

Make Something That Makes Something: A Report On The First Procedural Generation Jam

Michael Cook

Computational Creativity Group
Goldsmiths, University of London
ccg.doc.gold.ac.uk

Abstract

We report on the first procedural generation jam, PROCJAM, an event designed to bring together artists, researchers and game developers to experiment with new techniques and applications for generating content for videogames. Much of the event's resulting work has applications beyond videogames, however, and we believe the event may be a strong platform for engaging creators and programmers in Computational Creativity in the future. We discuss the structure of the event, the results it yielded, and the potential future impact of such events on the Computational Creativity community.

Introduction

Procedural content generation (PCG) is a crucial and rapidly developing area of videogames technology (Togelius et al. 2011). PCG is a rich area of games culture – it has been used as a supplement for human effort (Interactive Data Visualization 2002), a source of wonder and unpredictability (Toy et al. 1980), a tool for artistic expression (Betts 2014), and a unique mechanical design tool (Yu 2009). Its increase in popularity and its growing importance in the culture of videogames has also been mirrored by a surge in the aesthetic of generative software in art, web culture (such as Twitterbots) and other creative media. These are all areas which have considerable overlap with Computational Creativity in terms of the techniques they use to generate artefacts, and represent a great opportunity to share the field's philosophy and theory with a vibrant, active community of people.

Game jams are increasingly common events within the game development community where people develop games under the constraints of both a time limit and some kind of common theme (which might be a technical constraint such as containing the game within 4 kilobytes of Java¹ or a creative constraint such as incorporating a theme like *fish-ing*²). Entrants to game jams typically fall within one of two categories: novices looking to use the event to create their first game, and experienced developers looking to experiment and innovate (Zook and Riedl 2013). In both cases the

short timescale helps encourage entrants to set themselves projects which are small enough to be easily completed.

Popular game jam formats are repeated at regular intervals throughout the year. Ludum Dare,³ one of the most popular, runs every four months. Entrants make a game in 48 hours from scratch, including game design, code, music and visual art, following a theme voted on by entrants in the week prior to the game jam. In December 2014, Ludum Dare 31 received over 2000 entries for the theme *Entire Game On One Screen*. By running repeatedly, these regular game jams build communities of creators who meet to create together, share ideas, give feedback on games (there is an extensive period of reviews and ratings after the jam) and often form collaborations or extend their jam entries into full commercial releases (Zucconi 2014). They form strong global communities who share ideas, draw in new practitioners, and push forward the state of the art (Gray et al. 2005).

In this paper we present a report on the first procedural generation jam, or PROCJAM, an event held in November 2014. The jam ran over nine days, starting with a streamed day of talks about procedural generation and ending with 138 entries being submitted in the form of games, tools, experimental prototypes and artworks. Although styled as a game jam, PROCJAM deviated from the traditional format in several important ways, which helped expand the appeal of the event beyond game developers and draw in people interested in generative techniques in general. We will go into these changes to the format in depth later in this paper, as we believe they are crucial to the success of PROCJAM and point to a format for generative events that could form the basis of Computational Creativity outreach in the future. We also will outline how PROCJAM itself is fostering work related to Computational Creativity and how this can grow in coming years.

We believe that the community-building and experimental aspects of game jams are extremely valuable, and make the game jam format ideal for engaging communities of programmers such as Twitterbot writers, programmer-artists and game developers with Computational Creativity, as well as being rich sources of inspiration and code which would be of benefit to everyone working in and around this field. Additionally, events like PROCJAM can also be valuable ways

¹<http://www.java4k.com>

²<http://www.fishingjam.com>

³<http://www.ludumdare.com/compo>

to expose people to Computational Creativity for the first time, in much the same way that game jams encourage people to try out making a game, and may serve as a useful model for student workshops and similar activities.

In this paper, we will outline the format for PROCJAM and explain how and why we deviated from several common elements of typical game jam organisation to create a better community for creating and sharing ideas. We then give several specific examples of entries to the jam and discuss their relevance to Computational Creativity. We follow this with a general analysis of entries, identifying issues related to Computational Creativity that arose in them, and also areas where our research could contribute to future entries to the event. Finally, we discuss the jam format as a model for outreach and engagement, and look ahead to the future of PROCJAM.

PROCJAM Format and Organisation

PROCJAM took place from November 8th to the 17th 2014, co-ordinated across the web using Twitter hashtags and a central website where people could submit their entries.⁴ Subsequently, the jam has registered its own website to co-ordinate the community and future events.⁵

The most common format for a game jam is as follows: at the beginning of the jam a theme is announced, normally on the jam's website so that people can take part from around the world. Participants then have 48 hours to develop a game from scratch, including art and music assets, that somehow incorporates the jam's theme. Entries are then submitted at the end of 48 hours. A review phase then takes place in which people vote for their favourite entries, with the voting pool consisting either of the general public, other entrants to the jam, or a select panel of judges. Prizes may be awarded to the winners.

This format for a game jam is very popular and is replicated hundreds of times a year from large-scale jams with hundreds of entrants down to small-scale local jams run between small groups of friends. With PROCJAM we made several changes to the standard game jam format with the express aim of increasing participation, particularly with those who had relevant experience writing generative software but had not interacted with game developers before. A secondary aim for the jam's format was to encourage experimentation and allow people to prototype unusual projects that stretched the state of the art in generative techniques for games.

Unlike most other game jams, making a game was not the only way to enter PROCJAM. Entrants could alternatively submit a piece of software that simply generated something (the jam's slogan was *Make Something That Makes Something*). Developing a game is a highly specific skill that people are unlikely to have unless they already work in games, and developing a game in the timeframe of a game jam is even more difficult. By relaxing this constraint, people who have interesting ideas, knowledge or skills can contribute generative systems to the jam that might spur on projects or

inspire developers to integrate new kinds of system in their future games. As a result, the jam received many entries in the form of complete games, but equally saw systems which generated dungeons and planets, weapon ideas and fabric designs, music loops and more. Bringing in people from different backgrounds helped make PROCJAM feel more like a melting pot of ideas and less like a competition.

We removed the requirement to produce original artwork and music for the jam, too. Since the primary focus of the jam was on new ideas in procedural generation, rather than testing game development skills, it didn't make sense to require people to put effort into elements of a game that were unrelated to their main contribution. This encourages people to enter the jam by relieving pressures on them to take on more work. We also removed a similar requirement that all code should be written from scratch. Game jam games tend to be very simplistic in nature because of their short development cycles, which works well for the goals the jams often have. However, in order to allow people to spend the week focusing on procedural generation, it made sense to allow them to use existing codebases or even entire games. One group of developers took a game they had been working on and added a procedural generation system to it as part of the jam. This doesn't just make the jam more appealing to outsiders, it can also allow deeper work to be done that builds on existing efforts (Hecker 2012).

By allowing entrants to make anything from a small script to a full game, and removing the restrictions on existing code and art assets, the process of evaluation becomes an issue. This raises the question of how to rate and compare entries when they are so varied in their origins. PROCJAM circumvents this simply by removing the ratings process - people are encouraged to comment on each others' entries and share them among one another, but there is no numerical rating system and no winners are declared. This solves the issue of comparing, say, a script which generates quilt patterns with a full murder mystery game. However it simultaneously encourages people to try out more experimental ideas without the intimidation of being judged and ranked by someone else's idea of what a good jam entry should be.

All of these changes have the same ultimate aims: to encourage people to take part, particularly those who are not game developers by trade, and to encourage experimentation and the sharing of new ideas.

We supplemented PROCJAM with a day of talks which we livestreamed on the web on the first day of the jam⁶. 80 people turned up to attend the day of talks, with 200 viewers tuning in to each talk throughout the day, and many hundreds more have viewed the recordings of the talks online since. The talks provided inspiration to jam entrants, with many citing the talks during the development of their jam entry, but they also provided an opportunity to be exposed to new views on generative systems - the speakers included an academic researcher, an artist and a creative director at an indie games studio. One of the aims of the event was to elevate procedural generation in games beyond "random levels", and having a variety of speakers giving talks was a

⁴<http://itch.io/jam/procjam>

⁵<http://www.procjam.com>

⁶<http://www.procjam.com/talks/2014>



Figure 1: A screenshot from Dreamer of Electric Sheep.

good way of doing this. We hope to have a speaker at next year's PROCJAM event to promote Computational Creativity as a new philosophy for procedural generation in games.

Selected Entries

In this section we will briefly describe and discuss three entries to the jam. We look at their most interesting features and the responses some of them received from the games community. We selected three projects that we believed would be of most interest to the Computational Creativity community, either because of their philosophy, innovative concepts, or the relationship between the techniques used and ideas within the field of Computational Creativity. We give details of where to find these entries, as well as all other jam entries, in the next section.

Dreamer of Electric Sheep

Dreamer of Electric Sheep is a text adventure submitted to PROCJAM by Tom Coxon, who also gave a talk at the jam's opening event about his procedurally-generated adventure game, *Lenna's Inception* (Bytten Games 2014). Like most text adventures, players are presented with descriptions of their surroundings and can manipulate the world by inputting simple commands for their character to execute. Dreamer attempts to procedurally generate the game world using a combination of ConceptNet (Liu and Singh 2004), a commonsense knowledge database, and Spritely⁷, a tool which generates game art from web searches. ConceptNet stores its data in the form of concepts, series of facts that are all about a similar topic. These facts are linked to one another through triplets (such as {*magazine*, *AtLocation*, *bookstore*}) which can be explored through an API which Dreamer uses. ConceptNet has seen use in academic Computational Creativity research, such as (Llano et al. 2014).

Dreamer searches the ConceptNet API for concepts which other concepts are linked to via the relationship *AtLocation*. It then populates those places with objects and characters that ConceptNet says should be found in them, and

⁷<http://www.github.com/gamesbyangelina/spritely>



Figure 2: A screenshot from Inquisitor showing the conclusion to a case. The player has correctly guessed the motive but not the murderer or the weapon.

uses Spritely to generate an illustration of the location. Spritely queries online image databases such as Google Images and Wikimedia Commons, searching for images that can be cleanly shrunk down in size with their backgrounds extracted, to make relatively clear sprites for use in games.

The player can perform common text adventure commands such as moving in the cardinal compass directions to travel between places, as well as picking up objects. However, because the game lacks deeper knowledge about the objects it places in each location, this can result in surreal interactions like picking up shop assistants and taking them with you. The game gets around this somewhat by being set inside a dream world, thereby allowing unusual things to take place without the game's sense of reality breaking.

The Inquisitor

The Inquisitor is a murder mystery game by Malcolm Brown. The game tasks the player with solving a murder by investigating the crime scene, discovering evidence, questioning witnesses and identifying the murderer, murder weapon and motive. The crime is procedurally generated, generating a cast of characters and relationships between them, simulating the movement of the characters before and after the murder (so that evidence such as blood trails and witnesses are realistic and consistent) and then leaving the player to put together the details within a time limit.

Although the individual generative techniques within The Inquisitor are not new *per se*, the way it uses them to generate murder mysteries is novel and quite effective. In particular, interviewing witnesses yields partial and sometimes conflicting information, forcing the player to take notes and draw up potential scenarios in which certain characters are lying, and information is procedurally redacted from certain kinds of evidence, leaving out the contents of a letter but revealing its author, for example, or smudging the name of its author but revealing unrequited love. The Inquisitor also

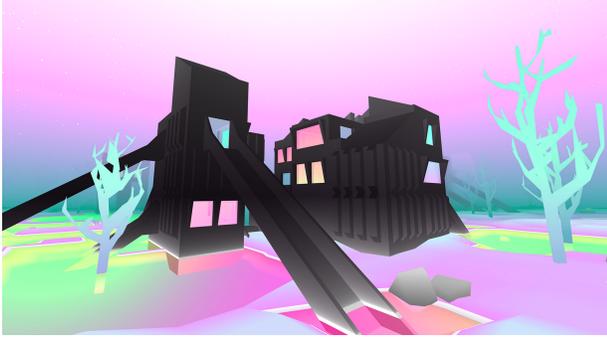


Figure 3: A screenshot from *Secret Habitat* showing a generated gallery in the generative game landscape. The player can walk inside and view the pieces, as well as exploring outside to find other galleries.

adds little additional touches on top of the game, such as a procedural system for applying accents to the pre-written dialogue. This takes dialogue written in plain English and then adds affectations to it to simulate a character who is drunk or has a particular speech impediment. To our knowledge this is a completely novel idea for content generation in games.

Secret Habitat

Secret Habitat is an ‘art gallery simulator’ and ambient exploration game by Strangethink. The game has no obvious end state. Instead, the player is encouraged to enjoy walking around the game’s generated world, entering the various buildings, viewing artworks and listening to audio recordings, all of which are also procedurally generated. The game seems to appeal to a particular aesthetic of wonder and mystery associated with generative software. One journalist wrote about the game:

[The paintings] seem to use a similar algorithm, or similar parts, or similar something, as colours, patterns, and other motifs repeat across them; you can recognise they’re part of a series. Seeing different spins on common themes can be delightful, and it’s awfully exciting when you discover one painting very different to the rest of its set. (O’Connor 2014)

Strangethink’s biography simply reads *I make strange computer worlds*⁸, and *Secret Habitat* leans towards digital, interactive art as much as it does the traditional ideas of videogames as systems of rules and objectives. Procedural generation has much overlap with both game development and interactive art, and generative software has a unique value in being able to present extremely large or infinite scales to a user (Betts 2014).

Analysis of Entries

PROCJAM received 138 entries in total, although more entries may exist that were not officially submitted, since we are aware that the jam was set as a class assignment in at

⁸<https://twitter.com/strangethink23>

least two universities and not all students submitted their entries to the site. All of the jam entries are available online⁹. The entries include full games, prototypes, tech demos, tools and libraries designed for developer use, as well as standalone generators and art pieces. We encourage the reader to visit the site and browse the entries themselves.

By way of a brief survey, we categorised the entries to the jam into two categories: *game*, or *tool*. The categories are defined loosely as follows: a **game** is any software designed to be interactive, but not for the purposes of producing something; a **tool** is any software designed to facilitate the generation of content as part of a larger creative activity. These definitions are not strict, but we offer them here as a rough partition of the entrants to the jam. Overall there were 79 game submissions and 59 tool submissions.

The games typically involved some kind of generative element in how they set up their game world, such as generating the 3D galleries in *Secret Habitat* or using simulation to create murder mystery scenarios in *The Inquisitor*. Most tools fell into one of two categories: some generated common kinds of content in accessible ways, such as world map generators of which multiple were submitted to the jam. This is partly because procedural generation lacks a cohesive community and established baseline software that solves common problems such as world generation - instead, developers tend to reinvent solutions to common problems repeatedly. We believe this is a key problem PROCJAM can target to benefit the generative software community in coming years. Other tools generated unusual kinds of content which are not commonly seen in games, like *GlyphGenerator*’s alphabets or *Bootleg*’s 3D shoe models. These are exciting because they break new ground in generative techniques and offer new applications for games, similar to those submitted to the jam.

Computational Creativity Issues

PROCJAM’s aim was primarily to produce generative tools and games, and to bring together both novices and experts to try out new ideas and learn more about the field. The theory and practice of Computational Creativity is gaining awareness in generative communities, but we believe that many developers are not confident about how to concretely use these ideas in the software they are building. That said, many entries to PROCJAM touch upon issues in the field, and others show clear areas where they could be extended to take advantage of results from Computational Creativity research.

Many of the tools explore co-creation, in which the software either creates alongside the user or tries to assist the user in achieving a particular goal, such as (Liapis, Yannakakis, and Togelius 2012). *Synthetic Poetry* allows the user to write poetry on alternating lines, along with one of three poet models based on Keats, Shakespeare and YouTube Comments. Other entries were more straightforward tools, such as *Nodemancer*, which allows users to specify the components of an item, such as a sword, and then lets the system design the specific details autonomously. Most

⁹<http://itch.io/jam/procjam#entries>

tools focused on the user retaining control, however: *SPARTAN* proudly announces that ‘the user has complete control over every step of the generation’ – encouraging people to explore ideas that break ideas like complete user control is something that will need to be emphasised in future years of the jam. We hope to expand the jam’s resource pool to include tutorials on basic Computational Creativity techniques and perhaps an invited talk from a Computational Creativity practitioner in a future event.

Issues relating to framing and context, as in (Charnley, Pease, and Colton 2012), arise in several entries including games like *The Inquisitor* which generate text as part of gameplay (for example, to provide dialogue and scene-setting for the murder mystery). This text is partly game content but also acts as contextual information that justifies decisions made by the generative system in producing other content. We have argued in the past that framing for game content generation is a broader concept than simply being ‘wall text’, and can extend to text that appears in-game to help the player understand and contextualise generated scenarios and systems (Cook 2014). Many of these games are beginning to explore these ideas, and we hope to see this trend continue in the future of the event.

Also related to framing, several entries play with the problem of communicating the logic, internal representation or behaviour of the generative system. Both *Diversitizer* and *Meadows* present the player with a natural environment populated with various flora. The locations of each plant, as well as its properties, are governed by procedural systems and vary each time the game is started. The player can gain an understanding of these parameters and the expressive range of the generator by observing the environment and repeatedly generating new worlds, even though the software does not communicate any information to the player through text. In this way, discovering the decisions made by the software become part of the purpose of interacting with the artefact, which is an interesting kind of *implicit* framing that is not often discussed in Computational Creativity discourse.

Many entries to the jam have clear ways in which they could be extended using techniques from Computational Creativity if the developer wished. Identifying simple ways in which common game ideas can be extended is important both in planning ‘code camp’ events for Computational Creativity, and for giving compelling examples at events like PROCJAM to show developers steps they can take to begin exploring the field. Many generative systems use parameters selected by the developer, such as *Infinity Explorer* which generates 3D worlds for the player to fly around in an airship. Encouraging developers to build their systems such that they can select parameters either based on external, contextual factors as in (Colton, Goodwin, and Veale 2012) or by evaluating its own output as in (Smith and Mateas 2011) is a good way to begin to move some of these generative systems in new directions.

The idea of the software evaluating its own work seems to be one of the most accessible ideas from Computational Creativity that generative software developers can start experimenting with. Generative systems tend to be developed in such a way that they are guaranteed never to produce bad

output: in other words, they rely on reorganising hand-made elements that the developer knows in advance will produce reliable content. This is an effective method for game development as it ensures the player will not be disappointed, however the culture of experimentation that we tried to encourage makes PROCJAM an ideal place for people to try out ideas that are less robust but perhaps more interesting and experimental.

Discussion

PROCJAM has relevance to the Computational Creativity community because it represents the founding of an interdisciplinary community of generative programmers who we hope will, over time, be introduced to and begin experimenting with ideas and approaches from Computational Creativity too. The results of the jam and the details about its organisation are important in their own right; that said, we believe that PROCJAM also holds interesting potential for future events that could strengthen and broaden the appeal and reach of our field.

Computational Creativity is a relatively young academic field that is still laying some of its foundations (Colton and Wiggins 2012). At the same time, many of the aims it has and the technologies and techniques it uses are highly relevant to movements in digital art, videogames and web culture as it stands today. In much the same way that outreach events must target academics in related fields, we should also look to engage these non-academic communities, to share solutions, and to encourage the adoption of our ideas. We often use the term ‘mere generation’ as a way of describing purely generative software, but we must also bear in mind that a lot of exciting and interesting work is being done in generative software communities, and we should seek to engage with these communities, learn from them, and try to convince them that ideas from Computational Creativity are exciting and interesting, too.

The format of an event like PROCJAM, particularly with some of the changes we made that we discussed earlier, make it ideal for informally bringing together several communities at once, making new connections and allowing them to demonstrate their working practices and techniques to one another. It also serves as a small-scale and self-contained event to set for students who may be interested in the field; PROCJAM was a credit assignment for one university class in particular, and we understand that feedback from the students was extremely positive. With encouragement and additional resources about Computational Creativity, future versions of PROCJAM (or perhaps a separate Computational Creativity jam) could serve as informal, global workshops that introduce people to the area in a practical way.

Despite their short length and highly applied nature, jams can serve a similar purpose for researchers as they do for programmers. There already exist examples of published research work which started off as a jam submission, in which an idea was quickly prototyped and then later developed after the jam (Cook and Colton 2014). PROCJAM also played host to jam games which were implemented to demonstrate an existing research tool or technique in a more concrete

way (Cerny 2014). We hope that subsequent PROCJAMs will see more researchers from this community take part to produce games or tools that demonstrate their work to game developers.

PROCJAM also leaves a legacy of code and ideas that persists after the jam has ended and gives the event lasting value in the months when it is not running. PROCJAM's 138 submissions offer ideas and inspiration, and in some cases code samples and open source projects. Entrants to the jam have already collaborated with one another, expressed intentions to develop their entries into full games, and in one case an academic issued an open call to the PROCJAM community for PhD applications, which was taken up by one entrant. PROCJAM had multiple features written about it in industry magazines and several others on major websites^{10,11}, and many of the games created for the jam were individually featured and written about as well. The jam's slogan, *Make Something That Makes Something*, has reappeared in other events relating to procedural content generation too¹². All of this shows that the jam is more than just the week in which it is held – it has a larger impact by creating a community of people that we hope will thrive.

To build upon this, we are planning for PROCJAM 2015 to have more resources ready online before the jam begins, aiming to encourage newcomers to writing generative games, tools and software. Through talking to entrants, we've identified several ways in which we can make the event easier to enter for people. We are hiring an artist to produce some public domain art assets for people to use, specifically designed to be easily recombined and mashed up in procedural systems. We're also talking to developers and researchers with the aim of producing some short tutorials that demonstrate simple generative techniques. These resources will persist beyond the jam itself, hopefully making it easier for people to begin learning about generative techniques at any time of year. We intend to include ideas from Computational Creativity among these resources, in the hope that it will encourage people to think of these ideas as being as essential as any algorithm for making things that make things.

Next year we hope to run some analysis on the entrants to the jam, primarily through optional surveys. This will help us get an idea of the jam's makeup, and people's motivations for entering the event. We are concerned that the lack of evaluation will make the jam complicated for people to curate and explore afterwards, and also acknowledge that some people will be interested in being rated by their peers. We are still reviewing our decision to remove rating altogether from the jam - we may make alterations in 2015 to improve filtering and curation, although it is still unlikely we will implement global ratings that declare overall winners, as we felt the lack of rating contributed a lot to the jam's informal atmosphere.

¹⁰<http://tinyurl.com/procjampcgamer>

¹¹<http://tinyurl.com/procjameurogamer>

¹²<http://tinyurl.com/aigameslecture>

Conclusions

In this paper we reported on the first procedural generation jam, or PROCJAM. The event was designed to create a new community around generative techniques for games and other software, with an emphasis experimentation, sharing of ideas and introducing new people to writing generative code. We described the changes we made to the classical game jam format to encourage more participants and make the event more accessible. We believe we were successful in this regard, but we also know that there is a lot of work left to be done in maximising the event's impact and accessibility, which we hope to address in future years. We then showed some illustrative examples from the 138 entries received, and discussed the potential for jams to impact communities close to Computational Creativity and potentially nurture relationships and collaborations between them.

Acknowledgements

Thanks to the reviewers which helped improve the paper, particularly the introductory segment on procedural generation. We would like to thank Azalea Raad for her help organising the event as well as the speakers, press and entrants who supported the jam. We wish to thank PROSECCO for kindly sponsoring the day of talks that launched the jam. Thanks also to Sophie Houlden, whose Fishing Jam inspired several of the changes to PROCJAM's format. This work was sponsored by EPSRC grant EP/L00206X.

References

- Betts, T. 2014. *An Investigation of the Digital Sublime in Video Game Production*. Ph.D. Dissertation, University of Huddersfield.
- Bytten Games. 2014. Lenna's inception. <http://lennasinception.com/>.
- Cerny, M. 2014. Machine understanding for interactive storytelling. <http://martin-cerny-ai.itcl.io/amuse>.
- Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the Third International Conference on Computational Creativity*, 77–81.
- Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI 2012 - 20th European Conference on Artificial Intelligence.*, 21–26.
- Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.
- Cook, M., and Colton, S. 2014. A rogue dream: Automatically generating meaningful content for games. In *Proceedings of the First Experimental AI in Games Workshop, AIIDE 2014*.
- Cook, M. 2014. If only we could talk to the generators. Talk at the Procedural Content Generation Summit at ITU Copenhagen.
- Gray, K.; Gabler, K.; Shodhan, S.; and Kucic, M. 2005. How to prototype a game in under 7 days. <http://tinyurl.com/7dayprototype>.

- Hecker, C. 2012. The depth jam. http://chrishecker.com/The_Depth_Jam.
- Interactive Data Visualization, I. 2002. Speedtree. <http://www.speedtree.com/>.
- Liapis, A.; Yannakakis, G.; and Togelius, J. 2012. Co-creating game content using an adaptive model of user taste. In *Proceedings of the Third International Conference on Computational Creativity*.
- Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22:211–226.
- Llano, M. T.; Hepworth, R.; Colton, S.; Charnley, J.; and Gow, J. 2014. Automating fictional ideation using conceptnet. *Proceedings of the AISB14 Symposium on Computational Creativity*.
- O'Connor, A. 2014. Time to admire a secret habitat. <http://www.rockpapershotgun.com/2014/12/17/secret-habitat-art-gallery-simulator>.
- Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. In Ventura, D.; Gervás, P.; Harrell, D. F.; Maher, M. L.; Pease, A.; and Wiggins, G., eds., *Proceedings of the Second International Conference on Computational Creativity*, 16–21.
- Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Trans. Comput. Intellig. and AI in Games*.
- Toy, M.; Wichman, G.; Arnold, K.; and Lane, J. 1980. Rogue.
- Yu, D. 2009. Spelunky.
- Zook, A., and Riedl, M. O. 2013. Game conceptualization and development processes in the global game jam. *Proceedings of the Foundations of Digital Games (FDG'13)*.
- Zucconi, A. 2014. Orbitalis. www.orbitalis.org.