Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans

Michael Cook and Simon Colton
Computational Creativity Group
Goldsmiths, University of London
http://ccg.gold.ac.uk

Abstract

We describe *ANGELINA-5*, software capable of creating simple three-dimensional games autonomously. To the best of our knowledge, this is the first system which creates complete games in 3D. We summarise the history of the ANGELINA project so far, describe the architecture of the latest version, and give details of its participation in *Ludum Dare*, a game design competition. This is the first time that a piece of software has entered a videogame design contest for human designers, and represents a step forward for automated videogame design and computational creativity.

Introduction

Videogame development is a highly complex creative task incorporating the production of music, art, animation, architecture, narrative, cinematography, rules and system design, amongst others. It is not merely the sum of all these creative acts either, but the result of such acts cooperating together to achieve a creative goal. It is fair to say that videogame development is one of the most creatively diverse mediums that Computational Creativity has available to study.

The games development community has grown rapidly over the last decade. The ubiquity of the Internet and the rise of digital distribution has allowed small developers to bypass traditional publisher routes to selling a game, and the spread of simple development tools and APIs such as Unity, Twine and Flixel has made it easier for people without a background in programming to develop games. This culture of rapid development, of shared learning experiences and the general popularisation of game development has led to game-making *jams* (competitions) playing an increasingly important role in allowing game developers of all levels to interact with and learn from one another. Their simple premise – a time-limited event where entrants develop a game from scratch according to a given theme – makes them ideal for newcomers who wish to work on something small-scale and simple. These features also make them ideal platforms for testing computationally creative software.

We describe here *ANGELINA-5*, henceforth *ANGELINA*, an automated game designer that creates 3D games and interactive experiences using Unity, a modern engine for game development. We give details of the system's implementation and how it differs from earlier versions. We also report on *ANGELINA*'s participation in Ludum Dare, a game design contest which drew 2064 entries in December 2013. We discuss *ANGELINA*'s performance, and the cultural response to its involvement in the contest.

The rest of the paper is organised as follows: in *Background* we give a brief introduction to the *ANGELINA* project and discuss the choice of Unity as a new platform for the system; in *Design Process* we describe the latest version of *ANGELINA*, and the challenges associated with building a game designer that works with modern 3D game design technology; in *Game Jams* we discuss game design contests such as Ludum Dare and their role in the culture of game development; we then discuss *ANGELINA*'s entry to the contest in *ANGELINA and Ludum Dare*. In *Related Work*, we summarise other approaches to building systems capable of designing videogames; in *Future Work* we outline a road map for *ANGELINA*; we then close with *Conclusions*.

Background

ANGELINA is a cooperative coevolutionary system for automating the process of videogame design. There have been several different versions of *ANGELINA* in the past (Cook and Colton 2011) (Cook and Colton 2012) (Cook, Colton, and Pease 2012), each tackling a different kind of game design problem, often on different platforms or game engines. The latest version of the system represents a large step forward and a large shift in the platform that *ANGELINA* is built upon. The research aims of the project are concerned with automated game design and the procedural creation of content, but also target issues in Computational Creativity. Later versions of *ANGELINA* investigated questions of thematic control, context and framing of design decisions, and also whether *ANGELINA* could discover new game mechanics with minimal game knowledge (Cook et al. 2013).

ANGELINA is built as an extension to the Unity game development environment (www.unity3d.com). Unity is an extremely popular, versatile and powerful game engine that ships with a comprehensive development environment that is also highly extensible. Unity games can be deployed to web browsers, all major desktop operating systems as native applications, every modern games console and handheld device, and most smartphone operating systems including iOS, Android and Blackberry. This versatility means that distribution of *ANGELINA*'s games is extremely simple, and are also distributable to a wide variety of people, hopefully increasing the success of future studies, as well as improving the dissemination of our results. Unity also supports both 3- and 2-dimensional game development, meaning that we can begin to investigate the automation of fully-3D game design.

Moving into the development of 3D games allows *ANGELINA* to explore a wider variety of game types, and also strengthens the image of *ANGELINA* as a game designer in terms of using contemporary technology, which is an important aspect of the project from a computational creativity perspective. It also allows us to improve on the design and structure of *ANGELINA* as a research tool: Unity's extensibility means that we can build *ANGELINA* as a series of modifications to the Unity tool itself. This means the system can have a full user interface, better visualisation and statistical analysis of the development process, and an easier platform on which to run experiments or integrate with other software. In terms of our project's focus, we also hope to use Unity's breadth as a platform to apply *ANGELINA* to design tasks on the spectrum between games and interactive artworks. Unity is used for a wide variety of projects besides traditional games, including interactive art installations such as *Canis Lupus*[1] and *Mothhead*[2]. We hope to make contributions to this spectrum also.

Game Jams

Structure A *game jam* is a co-ordinated event in which groups of people develop games in a fixed timeframe (commonly 48 hours), either alone or in groups. Some game jams are structured as contests, with judging, while others are organised for the self-improvement, to build communities of developers. Almost all game jams feature a theme which must be incorporated into the games designed for the event. These themes are used as creative aids, to focus people on a task or to make them explore unusual ideas.

Interpretation of the theme is often a crucial creative step in producing an interesting game, particularly when trying to distinguish an entry from potentially thousands of others. As an example, a game jam held in 2013 was run with the theme *Ten Seconds*. Entries to the jam included many games incorporating time limits of some kind, ten seconds in length. Here is a selection of alternative interpretations of the theme, used in games for the competition: the player controls an orphan asking for *seconds* of food; the player controls a *second*, someone who replaces someone else in a duel; the game records ten seconds of microphone input from the player, and procedurally converts it into a three-dimensional world to explore.

Role in Game Culture Game jams play a major role in the culture and community of game developers, particularly at independent and amateur level. In 2012, CompoHub[3] recorded a total of 134 game jams taking place, including *Ludum Dare*[4]. Ludum Dare is a thrice-annual event that takes place in April, August and December and has been running since 2002. Ludum Dare is split into two events which run in parallel – the *Competition Track* which is a 48-hour event in which solo developers make a game from scratch themselves, including any art and sound assets; and the *Jam Track* which is a 72-hour event in which the rules for the main competition are relaxed, allowing groups of developers to work together, and existing assets to be used. In December 2013, 2064 games were submitted.

After the submission period is over for Ludum Dare, a review period commences which lasts 22 days. During this period, anyone who submitted a game to the event in either track can enter ratings and leave comments on other submissions. On the main rating page, games are ordered based on a ratio of the number of ratings they have received versus the number of ratings they have given out, weighted so that this ratio is amplified at low numbers of ratings. This means that people who have submitted a game are encouraged to rate other games, since this is the fastest way of obtaining ratings for their own submission.

Reviews are broken down into eight categories: Fun, Overall, Audio, Mood, Innovation, Theme, Graphics, Humour. Note that Overall is a separate category, not an average of the other seven. Each category can be left unrated, or given a score between 1 and 5. Reviewers are encouraged to leave non-anonymous comments along with their reviews, but are not obliged to. At the end of the review period, the rankings are announced, including breakdowns per category, separated into the competition track and jam track.

Design Process

Predesign Phase

ANGELINA is given a word or phrase which acts as a theme for the game it is about to design. This method of starting a game design is derived from *game jams*, as described in the section *Background*. Examples of themes might be fairly straightforward, such as 'fishing', or more abstract, such as 'alone'. In some cases, the themes are intentionally unusual or restricting in order to stimulate creativity. For instance, the theme for the 2013 *Global Game Jam* was the sound of a heart beating. Developers are encouraged to incorporate the theme into their game in whichever way they can, such as through the ruleset, the narrative or the visuals.

When an input theme is given, if it is longer than a single word, *ANGELINA* will first attempt to isolate a single word most likely to be a suitable theme. Single words work better than phrases for our current methods of media acquisition and framing, because many of these processes are based on querying web services that expect singular queries. However, it should be noted that this single word approach is not a long term solution, and better theme parsing is a point of future work. In order to choose a single word from a phrase, *ANGELINA* uses a frequency analysis against a large corpus of English text[5], in order to find the least common noun. This approach was developed by analysing 150 game jam themes by hand and running similar filters on them. We

[1] http://tinyurl.com/canlupus

[2] http://tinyurl.com/mothhead

[3] http://www.compohub.net

[4] http://www.ludumdare.com/compo

[5] http://www.kilgarriff.co.uk/bnc-readme.html

found that the most prominent theming information tended to be in more specific words, particularly, nouns. 'You are the villain' simplifies to *villain*, for instance, while 'End of the Universe' simplifies to *universe*. The exception to this rule is where the theme includes meta-references to the game itself, such as 'build the level you play' – here, the important information is contained within the phrase as a whole and can't easily be condensed into a single word.

Once *ANGELINA* has a theme word, it attempts to expand the theme using word association databases[6]. We plan to replace this technique with a more relevant topic association approach in future, but for most applications word association provides a reasonable set of words relating to the source theme word. These word associations are combined with the theme word to provide a list of possible words relating to the game's overall theme. For example, the theme word *secret* would lead to a list of words including *secret*, *spy* and *mystery*. A typical list of associations runs to about thirty words. These associations are then used to perform a series of multimedia searches, one for each association, in order to build a database of assets for use in theming the final game. *ANGELINA* downloads public domain fonts from DaFont[7], 3D models from TF3DM[8] and sound effects from FreeSound[9]. These media are archived as they are downloaded, so that they can be retrieved quickly if needed in the future.

ANGELINA generates a *zone plan* which defines a number of themed zones for use within the game design. A zone is a collection of a floor texture, a wall texture, a 3D model for use as scenery, and a sound effect. The sound effect and scenery model are both randomly selected from the media downloaded from the associations list. In order to select the texture, *ANGELINA* searches through a list of 622 tagged texture files for ones which are related to one or more of the association words. A relationship can be established in one of two ways: first, it can compare the associations with the filename or folder name of the textures, which are categorised roughly according to their type (such as 'clouds' or 'paper'). Secondly, it can call on a database of word associations mined using crowdsourcing via Twitter. *ANGELINA* regularly posts random untagged texture files to its Twitter account[10] and asks its followers to provide single words which they associate with the image. These are retrieved and recorded in a database file, and used as a secondary means to relate associations to textures in the case that the filename match fails. Reply counts for a single tweet range from single replies to a dozen or more, and so far 901 responses have been recorded for 84 textures. If no matches are found through either method, *ANGELINA* selects textures randomly for the zones.

Once *ANGELINA* has selected two textures and randomly chosen a 3D model to act as scenery (we describe scenery later) and a sound effect for each zone, the zone map is complete. Before it proceeds to the main design phase, *ANGELINA* will generate a title for the game, and select a piece of music. The game's title is generated using a rhyming dictionary[11] and a corpus of popular culture references, including famous examples of media such as music and books collated from Top 1000 lists such as IMDB's Top 250 Movies[12], as well as idioms and common sayings. *ANGELINA* attempts to create puns using these resources and the list of source word associations, using a similar approach to the one described in (Cook, Colton, and Pease 2012).

To select a piece of music, *ANGELINA* attempts to choose a suitable mood for the game. It first takes the main theme word, and passes it to Metaphor Magnet[13] (Veale 2012) to obtain feelings people express in relation to the theme word. Metaphor Magnet is a tool for exploring a space of metaphors, mined from Google N-Grams. It has an array of features that are built on top of this concept, including the ability to show feelings people commonly express about a topic, such as poetic or metaphorical qualities of something, with the knowledge that these feelings are backed up by concrete examples in the N-Gram corpus.

As an illustration, if we submit the word *winter* to Metaphor Magnet, we are presented with a number of possible metaphors for winter, such as a 'frightening night' or a 'refreshing spring'. By selecting one of these, *ANGELINA* can use words which express feelings that Metaphor Magnet has corpus evidence for - e.g., winter in the context of a frightening night is commonly described as 'frightening'. This word is chosen as the base mood for the music for the game. It now has to relate this emotion to a piece of music. The music database *ANGELINA* currently uses is Incompetech[14], which categorises pieces according to twenty different moods. In order to relate the mood discovered through Metaphor Magnet with an appropriate tagged mood in Incompetech, we use DisCo[15] to rate the semantic similarity between each of the twenty known emotions and the one discovered emotion. The most similar emotion is used as the search mood for music, and a piece of music is randomly selected from the resulting pieces.

In total, *ANGELINA* uses fifteen web services or APIs during the predesign phase, from linguistic tools to databases of tagged content. In (Pease et al. 2013) the authors discuss the concept of serendipity in the context of creative software, and they note in relation to web services that "we believe this [accessing web services] will increase the likelihood of chance encounters occurring, [and] expect serendipity to follow". Note that the web services *ANGELINA* interacts with include unconstrained data sources such as Twitter as well as unedited automatically scraped databases such as Metaphor Magnet. This means that the results of the combinations of services are hard to predict, which offers a strong force of *chance*, one of the three dimensions of serendipity highlighted in (Pease et al. 2013).

[6] http://wordassociations.net

[7] http://www.dafont.com/

[8] http://tf3dm.com/

[9] http://freesound.org/

[10] twitter.com/angelinasgames

[11] http://www.wikirhymer.com

[12] http://www.imdb.com/chart/top

[13] http://ngrams.ucd.ie/metaphor-magnet-acl/

[14] http://www.incompetech.org

[15] http://www.linguatools.de/disco/disco_en.html

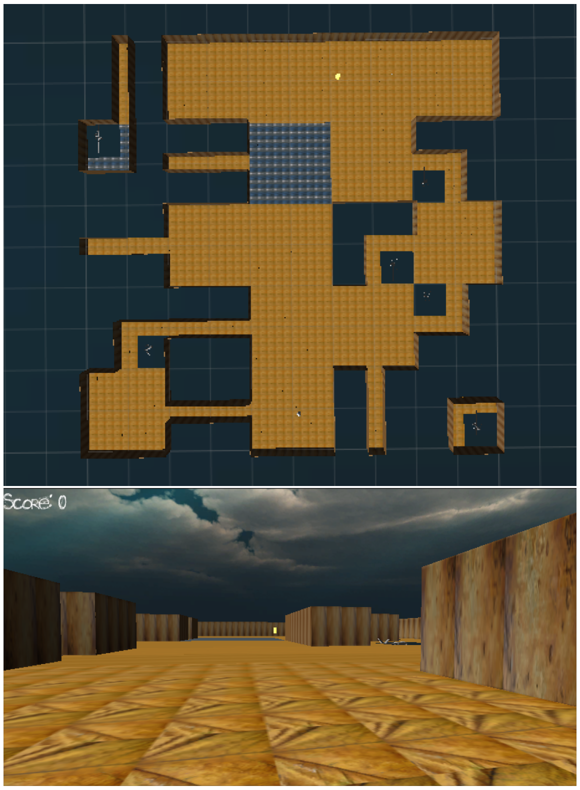


Figure 1: Screenshots of *Hit The Bulls-Spy*, a game designed by *ANGELINA-*. *Top:* The game world as viewed from above in the Unity editor. *Bottom:* A screenshot from the running game.

Design Phase

As with *ANGELINA-3* described in (Cook, Colton, and Pease 2012), *ANGELINA* is composed of several evolutionary systems that work in tandem to cooperatively evolve a game design. Each evolutionary system has two aspects to its fitness function: internal, objective rules that are considered to be unchanging regardless of the overall game design, and external, subjective rules that take into account what properties the current most fit game design has to adjust its fitness evaluation accordingly. In order to evaluate these subjective rules for a given member of a population, *ANGELINA* takes the most fit example from every other evolutionary process, combines them together to form a game, and then simulates playing that game in real-time. Currently, this simulation is very basic – *ANGELINA* will attempt to guide the player object from the starting point to the level exit, if such a path exists, and records any rules which activate (as well as how often they activate) during the course of the pathfinding. This data is used in the evaluation of the game designs, as detailed below. For more details on cooperative coevolution, see (Potter and De Jong 2000). For more details on our specific use of cooperative coevolution in *ANGELINA*, including details on the applicability of cooperative coevolution to multifaceted design problems, see (Cook and Colton 2011) and (Cook and Colton 2012).

There are currently four separate evolutionary processes:

- Level Design – which forms a basic layout of solid space in the game world. The top image in Figure 1 shows a birds-eye view of a level designed by *ANGELINA*. Level designs are currently built out of smaller tiles which are selected from a library of hand-designed tiles and arranged into a variable-size array. For instance, in Figure 1, the size of the map is five tiles wide by five tiles high. A tile is a ten by ten array of integers denoting solid ground, empty space or scenery. Scenery regions are impassable to the player, and when the game is exported, they are replaced with large, static 3D models for theming purposes.
- Zoning – which describes the visual and aural qualities of different regions of the game world. Zones are defined in the predesign phase, and during evolution a *zone map* is evolved, which is an array of integers relating each tile in the Level Design to one of the premade zones.
- Placement – which describes the start position of the player, and the position of the level exit. The primary objective in all of *ANGELINA*'s games is to reach the exit. In addition, a Placement defines the number and starting position of the game's *entities*. Entities are objects which are placed in the game world and given code to execute to play a role in the game's rules. A Placement contains a list of starting positions for each type of entity – currently all games by *ANGELINA* include exactly two entity types, the purpose of which is defined by the Ruleset.
- Ruleset – which describes the set of *behaviours* possessed by each entity. In Unity, 'behaviour' is an overloaded term used to describe any piece of code which implements a particular interface. In the current version of *ANGELINA*, we have supplied a stock of behaviours which can be attached to the entities in *ANGELINA*'s games to form a basic ruleset. These behaviours include providing motion for the entity (such as random walks, or wall following) and adding mechanical rules (such as killing a player, or providing score when collected). Expanding this set with automatically generated code is a point of future work, see (Cook et al. 2013) for details.

Each of these four processes evolve their populations in isolation, according to various fitness criteria, normally expressed as parameters which can be easily varied, so as to give *ANGELINA* the ability to alter its own fitness functions in the future. Currently, all parameters have been set through experimentation to find values which produce an interesting variety of outputs in such terms as maze style variation (a mix of open spaces as well as some labyrinthine designs too) or level layouts (dense and sparse entity placement, varying approaches to extending the distance between start and exit). The fitness criteria are as follows:

- Level Designs are selected to maximise the size of the largest contiguous island, whilst simultaneously avoiding overfitting by limiting fitness to a maximum island size. This encourages level designs in which the tiles join up to form a single level space, but avoids the situation where the entire level is one open expanse by penalising levels which are too full of solid tiles. A level design is penalised if the player or exit start position is in empty space.

- Zone Maps are selected to maximise connectedness in zones of the same type. This means that a zone map which has two Zone 1 zones separated by a Zone 2 zone scores lower than a zone map which has a single contiguous Zone 1 zone and another single Zone 2 zone. This is done to provide consistency in when and how often a zone is encountered by the player. We anticipate this will become more important as *ANGELINA* develops, as zones will define clearly themed areas such as a forest, and having these frequently broken up by other zones would be disorienting and may reduce immersion for the player.
- Placements are selected to maximise spread of entity placements across the map, but are penalised for any placements, including player or exit placements, which are not on solid ground. Placements are also selected to maximise the distance of the path from the start position to the exit position, with a penalty if no such path exists.
- Rulesets are selected to maximise the number of rules fired in a simulation of a game. *ANGELINA* records which rules fire during an execution of the game, using a simple player controller which attempts to follow a direct path to the exit. Rulesets are penalised if there is no way for the player to gain score or die, but does not guarantee both score gain and death are in the game.

It should be noted that many of these fitness criteria are in place only to complete *ANGELINA* as a game design system, particularly Rulesets and Zone Maps. We intend to replace these by giving the system the ability to create its own fitness criteria. These might therefore be considered baseline criteria for producing a complete game design.

A typical setup for *ANGELINA* consists of a population size of 30 for each of the four evolutionary species, and a run of 40 generations for the system as a whole, meaning that each species undergoes 40 generations of evolution itself. We utilise one-point crossover and single-element mutation for all four species, since representation is almost entirely array-based. Selection is elitist, and we carry forward the parents of the previous generation, something which we found useful in previous versions of *ANGELINA*, due to the volatile nature of cooperative coevolutionary systems.

Postdesign Phase

When *ANGELINA* has completed the set number of generations and completed a game design, the game export process begins. Unity games are meant to be developed inside a single project which contains all the art and audio assets for the game, the data, the levels, the code and logic. Unity has export features that compile these various components together into a single package for a chosen platform (such as iOS). However, in our case it is *ANGELINA* that is the Unity project, not any single game that it develops. This means that the asset folders contain databases of models used in the past, music that has been downloaded, metadata and information about *ANGELINA* as a system, and so on. Exporting the games as-is is therefore not possible, as Unity cannot be told to avoid exporting certain resources, and would attempt to export gigabytes of data for each small game developed.

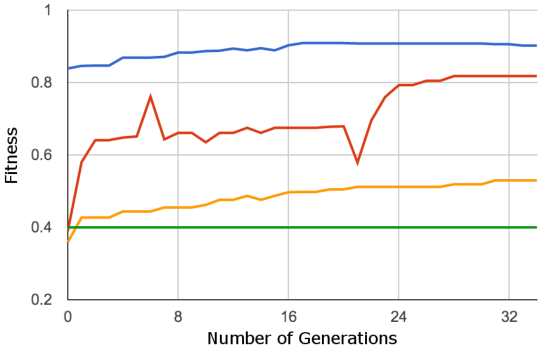


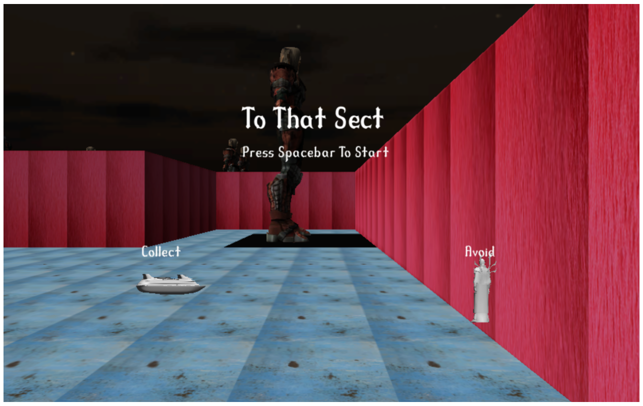
Figure 2: A graph showing the highest fitness as generations pass, for a single run of ANGELINA. The blue is Zone Map fitness; the red is Placement fitness; the yellow is Level fitness; and the green is Ruleset fitness.

For this reason, and because of a desire to archive games designed by the system, we have *ANGELINA* export all the relevant information about a game design into a separate folder. This includes a text file describing the level design and the locations of resources, as well as the asset files such as models and textures. This folder can then be read as a standalone Unity project that only imports the necessary resources, and can then export executable game binaries.

In addition to the game export, *ANGELINA* also produces a commentary describing some of the decisions it made in the production of the game, using template paragraphs which are filled in using resources it finds on the Internet, and data from the game's production. Previous versions of *ANGELINA* also used commentaries, as per (Cook, Colton, and Pease 2012). Figure 3 shows a sample commentary.

Evolutionary Performance

Figure 2 shows a sample fitness graph for each of the four evolutionary species that make up *ANGELINA*. The coloured lines are described in the caption to the figure. Note that there is little evolutionary improvement in the Zone Map or Ruleset species – these species are underdeveloped in the current version of *ANGELINA*. The system will eventually be able to track information about player routes through levels and use this to guide the placement of zones so that they affect the player's experience in a particular way, such as matching it against the emotional valence of a narrative, or to reflect changes in location. Similarly, the Ruleset species is awaiting an extension of work done on generating game mechanics through code (Cook et al. 2013) so that *ANGELINA* can propose rules itself which it can then use in a game design. Until then these evolutionary species remain incomplete. However, in the Level and Placement design species, we can see more clearly that evolution is working as intended. We anticipate that the other species will behave in this way, as they are integrated more fully into the cooperative coevolution.



This is a game about a disgruntled child. A founder. The game only has one level, and the objective is to reach the exit. Along the way, you must avoid the Tomb as they kill you, and collect the Ship.
I use some sound effects from FreeSound, like the sound of Ship. Using Google and a tool called Metaphor Magnet, I discovered that people feel charmed by Founder sometimes. So I chose a unnerving piece of music to complement the game's mood.

Figure 3: Title screen and excerpted commentary.

ANGELINA and Ludum Dare 28

The Ludum Dare 28 game jam took place on the weekend of December 13th 2013, following a week of voting which narrowed down a list of 100 themes to a shortlist of 20, and a final announcement of the winning theme at the moment the game jam started. The chosen theme was the phrase *You Only Get One*. It generated 1284 entries to the competition track, and 780 entries to the jam track.

ANGELINA entered Ludum Dare with two entries. In both cases, the system was given the theme in plain text, and configured to run for 60 generations, with a level population size of 35, a placement population size of 35, a ruleset population size of 20, and a zone population size of 15. Both games took approximately three hours to generate in their entirety, including the retrieval of game assets from the web.

The motivation behind producing two games for the jam was to investigate the presence of bias in the assessment of creative software in the medium of videogames. Our hypothesis was that, contrary to anecdotal reports and studies from Computational Creativity researchers e.g. (Pease and Colton 2011) and (Moffat and Kelly 2006), people tended to be *positively* biased towards creative software working in videogames. We submitted the first game ANGELINA produced with a commentary explaining the background of the system, and an unabridged commentary from ANGELINA about the game[16]. To anonymise the entry, the second game was submitted under a pseudonym to the game jam, without any reference to ANGELINA or the research project, and with ANGELINA's commentary edited to avoid references to software or other phrasing that might give away the game's background.[17]

Entries

To That Sect *ANGELINA*'s first game, and the one which was submitted with full disclosure, was titled *To That Sect*. Figure 3 shows a screenshot from the game. The player must avoid strange demonic statues while collecting ships, on their way to reaching the exit. An unsettling piece of music plays, and a ship's bell tolls in the background. The scenery chosen for the game is a model of a player character from the game Lineage 2, dressed in armour. In both this game and *Stretch Bouquet Point* below, *ANGELINA* extracted the word 'one' from the input theme as the most likely theme word, but then found it to be too general to use as a specific theme, and so chose to use the narrowing technique we described earlier to select a word associated with 'one' as the target theme. In the case of To That Sect, it chose the word *founder*. Words associated with 'founder' included religion and sect, which accounts for the references in the game's title as well as the musical choice. Metaphor Magnet suggested that people feel charmed by founders – presumably relating to the context of a cult or a religious sect – and *ANGELINA* narrowed this emotion down to 'unnerving' using DISCO. The references to ship are due to an ambiguation of the theme word – since a ship can *founder* on rocks, as a verb.

Stretch Bouquet Point This game was submitted anonymously under a different username, without any references to software or ANGELINA in the description, and an edited commentary to hide similar references in ANGELINA's output text. The player must avoid girls referred to as 'daughters' while trying to reach the exit. An untextured model of a woman is used as scenery, and very loud chanting plays over the top of the game's music, drowning it out.

As with the previous game, 'one' is further narrowed due to it being deemed an insufficient theme. This time, 'bridesmaid' is chosen as the target word, as it was found to be associated with the word 'one'. This leads to words such as bouquet, found in the title, as well as woman and daughter. The chanting that plays over the top of the game is from the keyword 'marriage' – a recording of an African griot singing during a marriage ceremony. The connection of 'bridesmaid' to 'one' is not obvious. Many of the results from basic word association rely on words appearing in proximity to one another, and 'one' is a very generic word which may lead to erroneous or weak connections being made. Improving the association step is a point of future work.

Results

The scores for both games for each of the eight categories are listed in Table 1. Votes are not made public in Ludum Dare, and we were unable to obtain specific data from the organisers. Despite this, we can see that for many of the rating categories, the game which was publicly labelled as being created by a piece of software was ranked higher in all categories except humour – hundreds of places in some cases. For humour, we believe the sole reason the anonymised game was ranked higher was because the (unintentional) surreality of the games was perceived as funny when it was believed to be coming from a person rather than software.

[16] This game can be viewed at http://tinyurl.com/tothatsect
[17] This game can be viewed at http://tinyurl.com/stretchpoint

	To That Sect	Stretch Bouquet Point
Overall	500	551
Fun	515	543
Audio	211	444
Graphics	441	520
Mood	180	479
Innovation	282	525
Theme	533	545
Humour	403	318

Table 1: Rankings for *ANGELINA*'s two games entered into Ludum Dare 28. There were 780 total submissions to this track. Lower rankings are better.

In order to try and maintain equal prominence for the two submissions, we rated an equal amount of Ludum Dare submissions whilst logged in as each account. To avoid both games rising to the top of the rating system at the same time and risking identification, we performed rating sessions at least 24 hours apart and at different times of the day, to minimise the risk that the same reviewer would encounter both submissions. In order to minimise the impact of our experiment on the event as a whole, we ensured that no game was rated twice, and we did not leave any written comments when rating other entries.

While the results indicate some potential positive bias towards the non-anonymised entry to Ludum Dare, we were unable to obtain specific voting data from the event organisers, leaving us unable to calculate specific confidence values for the reviews. Nevertheless, it does act as a good foundation for further investigation to be done in this area. These results are further reinforced by the written comments left underneath each submission by reviewers.

Reviews for *To That Sect* largely balanced positive with negative remarks. No comments were universally negative, tempering any criticism with positivity: "Angelina seems really good at creating an atmosphere with both sound and visuals. But the game part of it seems a bit lacking still." "The game itself is too simple. It seem the AI got the mood, but not the [game]play." By contrast, comments on *Stretch Bouquet Point* were passive-aggressive or outright critical: "this was a rather annoying experience." "You made me feel something there. Don't make me put it into words though."

The response to *To That Sect* was not without bias. One comment on the game notes that "If it [had] added shooting at the statues that you must avoid and a [target] of ships you to collect, it would have been better. It felt like playing [an] 'art-message' type of game". We can contrast this with *LITH*,[18] a game entered into the competition by a human designer, where the player navigates a maze and collect bags of gold coins, while avoiding patrolling robots. While not exactly the same, the rules of *LITH* are very close to those of *To That Sect*: search for as many objects of a certain type as possible, while avoiding another object, then exit. *LITH* was entered in the same track as *ANGELINA*'s games, and ranked 95th Overall, 125th for Fun, and 274th for Theme.

None of the comments on *LITH* reference the game's rulesets in a critical way. Contrary to the comments that *To That Sect* felt like an 'art' game, one comment actually praises LITH for feeling 'old-school', a quite opposite compliment. The games are by no means identical: *LITH*'s level is more closed in to accentuate a feeling of claustrophobia, but the similarities are many. This analysis suggests a fundamental difference in how people evaluate a game when they have knowledge and when they have no knowledge of its designer and design process. We plan further experimentation to investigate this notion.

Although the results for Ludum Dare have an extremely long tail, it is still notable that *ANGELINA*'s entry outperforms many hundreds of other entries to the contest. Low ranking entries included games which had very passive gameplay mechanics (such as a game in which single bets are placed on extremely long non-interactive races) or games which were lacking in appropriate art and audio content (many games were lacking audio entirely, or used music or sound effects which clashed with the game's theme). While these are small differences, and this was not a large, conclusive study, it is nevertheless significant that *ANGELINA* was ranked, by a community of game developers, to have outperformed many other entrants.

Related Work

Procedurally generating specific types of content for videogames is a well-explored area of research (Togelius et al. 2011). Many different types of content have been generated automatically, from rulesets (Togelius and Schmidhuber 2008) to levels (Williams-King et al. 2012) to art assets (Liapis et al. 2013) and even procedural generators themselves (Kerssemakers et al. 2012).

More specifically, the creation of software to automate the process of game design has been looked at by others in the past. In (Treanor et al. 2012) the authors describe the Game-o-Matic, a design assistant for journalists that could be given a graph representing relationships between concepts (such as *police arrests protester*) and then construct a game that reflected the network of relationships. The Game-o-Matic only understood a limited set of verb relations, and sourced its initial rulesets from a library of human-authored rules. However, it was able to source artwork for its games automatically, and could tweak rules to refine a game design, which gave it a good expressive range.

In (Nelson and Mateas 2007), the authors present a simple mini-game generation system that takes *verb-noun* constructions and presents games based on the given relationship. The input *shoot pheasant*, for example, presents games where the player controls a crosshair trying to shoot birds, or controls a bird trying to avoid being shot. Connections are made between human-tagged game mechanics and known words using a combination of ConceptNet and WordNet.

ANGELINA is not the first piece of creative software to engage with people in a social or cultural context. The Painting Fool, a piece of software its designer hopes will one day be taken seriously as an artist, has exhibited its work in public fora multiple times, e.g. (Colton and Pérez-Ferrer 2012), and has sold its artworks to collectors. Elsewhere, Ventura's PIERRE system (Morris et al. 2012) evolved soup recipes using a database of existing recipes and an understanding

[18] *LITH* game: www.tinyurl.com/lith-ludum

of food groups. PIERRE's recipes were evaluated anonymously in online cookery forums, as well as having its creations cooked by a person and evaluated via tasting on multiple occasions, with the knowledge of the recipe's origin in these latter cases. Anecdotal evidence suggested positive bias where the consumers had knowledge of PIERRE's existence, however we do not present this as serious evidence for positive bias, as the author notes that the presentation of the recipes may have contributed to the negative response to the anonymised recipe submissions.

Future Work

The work described here represents a new foundation for our research into automated game design. The flexibility of Unity as a platform, and the more general architecture of *ANGELINA*, means that we hopefully will be able to work on a single piece of software for some time, and go deeper into some of the issues we have brushed up against over the past few versions of the software. In particular, the following areas present themselves to us for further study.

- **Improved Communication** Entering *ANGELINA* in a game jam underlined the importance of the use of commentaries and context in conveying the intelligence and creativity of a system to an observer. For further exploration of the role of the observer in the context of *ANGELINA*'s entry to Ludum Dare, see (Cook and Colton 2013). In the future, ANGELINA will provide interactive commentary material that can be interrogated in-game to provide more detailed information about the design process. We believe this will ultimately increase the perception that the software is creative.
- **Innovation in Design** Because of the preliminary nature of some elements of *ANGELINA*, the game's main gameplay and objectives varied very little between different runs of the system. In order to improve this, we aim to bring in previous work on generating code for the invention of game mechanics as described in (Cook et al. 2013), and expand this to allow *ANGELINA* to generate code that produces new types of gameplay, and new styles of game. This will help strengthen the argument that ANGELINA is designing new games, and will also increase the independence of the system.
- **Better Theme Interpretation** A key aspect of entering a game jam is interpreting the given theme and working it into the final game design. We aim to integrate the theme into more aspects of the game's design than just the visual and aural theming. Good games incorporate the theme into their mechanics and design. We have discussed methods for doing this previously in (Cook and Colton 2013), and we will look to build some of them into *ANGELINA*.

Conclusions

We have described *ANGELINA*, the latest iteration of our automated game design system. *ANGELINA* is a redevelopment of the system in the Unity game engine, the first automated game designer that we know of to produce output in 3D. *ANGELINA* was developed to take a different approach to previous versions of the software, in that it would work from arbitrary phrases acting as themes. This allowed the software to take part in a game jam – the first time an automated game designer has done so, gaining a higher ranking than hundreds of other human-authored games.

We described the process of entering a game jam, as well as describing the system's two entries into the jam – one of which was publicly annotated as being developed by *ANGELINA*, while the other was anonymously submitted. We looked at the different reactions, both in terms of the scores the games received and the surrounding commentary on the games, and discussed the potential implications for creative software acting in the videogames medium in the future.

For all the mixed reactions and ratings, the response to *ANGELINA* entering a game jam was overwhelmingly positive, and the interaction with the development community will benefit us as researchers as well as the project in the long run. Hopefully we will see this trend continue, and we aim for more interaction between *ANGELINA* and the community in the future.

Acknowledgements

The authors wish to thank the reviewers for their comments which helped improve the paper, as well as Mike Kasprzak, Phil Hassey, Seth Robinson and Mike Hommel. This project has been supported by EPSRC grant EP/L00206X/1.

References

Colton, S., and Pérez-Ferrer, B. 2012. No photos harmed/growing paths from seed - an exhibition. In *Proceedings of the Non-Photorealistic Animation and Rendering Symposium*.

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: Observer issues in computational creativity. In *Proceedings of the 7th AISB Symposium on Computing and Philosophy (forthcoming)*.

Cook, M., and Colton, S. 2011. Multi-faceted evolution of simple arcade games. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Cook, M., and Colton, S. 2012. Initial results from co-operative co-evolution for automated platformer design. In *Proceedings of the Applications of Evolutionary Computation*.

Cook, M., and Colton, S. 2013. From mechanics to meaning and back again: Exploring techniques for the contextualisation of code. In *Proceedings of the AI & Game Aesthetics Workshop at AIIDE*.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of 16th European Conference on the Applications of Evolutionary Computation*.

Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*.

Kerssemakers, M.; Tuxen, J.; Togelius, J.; and Yannakakis, G. N. 2012. A procedural procedural level generator generator. In *IEEE Conference on Computational Intelligence and Games*.

Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Moffat, D., and Kelly, M. 2006. An investigation into peoples bias against computational creativity in music composition. In *Proceedings of Third Joint Workshop on Computational Creativity*.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the Third International Conference on Computational Creativity*.

Nelson, M. J., and Mateas, M. 2007. Towards automated game design. In *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence*.

Pease, A., and Colton, S. 2011. On impact and evaluation in Computational Creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. 2013. A discussion on serendipity in creative systems. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Potter, M. A., and De Jong, K. A. 2000. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computing* 8(1).

Togelius, J., and Schmidhuber, J. 2008. An experiment in automatic game design. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

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*.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the Third Workshop on Procedural Content Generation in Games*.

Veale, T. 2012. From conceptual "mash-ups" to "bad-ass" blends: A robust computational model of conceptual blending. In *Proceedings of the Third International Conference on Computational Creativity*.

Williams-King, D.; Denzinger, J.; Aycock, J.; and Stephenson, B. 2012. The gold standard: Automatically generating puzzle game levels. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.