Intelligent Procedural Content Generation for Computer Games

Thomas Smith
Supervisor: Enrico Gerding

Problem: In modern computer game development, content production accounts for a large proportion of the initial (and in some cases, ongoing) outlay. As both budgets and in-game worlds get larger, there is increasing demand to offload some of these production efforts to automated systems. The concept of procedural content generators (PCG) has been around for some time, and they have been used in many successful games, but many of the advantages made available by these systems have so far been largely overlooked. When content is generated manually or algorithmically during the design phase of a game, it can only be created according to the designer's expectations of the players' needs. By instead generating content during the execution of the game, and using information about the player(s) as one of the system's inputs, PCG systems should be able to produce more dynamic experiences that can be far more tailored to enhance individual player's experiences than anything manually created. An intelligent procedural content generator (IPCG) should therefore consist of two parts: some means of evaluating (some aspect of) the player and generating a model, and a PCG system that is able to accept this model as an input and dynamically generate variants on its standard output based on the contents of the model.

Goals: As specified above, an intelligent PCG should consist of two subsystems: an evaluator and its companion generator. The aim of the project will be to create a simple game-like application that uses an IPCG system to produce dynamically variable content based on the player's behaviour. I will begin by creating a variable PCG that is able to produce content based on specimen player models, and then use environments created in this way to create and tune a player evaluator for further generation.

Scope: In order to attempt to ensure that the project goals remain achievable, the scope should be restricted to the simplest possible system. Based on inital inspection of the problem space and existing literature, it appears that this would be adjustment due to player skill in a 2D platforming environment. The project will be coded in Java, as that comprises the majority of my recent coding experience, and it has a wealth of 2D graphics drawing support which will simplify the less-relevant areas of coding. Similarly, many of the peripheral components traditionally included in computer games are irrelevant to the project and will not be needed.