**Creation of a data-driven algorithm for use in content creation**

Isaac Fagg

S4920008

Bournemouth University

BSc Games Design

**Abstract (0/200)**

I wish to request that my dissertation is assessed in accordance with marking option:

1: Lit Review 1400 words (30%); Dev Report 3000 words (55%);

* 1. **Introduction (0/1400)**

As the gaming industry continues to expand any content that hopes to engage a wide audience must be able to accommodate for the varied preferences. However, designing and creating individualised game content would take up too much time and resources. Instead there has been a rising interest in automatic generation of game content to assist with producing

* 1. **Background**

**Procedural Generation**

**General Procedural Generation**

A Multi-Faceted Surrogate Model for Search-based Procedural Content Generation

Evolving Levels using Grammatical Evolution

Mixed Initiative Content Creation

**Search-Based Procedural Content Generation**

Search-Based Procedural Content Generation

The representation of game content is a central question when it comes to evolutionary computation (Togelius, 2010). Search-Based Procedural Content Generation (SBPCG), which is a type of procedural generation that tests the generated content on its fitness based on set parameters. This is used in conjunction with an evolutionary algorithm that changes what the parameters can be. When using SBPCG the representation of the content is very important as it determines how the algorithm analyses the content.

**Track Based Procedural Generation**

Interactive evolution for PCG of tracks in high-end racing games

Automatic Track Generation for High-End Racing Games Using Evolutionary Computation

TrackGen: An interactive track generator for TORCS and Speed-Dreams

**Personalised Procedural Generation**

Adapting Models of Visual Aesthetics for Personalized Content Creation

Towards Player-Driven Procedural Content Generation

**Player Behaviour**

**Player Modelling**

Player Modelling

Defining Personas in Games Using Metrics Experience-Driven Procedural Content Generation

Measuring the experience of digital game enjoyment

Modelling Player Experience for Content Creation

Play-Persona: Modelling Player Behaviour in Computer Games

**Racing Games**

Making Racing Fun Through Player Modelling and Track Evolution

Towards automatic personalised content creation for racing games

**Track Analysis**

Towards a Generic Method of Evaluating Game Levels

**Analytics**

Learning Analytics for Serious Games

# Tracking Real-Time User Experience (TRUE): A comprehensive instrumentation solution for complex systems

**Evolutionary Algorithm**

**Other**

The big five personality dimensions and job performance

* 1. **Context**

**1.4 Research**

**2 Development and Implementation (0/3000)**

**Version Control**

As I worked across 3 different devices (Home PC, Laptop and University Computers) I needed a way to reliably transfer data between them. I decided to use GitHub as it a popular cloud-based version-control system, which is available on all platforms. Commits were performed regularly during and after any significant work.

**Unity Engine**

I chose to use Unity Engine to develop the game as it was the platform, I was most familiar with as well as providing excellent tools such as Analytics and the ability to port to mobile devices.

2D

**Gameplay**

Car

UI

**Track Representation**

**Track**

Each track is stored as a sorted array of two-dimensional points, in order to keep file sizes to a minimum. Once the track points are needed, they enter a generator which calculates new control points in order to create a path.

The curve is then closed, so that the track can be completed multiple times. I decided to have the tracks be closed circuit rather than an infinite road, as it would yield better results in the evolution process.

**Segments**

Each segment is comprised of 3 points, the start, middle and end points. Using vector maths, the distances and angles between them are calculated. With these measurements, it can be determined what direction the segment turns as well as its size and area. This is done for each point in the track to create an array of segments which are then used in the evolutionary algorithm.

**Track Generation (Pseudo-code)**

**Convex Hull**

**Radial**

**Minimap/Icon**

**Player Tracking**

Player Profiles

Analytics

During the Race (Checkpoints)

**Ratings**

Stars (Like/Dislike v 5 Stars)

Requests (How they are generated, what impact they have)

**Track Evolution (Pseudo-Code)**

Population

‘Mating’ (Restrictions etc.)

Children (How many are made)

**Track Selection**

Output Selection (From Children)

**Future**

How the algorithm would work going forward

Population

**3 Self-Assessment of Learning (0/400)**

**4. References**

**Procedural Generation**

Togelius, J. Yannakakis, G. Stanley, K and Browne, C., 2010. Search-Based Procedural Content Generation. Applications of Evolutionary Computation [online], 1, 141-150

**Player Behaviour**

**Content Analysis**

**Evolutionary Algorithm**