

Preface

Welcome to the Procedural Content Generation in Games book.¹ This is, as far as we know, the first textbook about procedural content generation in games. As far as we know it is also the first book-length overview of the research field. We hope you find it useful, whether you are studying in a course, on your own, or are a researcher.

We wrote this book for two reasons. The first reason was that all three of us were doing research on PCG in games, and we wanted a good overview. As we come from somewhat different methodological backgrounds, we realized that many researchers did not know about methods that had been developed in other communities. For example, researchers using logic programming and those using evolutionary computation might not know that the other type of algorithms was applicable to the same problem; and researchers coming from computer graphics might not even know that artificial intelligence methods are being used for PCG problems. As PCG in games has just recently started to be seen as its own research field, this was not surprising, but pointed to the need for a book such as this one.

The second reason was that we were teaching a course on PCG (in fact, entitled simply “Procedural Content Generation in Games”) at the IT University of Copenhagen, where at the time the three of us were faculty members. When this course was started in 2010, it was probably the first of its kind in the world. Naturally, there was no textbook to teach it from, so we assembled a syllabus out of academic papers, mostly recent ones. As we taught the course in subsequent years, the syllabus matured, and we felt that we were ready to turn the content of our lectures into a textbook.

In writing the book, we based it on the structure of the existing course. In fact the first draft of this textbook was written quite literally as part of the fourth iteration of the course in autumn 2013. A draft of each chapter was completed in advance of the corresponding lecture, and given out as a handout to accompany the lecture. This ensured that a complete draft of the textbook was written within one semester, and perhaps more importantly ensures that the book is designed to be used as a textbook. Unfortunately, adding polish and finalising each chapter took a lot longer,

¹ Throughout the book, we will often use PCG as an acronym for procedural content generation.

which explains why the book did not come out in 2014. We believe however that the added time to work on the book was worth it, as the final product is much better than those early drafts were.

As you will see, the book is not strictly divided according to either methods or application domains. Most chapters introduce both a domain and method. This follows how we structured our PCG course, which we did in order to make the course more engaging and easier to teach: new methods are introduced together with interesting and relevant domains that demonstrate why they are practically useful, and can be used as settings for lab exercises to further experiment with the methods.

We decided early on that we wanted to involve many of the most active people in the research field, in particular those who had written the papers we relied on when teaching the course initially. Therefore, most chapters are coauthored with other researchers. This ensures that we have the most relevant expertise for the topic of each chapter.

As mentioned above, one of the main purposes of this book is as a textbook. Its origins as a set of course notes has also helped ensure that the book is “battle-tested” and ready to teach from. In particular, the book can be used as the main course text for a graduate-level or advanced undergraduate-level course on Procedural Content Generation in Games. It is assumed that students are familiar with basic artificial intelligence concepts (in particular heuristic search and basic concepts of logic and machine learning) and it is very beneficial (though not strictly necessary) that students have some experience of game development and using a game engine.

This book could be used as course literature in several ways. One is to base each lecture on a specific chapter, and assign a few recent papers from the literature related to that chapter as additional reading for that week. The assignment at the end of each chapter could then be used as that week’s assignment, and a conventional pen-and-paper exam could be held at the end of the course. Another way of organizing such a course, more closely aligning with the way the original course at the IT University of Copenhagen is taught, could be to use the first half of the semester for intensive lectures, covering two chapters per week. The second half of the semester is then used for group projects.² Finally, you can always use parts of the book, for example if you want to teach PCG as part of a larger course in AI for games. Most chapters are reasonably self-contained, with the most important dependencies being on the first two chapters, which establish core concepts and terminology. Therefore it is advisable to start with chapters 1 and 2 even if only using parts of the book.

The book is accompanied by a webpage, pcgbook.com, which contains digital versions of the book chapters, along with example lecture slides, links to relevant mailing groups and conferences, and other supplemental material. We welcome suggestions for new supplemental material (e.g. your own lecture slides) to add to the website. Our updated contact information can also be found there.

Of course, in any book-sized effort, one relies on the help of a large number of other people. Our first and foremost thanks go to our collaborators and co-authors on the various chapters of the book. We are also very grateful to our students in the

² Many good papers came out of the group projects from that course.

PCG course who endured various draft versions of the book, as well as our actual lectures, and in many cases provided very useful feedback. Several other colleagues have provided useful feedback or helped out in other ways; the list includes Steve Dahlskog, Amy Hoover and Aaron Isaksen.

Noor Shaker, Julian Togelius, and Mark Nelson