovranno essere caricati tutti in una sola volta.

Molteplici algoritmi applicati allo sviluppo tramite la generazione procedurale stanno riscuotendo sempre più apprezzamento, grazie alle novità che apporta, come ad esempio l’incessante originalità nella produzione degli ambienti e la riduzione notevole dei costi di sviluppo.

Di fatto sono noti ormai essere molti gli sviluppatori indipendenti che attutiscono i costi delle fasi di level design proprio sfruttando queste tecniche.

Videogames basati interamente sulla Procedural Content Generation sono stati realizzati anche da piccoli team di programmatori con ambienti generati in ottima varietà e in real-time, rendendo in questo modo il gameplay infinito e più attraente.

## 2.2 Obiettivo del progetto

Analizzando lo stato dell'arte attuale nella generazione di mappe complesse (foreste, boschi, pianure, città, palazzi/case, interni degli edifici) e studiando l’implementazione di algoritmi specifici ( L-system, diagrammi di Voronoi, perlin noise) l’obbiettivo del progetto è quello di implementare una piccola demo di funzionamento.

L’idea fondamentalmente riguarda la creazione della piantina di una mappa di una città in 2D con vista dall’alto, cercando di immaginare che, tendenzialmente, dovrà essere possibile utilizzarla per la realizzazione futura in 3D delle strutture che la caratterizzano.

La mappa prodotta dovrà anche essere divisa in due o più quartieri/zone in modo tale da non essere costituita solo da un insieme unico di complessi edilizi e strade.

Il software dovrà essere in grado di generare a run time; per agevolare e snellire il salvataggio dei dati inoltre deve essere possibile rigenerare una mappa identica a partire da uno stesso seed.

2.3Utenti

Team Untold Games: le mappe realizzate dal tool verranno utilizzate come punto di partenza dal quale sarà possibile sviluppare in 3 dimensioni ambienti cittadini virtuali.

# 3​ User Requirement

In questa sezione descriveremo i requisiti dello strumento chiamato Procedural World Generation. In particolare ad ogni requisito daremo un id univoco e una priorità seguendo la seguente tabella:

|  |  |
| --- | --- |
| M | Mandatory. Requisito Obbligatorio. |
| D | Desiderable. Requisito che dovrebbe essere inserito nel sistema. |
| O | Optional. Una funzionalità marcata come optional può essere inserita nel sistema, a discrezione del manager del progetto. Ad esempio se il tempo di sviluppo è minore di quello previsto oppure se il costo di implementazione non è troppo alto. |
| E | Future Enhancement. Questo requisito viene lasciato per la prossima release. |

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrizione** | **Priorità** |
| **1** | *Seed:* per la generazione della mappa viene usato un numero cosiddetto seed: un intero a 32 bit casuale. Al variare di quest’ultimo la piantina sarà sempre diversa e unica ma per agevolare e snellire il salvataggio dei dati deve essere possibile rigenerare una mappa identica a partire da uno stesso seed.  Di fatto è un numero senza significato poiché non rappresenta nulla ma funziona da indicazione per l’algoritmo di randomizzazione. | **M** |
| **2** | *Divisione in zone:* al fine di non generare una mappa rappresentante un unico insieme di edifici, strade e strutture cittadine è richiesta la rappresentazione di due o più quartieri.  Ad esempio una zona industriale conterrà la pianta di edifici più grandi e poche strade mentre un quartiere popolare verrà rappresentato da una piantina con molte più case e percorsi. | **D** |
| **3** | *Efficienza:* l’algoritmo non deve impiegare un tempo esagerato per la produzione del risultato finale. | **D** |
| **4** | *2D:* la piantina deve essere in due dimensioni, visualizzabile in modo frontale dall’alto. | **M** |
| **5** | Motore grafico: il software dovrà essere in grado di integrarsi con il motore grafico Unreal Engine 4. | **M** |