TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Procedural Content Generation in Computer Games

Level Generation for Angry Birds using Genetic Algorithms

Autor
Laura Calle Caraballo

Directores
Juan Julián Merelo Guervós

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

May 15, 2018

ABSTRACT

ACKNOWLEDGMENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

PCG	Procedural Content Generation
NPC	Non-Playable Character

Part I

INTRODUCTION

1 INTRODUCTION

1.1 THE NECESSITY OF PROCEDURAL CONTENT GENERATION

Computer games are a relatively new form of media whose popularity has been increasing non-stop since they appeared in the 70s. Many challenges have arisen –and will keep doing so– throughout the evolution of the industry, from the early arcades to the most complex modern open-world video games. Developers have come up with all sorts of creative ways to overcome hardware limitations, delivering better graphics and audio. They have pushed the boundaries of the medium by finding new forms of interaction and engaging players with compelling storytelling and original game mechanics. Many of them are related to the fast pace at which the game industry is growing, reaching every passing year to broader audiences that demand a wider range of experiences. Crafting them requires great effort and a high consumption of resources. How do we create a vast amount of content that suits players expectations with lower investment? The answer can be replayability, adaptative content or reduction of designers' workload. All of which can be tackled using Procedural Content Generation (PCG).[2]

Replayability, also referred to as replay value, relies on how interesting is playing a game more than once. It is easy to understand why it would be a desirable feature for both players and developers: from the player's point of view, they can extend the game experience further–past the credits roll. For designers, replay value means their product offers more with less manually crafted content.

Games can also engage players by adapting gameplay elements to each individual player. In a literal sense, this would be impracticable. Instead, users are usually presented with options to adjust to their preferred style. What is more interesting, the game itself can change based on in-game player behaviour. It can regulate its difficulty level to fit the player's learning curve or create content that matches the player's taste.

Both applications above use PCG as a replacement for human designers. However, PCG can be used as a tool to assist developers. It can suggest what might be a base for later development, enhancing human creativity rather than displacing it.

1.2 WHAT IS PROCEDURAL CONTENT GENERATION

The definition of Procedural Content Generation (PCG) has been broadly discussed and there is not a single one. We have plenty of examples of what *is* and what *is not* PCG, but every definition struggles to cover all cases, either being too inclusive or too exclusive. The one we choose here balances well between the two, defining PCG as *the algorithmical creation of game content with limited or indirect user input*[3].

Although PCG often uses AI techniques, this definition does not include all uses of AI in games. We do not consider NPC behaviour or AI playing agents as content, thus they are not PCG either. Aesthetic elements, game rules, levels, items, stories and characters among others are considered content in this definition.

Note that computers nor video games are mentioned in the definition. In fact, PCG has its roots in analogical games. This may conflict with the *limited or indirect user input*, but it is reasonable to assume that following a detailed set of instructions –even if it is done by a human– is not *input*. The underlying concepts used much earlier by non-digital games still prevail in modern video games. Using an algorithm to assembly pre-design pieces is a common technique in tabletop role playing guides –where the algorithm usually consists in several dice rolls– such as *Dungeon & Dragons*. It is not surprising that one of the early adaptations of PCG to digital platforms aimed to generate monsters and dungeons for physical games.[1]

There are many PCG methods and it is necessary to look at some traits that characterize and differentiates them from each other: [2]

- *Online/offline*: In online generation, the PCG occurs during the game session, while the user is playing the game. If it is done before the game session or during development, we have offline generation.
- *Necessary/optional*: Procedurally generated content may be part of the essential structure of the game or can be additional to the game experience and can be discarded. In the first one the content is *necessary* and needs to be correct while the latter is *optional*.
- *Degree and dimensions of control*: As any other algorithm, a PCG method can have a number of parameters which affect the output. If it uses random numbers, the seed is one of them.
- *Generic/adaptative* Adaptative generation takes into account player's behaviour while generic does not. Although there are exceptions, most commercial games choose generic over adaptative.
- *Stochastic/deterministic* Deterministic PCG will produce the same output given the same input, in contrast to stochastic generation that is not easily replicated.

- *Constructive/generate-and-test* Generate-and-test produces potentially correct solutions that are tested and adjusted in each iteration before giving the actual output. Constructive methods build partial solutions and add on them.
- *Automatic/mixed authorship* PCG can be used as assisting tool for designers, whether the output is used as a base or as an interactive process. Then we talk about mixed authorship, as opposed to automatic generation where the designer does not take part.

1.3 EVOLUTIVE ALGORITHMS

Part II

THE SHOWCASE

You can put some informational part preamble text here. Illo principalmente su nos. Non message *occidental* anglo-romanic da. Debitas effortio simplificate sia se, auxiliar summarios da que, se avantiate publicationes via. Pan in terra summarios, capital interlingua se que. Al via multo esser specimen, campo responder que da. Le usate medical addresses pro, europa origine sanctificate nos se.

Part III

APPENDIX

BIBLIOGRAPHY

[1] Gillian Smith. "An Analog History of Procedural Content Generation." In: *FDG*. 2015.

[2] Julian Togelius, Noor Shaker, and Mark J. Nelson. "Introduction." In: *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Ed. by Noor Shaker, Julian Togelius, and Mark J. Nelson. Springer, 2016, pp. 1–15.

[3] Julian Togelius, Emil Kastbjerg, David Schedl, and Georgios N Yannakakis. "What is procedural content generation?: Mario on the borderline." In: *Proceedings of the 2nd international workshop on procedural content generation in games*. ACM. 2011, p. 3.

DECLARATION

Yo, Laura Calle Caraballo, alumno del Grado en Ingeniería Informática de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada con DNI *, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Granada, May 15, 2018

Laura Calle Caraballo

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/

Thank you very much for your feedback and contribution.

Final Version as of May 15, 2018 (`classicthesis` versiono).