Artefacts: Minecraft meets Collaborative Interactive Evolution

Cristinel Patrascu
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
Email: patrascu.cristinel@gmail.com

Sebastian Risi
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
Email: sebr@itu.dk

***Abstract*—Procedural content generation has shown promise in a variety of different games. In this paper we introduce a new kind of game, called *Artefacts*, that combines a sandbox-like environment akin to Minecraft with the ability to interactively evolve unique three-dimensional building blocks. Artefacts does not only allow players to collaborate by building larger structures from evolved objects but also to continue evolution of others' artefacts. Results from playtests on three different game iterations indicate that players generally enjoy playing the game and are able to discover a wide variety of different 3D objects. Morever, while there is no explicit goal in Artefacts, the sandbox environment together with the ability to evolve unique shapes does allow for some interesting gameplay to emerge.**

I. Introduction

In recent years there has been a growing interest in *procedural content generation* (PCG). This field includes algorithms and methods for generating a wide variety of different types of content (e.g. levels, three-dimensional objects, textures, stories, 3D caves etc.) that can be part of the virtual world of a video game [4, 5, 8, 10, 12, 19, 25]. One advantage of automatically generating game content is the reduced amount of work required by artists and game designers. Besides production cost reduction, games have also benefited from the novel gameplay emerging from PCG techniques [19]. Additionally, PCG can increase a game's replay value because content is constantly updated and varied throughout different play sessions.

A main inspiration for the game presented here is Minecraft[1], which is a sandbox video game that allows players to build three-dimensional structures together with others from a selection of predefined cubes made out of different materials (e.g. stone, wood). Minecraft encourages players to play creatively by giving them a variety of different ways to play the game. While the cubes are predefined, Minecraft does employ a PCG-based approach to generate the 3D worlds for the players to explore.

In the new game presented here, called *Artefacts*, players can collaboratively build 3D structures in a sandbox environment similarly to Minecraft. However, in contrast to Minecraft, in which players only have a predefined number of cubes to chose from that all have about the same shape, Artefacts allows players to create an unlimited variety of differently shaped 3D building blocks through an evolutionary computation (EC) approach. EC methods in particular have proven effective at automatically generating diverse content for games such as weapons in Galactic Arms Race (GAR [7]), levels for a competitive multiplayer FPS game [13], flowers in the social video game Petalz [16], or even complete games [5, 24].

[1]Copyright (c) 2011 Mojang

Fig. 1. **Artefacts - The Video Game.** Players in Artefacts can collaboratively evolve unique 3D objects in an open physics sandbox and combine them to build larger structures.

The 3D objects in Artefacts are genetically encoded by a special kind of neural network called a *compositional pattern producing network* (CPPN; [3, 20]). The generative CPPN encoding enables players to breed an unlimited variety of different 3D objects with regularities such as symmetry or repetition. Importantly, the NEAT algorithm [21], which evolves the CPPNs in this paper, allows the 3D objects to become increasingly complex and more intricate over generations.

Players in Artefacts can guide evolution by choosing from a set of artefact seeds that spawn around a planted object. Importantly, players can collaborate in the breeding process by picking up seeds produced by others and continuing evolution from there. Moreover, players are able to manipulate the placement of objects in three-dimensional space and can express their creativity by building a wide variety of different structures using the evolved artefacts.

There are no explicit goals in Artefacts. The game was designed to encourage players to explore and to play with the ability to evolve and build with 3D objects, which means that players can use the artefacts in any way they see fit. For example, players can focus on building tall structures or on destroying other peoples' structures.

To investigate what type of new game affordances Artefacts offers, both quantitative and qualitative data from a series of playtests were collected and analyzed. The results from the initial playtests suggest that, while still in an early stage, the novel combination of evolved 3D objects in an open world is a promising game concept that offers many potential directions to expand upon.

II. Background

This section first discusses existing work combining PCG with video games and concludes by reviewing the technical building blocks of the PCG algorithm employed in Artefacts.

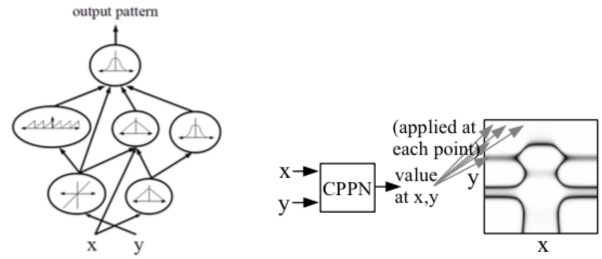
A. *Procedural Content Generation*

When applied to games, PCG allows game elements (e.g. maps, textures, items, quests, etc.) to be generated algorithmically rather than through direct human design [5, 8, 25]. For example, the popular Diablo series[2] features procedurally generated dungeons that players explore as a central focus of the game. Like Diablo, many other PCG approaches similarly rely on a fixed set of parameters and randomness to generate content within a heavily constrained space of possibilities. However, a recent focus is to apply artificial intelligence approaches to enable more open-ended generation of PCG.

In particular, evolutionary computation and other search-based approaches [25] can limit the need for hand-designed rules, and may thus further save on PCG development costs. More interestingly, it also enables design of new content outside the scope of a fixed space of rules. One popular technique is interactive evolutionary computation (IEC [23]), in which the user in effect guides an evolutionary algorithm. An example of IEC applied to video games is provided by NeuroEvolving Robotic Operatives (NERO [22]), in which players guide the evolution of a team of fighting robots. In another example, called Galactic Arms Race (GAR [7]), weapons are evolved automatically based on user behavior, and in the social Petalz video game, players can evolve an unlimited variety of different flowers [16]. Further examples include Avery et al. [1], who evolved several aspects of a tower defense game, Shaker et al. [18] who evolved levels for the platform game Super Mario Bros, Olsted et al. [13] who interactively evolved levels for a competitive multiplayer FPS game, and Togelius and Schmidhuber [24], who experimented with evolving the rules of the game itself.

The particular evolutionary representation that is applied to represent evolved 3D objects in Artefacts, is reviewed next.

[2]Copyright Blizzard Entertainment, http://blizzard.com/



(a) CPPN (b) CPPN to Image

Fig. 2. **Compositional Pattern Producing Networks for 2D Images.** (a) CPPNs can use a variety of different functions like sigmoids, Gaussian, sine and many others in contrast to more traditional ANNs with sigmoid activation functions (b) The CPPN example in this figure inputs two arguments x and y that are interpreted as coordinates in two-dimensional space. Applying the CPPN to all the coordinates and drawing them with an ink intensity determined by its output results in a two-dimensional image.

B. *Compositional Pattern Producing Networks (CPPNs)*

The 3D objects in Artefacts are generated by a variation of artificial neural networks (ANNs), called *compositional pattern producing networks* (CPPNs [20]), which differ in their set of activation functions and how they are applied. While ANNs often contain only sigmoid or Gaussian activation functions, CPPNs can include both such functions and many others. The choice of CPPN functions can be biased toward specific patterns or regularities. Additionally, unlike typical ANNs, CPPNs are usually queried across a space of possible input patterns to represent a complete image or pattern. Specifically, CPPNs produce a phenotype that is a function of n dimensions, where n is the number of dimensions in physical space. For each coordinate in that space, its level of expression is an output of the function that encodes the phenotype. Figure 2 shows how a two-dimensional phenotype can be generated by a function of two parameters that is represented by a network of composed functions. CPPNs in effect encode patterns at infinite resolution and can be sampled at whatever resolution is desired.

Successful CPPN-based applications include Picbreeder [17], MaestroGenesis [9], EndlessForms [3], the Galactic Arms Race (GAR) video game [7], folded wire robots [15], and virtual soft-body robots [2]. Clune and Lipson [3] introduced a modification to the general CPPN representation to produce 3D objects, which is the basis for the object representation in Artefacts. It is described in detail in Section III-B.

C. *Neuroevolution of Augmenting Topologies (NEAT)*

Because CPPNs are ANNs, they can be evolved with the *Neuroevolution of Augmenting Topologies* (NEAT) algorithm [21], which is the standard neuroevolution algorithm for such purposes [6, 17, 20]. Neuroevolution in general has shown promise in a variety of different games [14].

NEAT begins with a population of simple neural networks or CPPNs and then *adds complexity* over generations by adding new nodes and connections through mutations. Novel topologies gradually accumulate, thereby allowing diverse and complex phenotype patterns to be represented. No limit is placed on the size to which topologies can grow. New structures are introduced incrementally as structural mutations occur, and only those structures survive that are found to be useful (traditionally through fitness evaluations and through player selection in this paper). In effect, then, NEAT searches for a compact, appropriate topology by incrementally increasing the complexity of existing structure. A complete overview of NEAT can be found in Stanley and Miikkulainen [21]. For evolving content, complexification means that content (e.g. 3D objects in Artefacts) can become more elaborate and intricate over generations.

III. Artefacts – The video game

Artefacts (publicly available at https://cristi.itch.io/artefacts) has been designed as an open world in which players can explore and interact with evolving objects. An important aspect of the game is the social multiplayer component, which allows players to collaborate in the evolution of the 3D objects but also – similarly to Minecraft – in the construction of larger structures. In other words, Artefacts is a construction game with a potentially infinite number and variety of resources. Players experience the game through a first-person perspective and can perform the standard first-person actions such as walking, running and jumping. The game aims to create an immersive experience in which the players feel as being part of the world they are creating.

A. Development and Multiplayer Framework

Players can easily host their own multiplayer games and play together with others in the same virtual space. The game and its multiplayer component were implemented using the Unity game engine[3] and its built-in networking framework. The CPPN implementation is based on UnityNEAT[4], which is a port of the C♯ implementation of NEAT, called SharpNEAT[5].

B. Generating 3D Artefact Objects

The algorithm to generate the 3D artefacts is based on the CPPN object representation introduced by Clune and Lipson [3]. Instead of CPPNs with two inputs that can generate two-dimensional images (Figure 2), CPPNs to generate 3D objects have three inputs x, y, and z. The algorithm works by (1) inputting the coordinates of each point p (e.g. x=1, y=3, z=2), of a three-dimensional voxel volume (e.g. a grid of $5 \times 5 \times 5$ voxels) into the CPPN, (2) activating the network, and (3) determining if the voxel at that particular position p should be filled if the CPPN output is higher than some threshold, or empty otherwise. The coordinate input values are normalized within the [-1, 1] range before being passed into the CPPN.

The voxel array outputted by the CPPN is processed by the Marching Cubes algorithm [11], which generates a 3D mesh representation that can be easily rendered by common graphics APIs. After the polygonal surface is determined, the algorithm calculates the normal for each of the vertices.

[3]https://unity3d.com/
[4]https://github.com/lordjesus/UnityNEAT
[5]http://sharpneat.sourceforge.net/

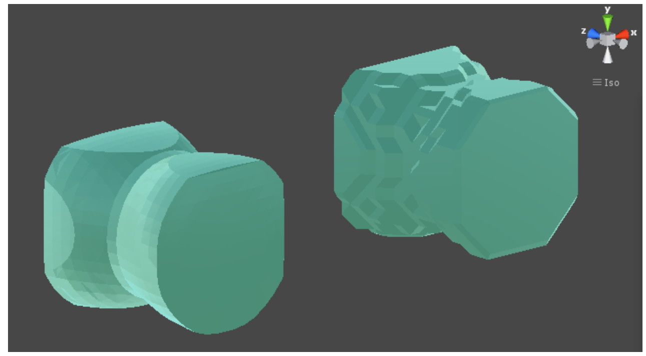


Fig. 3. **Artefact Generation.** 3D objects created with the representation introduced by Clune and Lipson [3] and with the "blockier" Artefacts modification (right).

(a) Inventory Selection (b) Planting and Positioning

Fig. 4. **User Interface.** (a) Players can store seeds in their inventory and plant them anywhere in the virtual world. (b) Players can also position and rotate the artefacts before they are planted.

The voxel volume size in Artefacts, which is set to $16 \times 16 \times 16$ units, tries to strike a balance between the level of detail of the generated 3D meshes and the time required by the Marching Cubes algorithm to create the mesh. As a bias towards rounded objects, the distance from the center of the workspace volume is given as an additional input to the CPPN.

In contrast to the approach by Clune and Lipson [3], the CPPN representation in this paper is slightly modified to create meshes with sharper edges that give the artefacts a "blockier" aesthetic. The CPPN output values are processed in the following way: (1) During the calculation of the output value for each coordinate, the algorithm keeps track of the minimum min and maximum max produced values. (2) The central value c between the minimum and maximum is calculated. (3) For each position p, a voxel is created if CPPN output $m \geq c$. In addition to the 3D mesh, the CPPN also determines RGB color values for each artefact through three additional outputs. Figure 3 shows an example of objects generated with the original representation (left) and the modified min/max representation (right).

C. Game Mechanics Overview

While exploring their environment, players can find and interact with artefacts of different shapes and colors evolved by themselves and other players. The user interface was created with the goal of making each available player action as intuitive as possible. Players have an inventory, which allows

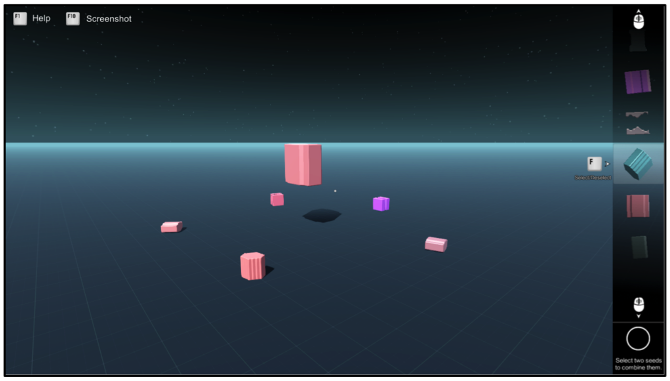


Fig. 5. **3D Artefact and Offspring.** When an artefact is planted it produces five smaller seed artefacts through mutation, which can be picked up by the players. Once planted, the seed produces a full-sized artefact. Mutations on the parent CPPN (e.g. adding new nodes and connections or changing the weight of a connection) create offspring that, while resembling the parent, vary in different ways. By selecting which seeds they prefer, players can guide evolution based on their personal taste.

them to store collected artefact seeds. They can select seeds by scrolling through them (Figure 4a) and plant selected seeds in the virtual world (Figure 4b). Planting a seed produces a full-sized artefact and spawns five offspring seeds surrounding the artefact (Figure 5). These offspring seeds are created by mutating the parent artefact, and while similar to their parents, they can vary in interesting ways. The created seeds can be picked up and planted by others, allowing multiple players to collaboratively influence the lineage of an evolving artefact. It is also possible to select two seeds from the inventory, thereby performing a crossover between them.

While planting artefacts, players have precise control over their position and rotation. By holding down the right mouse button, artefacts can be rotated about different axes through the keyboard, with a rotation speed of 100 degrees per second. Players can also move around while holding the artefact to position it anywhere in the virtual world. Once an artefact has been planted, it can be picked up and repositioned by other players. Players can also take screenshots of their creations, from which some are shown in the next section.

IV. Playtests and Iterative Development

While developing Artefacts, an iterative development approach was chosen. New features were added progressively, tested and evaluated based on player questionnaires. Especially the user interface (UI) went through many iterations. Besides the UI, the controls to interact with the artefacts also changed significantly together with the way different artefacts physically interacted with each other. In the following sections we present the three game iterations in chronological order together with the results of the player questionnaires. Participants were not given concrete instructions on how to play and were only encouraged to explore the game's affordances.

A. Experimental Parameters

The available CPNN activation functions were Linear, Bipolar sigmoid, Gaussian and Sine, all with equal probability of being added. Offspring had a 45% probability of weight mutation, 20% chance of node addition, 20% of adding a new connection, and a 15% probability of deleting a connection. The mutation probabilities were set to relatively high values to ensure that players see fast evolutionary progress while still producing offspring that resembles the parent artefacts.

TABLE I
Results of First Iteration Multiplayer Test

Total number of players	7
Number of sessions	4
Number of players in each session	5
Average duration per session (in minutes)	16
Average # artefacts planted per session	158
Average # mutations per session	113
Average # crossovers per session	45
Average # seeds picked up per session	227
Average # player contributing per artefact	2.3
Max # player contributing per artefact	6
Average # artefacts planted per players	29
Total # artefacts planted	633
Total # of spawned seeds	633 * 5 = 3,165
Total # of collected seeds	908
Max generation	54
Total # of mutations	453
Total # of crossovers	180

B. First Version

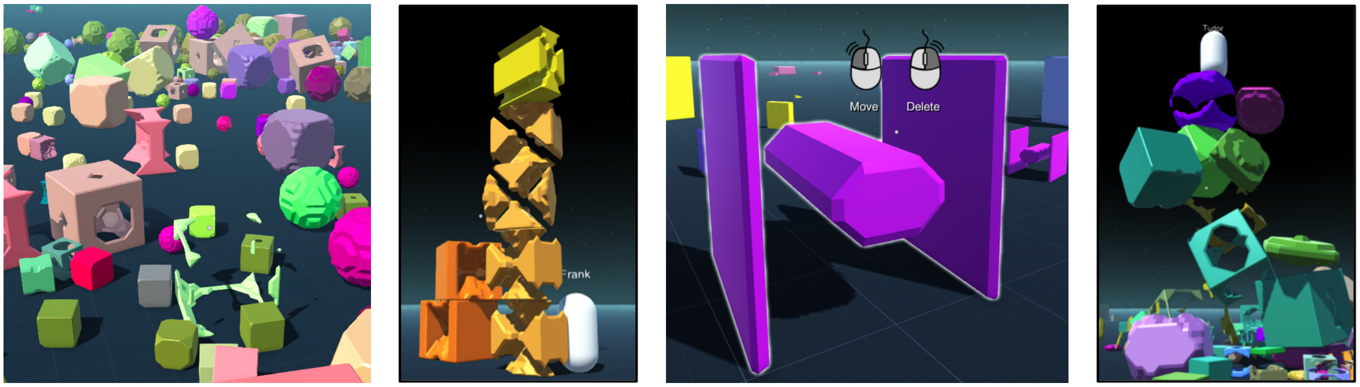
Seven people participated in the playtest of the first game version on site at the IT University of Copenhagen. However, due to technical limitations, only five players could play the game at the same time. In the first iteration of the game, all artefacts were controlled by rigid-body physics, i.e. they were affected by gravity and could collide with each other. The testers played for approximately one hour (divided into four separate sessions with five players each) and filled out a questionnaire afterwards.

A summary of the results is shown in Table I. Players planted a total of 633 artefacts and collected 908 seeds. Not surprisingly, players seemed to plant more artefacts in less time as they got accustomed to the game mechanics and user interface. Figures 6a,b show some of the evolved artefacts, which come in a variety of shapes and colors, and a tall structure that was built by multiple players.

Picking up seeds evolved by others allowed users to continue evolution and collaborate on the design of other players. Up to six players contributed to the lineages of some artefacts[6], with 2.3 players contributing on average per artefact. This suggests that the multiplayer component of the game allows meaningful interactions to emerge between players and the artefacts they create. Of the 3,165 spawned seeds, 908 were picked up by players, which is roughly 28%. The reason that players did not pick up every seed is likely due to the fact that (1) some of the produced offspring look similar to each other, and (2) players decide whether or not to pick up seeds based on their aesthetic preferences.

Interestingly, the placement of artefacts in the virtual world appears to form one or more clusters (i.e. a large number of

[6] While only five participants could play at the same time, players that left the game made room for others to join, making lineage contributions of more than five players possible.



(a) First Version (b) First Version (c) Second Version (d) Second Version

Fig. 6. **Artefacts Evolved by Players During the First and Second Version.** The CPPN-based representation allowed players to evolve a variety of different 3D objects (a). Players also tried to build taller structures together in the first version of the game (b), which proved quite difficult because it was not possible to permanently combine two artefacts. In the second version, players were able to glue artefacts together, thereby allowing the construction of a wider variety of different structures (c, d).

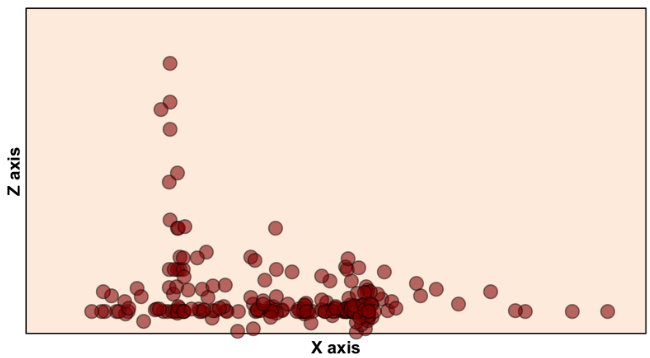


Fig. 7. **Artefact Placement.** The placement of the artefacts and observations during the playtest suggest that players often build structures together, thereby creating clusters of objects in the virtual world.

artefacts in a close distance from each other) and a number of artefacts spread across different directions (Figure 7). These results and observations from the playtest indicate that players often build structures with others or in their vicinity, instead of building structures by themselves in isolation.

1) Questionnaire Results: To analyze the players' subjective experience, they were asked to fill out a questionnaire after playing. A total of seven questionnaires were filled out. To characterize patterns in the players' responses, they were labeled with tags and then aggregated tags were created consisting of several related ones. The following is a list of tags for each of the answered questions from the first playtest:

- Most interesting part: interaction with other players (2), creating unique shapes (4), physical interactions (2), combining artefacts (1)
- Least interesting part: hard to build large structures (2), buggy interface (1), lack of more gameplay elements (2), structures getting destroyed by other players (1), interaction with other players (1)
- Could be better: having a way to easily connect (glue) artefacts (2), players flying around (1), buggy interface (1), more physical rules (3), more shapes in the beginning (1), having some predefined goals (1)
- About evolving artefacts: difficult to predict (1), should have more initial variation (1), intuitive (2), player felt curious (3), breeding seeds should be more visible (1)
- More interesting in multiplayer than it would be in single-player: yes (6), the same (1)
- About combining (breeding) artefact seeds: outcome was sometimes unexpected (2), interesting outcome (2), fun/cool (2), could be more elaborate (1)
- About building structures: difficult (3), objects are too light (1), not so intuitive (1), more building controls (1), physics was a limiting factor (1), could be better with different sized artefacts (1)

The questionnaire answers and observations during the playtest indicate that players enjoyed (1) creating unique artefacts, (2) the physical interactions between artefacts and (3) the element of an open world, in which one can play together with others. Most players thought that the process of planting seeds was intuitive and clear. However, some players found it difficult to understand how combining seeds worked while others reported that it was difficult to predict the result of mutations and crossover. Additionally, some players would have preferred more variation in the seeds that are initially created to populate the world.

While the ability to create unique artefacts in a physics sandbox allowed some emergent gameplay (e.g. building the tallest structure), composing more complex structures proved challenging; objects would tend to easily knock each other down while the players were trying to place them next to each other.

For example, building the stacked structure in Figure 6b proved to be a very difficult task because players needed to place artefacts with extreme precision for the structure not to collapse. Players had to create an additional supporting structure that allowed them to climb high enough to place more artefacts on top of the already existing structure. Furthermore, some players were frustrated by the fact that anyone can interact and therefore destroy someone's constructions.

C. Second Version

The first playtest provided valuable information about the players' experience and potential ways to improve it. In addition to minor bugfixes and interface improvements, more variation was added to the seeds initially present in the world by randomly evolving them for 10 up to 20 generations. Players now also had the ability to delete seeds and artefacts. The biggest change from the first version of the game was the added ability to attach or "glue" artefacts together by placing them so close to each other that they touch each other's bounding boxes. These modifications aimed to make it easier for the players to combine artefacts into larger and more complex structures.

Four participants that were new to the game took part in the second playtest. Because it focused on testing more specific game adjustments, performing two sessions (lasting 13 and 10 minutes) was deemed sufficient. Players evolved a total of 190 artefacts, 42% of those through crossover. Figure 6c,d show examples of structures built during the second user test: a large tower build by multiple players and a structure resembling a spaceship. While the new game modifications made it easier to build tall structures, the artefacts could still collide with each other, making it difficult to place them precisely next to each other.

1) Questionnaire Results: After playing the game, the participants were again asked two answer a questionnaire about their experience:

- Most interesting part: manipulating the evolution of artefacts (1), playing with other people (1), building structures (1), attaching artefacts to each other (1), variety of shapes (1)
- Least interesting part: attaching shapes was buggy (1), lack of more gameplay elements (1), artefacts do not evolve significantly enough (1), the flat plane environment (1)
- Could be better: hard to figure out how to combine seeds (1), more varied and complex shapes (4), attaching artefacts (1)
- About evolving artefacts: selection was counter-intuitive (1), felt repetitive (1), intuitive (1), artefacts look too much like boxes (1), player felt curious (1)
- More interesting in multiplayer than it would be in single-player: yes (3), equally interesting (1)
- About combining (breeding) artefact seeds: there should be more control over the outcome (2), some repeating archetypes (1), outcome could be more varied and complex (2)
- About building structures: many glitches (3), fun/cool (2), difficult (1), could be more interesting by having objects of different durability (1)

The answers from the questionnaire and observations during the playtest suggest that players enjoyed breeding artefacts and trying to control the direction in which they evolved. However, they felt that there could be more variation in the created artefacts. While the CPPN representation can produce different 3D objects, as shown in Figure 6, the volume in which the artefacts are generated in is always cube-shaped, resulting in many artefacts with flat sides that do not vary much in size. In the future it will be interesting to experiment with different 3D object encodings, allowing players to scale the artefacts, or to control the shape of the volume used to generate them.

In comparison to the first game iteration, the new modifications did in fact facilitate the construction of more complex structures. Players found it easy to attach artefacts together and to build on top of them. However, some issues remained that should be addressed to further enhance the experience of building structures. First of all, it was hard to align artefacts precisely with each other; artefacts move based on physical forces and synchronizing these physical simulations over the network was challenging. As a result, the artefacts could sometimes end up in a state in which the client-side objects failed to keep up with the server-side objects. Secondly, fitting artefacts together sometimes proved difficult; some had very different forms, not exactly fitting next to each other like the pre-made building blocks in games like Minecraft. Additionally, due to computational constrains, convex colliders were used on shapes that were concave, which meant that the colliders did often not match the exact shape of the object.

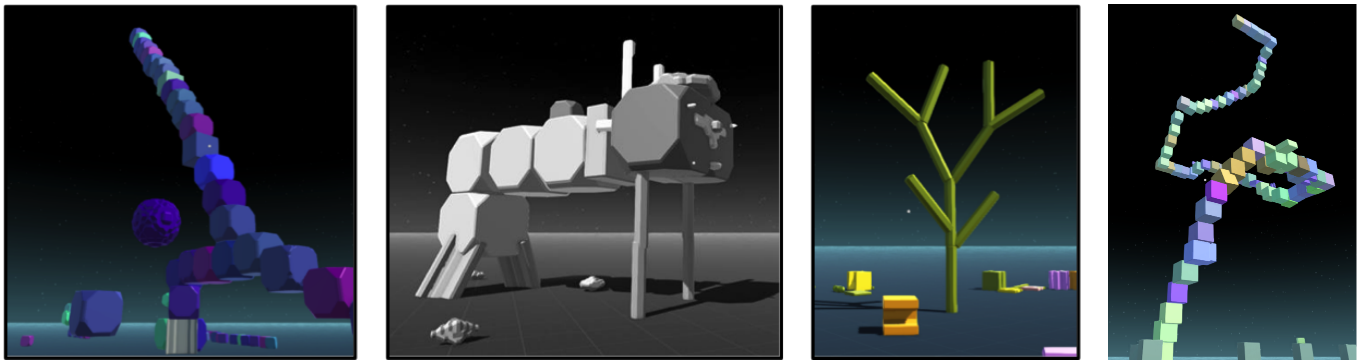
While some issues remained, the results from the second playtest suggested that the changes made after the first test did improve the players' experience. It also provided valuable information on how to further enhance the game experience.

D. Third Version

The final playtest took place online instead of in a physical location. We allowed players to create their own servers that other players could join to play together. The game was made publicly available and was advertised for approximately two weeks. In that period the web page was visited 372 times, while the game was downloaded 35 times. However, only eight people that downloaded the game generated enough data for any analysis.

Based on the results of the first two playtests, physical interactions between artefacts were disabled (i.e. they could now intersect) and the artefacts themselves were not affected by gravity anymore. While the previous iterations showed that physical interactions between objects can allow for some interesting gameplay to emerge, the new modifications aimed to make it easier for players to build larger and more organic looking structures since the artefacts could now overlap. Additionally, the controls for placing the artefacts were fine-tuned, allowing for more precision and control.

A summary of the results of the final playtest are shown in Table II. Because of the small number of players for all eight games there was only one person playing the game at a time. The collected results suggest that players of the third game version found it much easier to control the placement of artefacts. Additionally, players were able to build structures faster than before, without spending too much time trying to work around the physical constraints of the previous versions. As Figures 8 and 1 show, players were able to more easily



(a) (b) (c) (d)

Fig. 8. **Artefacts Evolved in the Third Game Version.** By disabling physics and collisions between artefacts, players were able to more easily build a variety of different structures. Players bred a variety of 3D objects (e.g. long sticks, blocks) that allowed them to build different types of structures such as a tree (c) or a staircase (a).

TABLE II
RESULTS OF THIRD ITERATION MULTIPLAYER TEST

Total number of players	8
Total number of artefacts planted	95
Max generation	51
Total number of mutations	90
Total number of crossovers	5
Average game duration (in minutes)	5
Average number of artefacts planted per player	12
Average number of mutations per player	11
Average number of crossovers per player	1
Average number of seeds picked up per player	16

build a variety of different structures such as a staircase, a robot, and a tree-like artefact. While disabling the physical interactions between artefacts might prevent some of the earlier emergent gameplay, it did allow players to have more creative freedom over the structures they were building. A video of some gameplay footage form the third version can be found at the project page: https://cristi.itch.io/artefacts.

While only a small number of people downloaded the game, the feedback received from the ones that did was mostly positive. Below are a number of quotes received from the players: *"Fun, made a giant spiral staircase"*, *"Nice concept, keep it up!"*, *"Nice interface, easy to use. I quickly got frustrated trying to place objects together accurately. You may consider adding a "snap" so objects are flush against each other. Overall a nice sandbox, waiting to see how you expand on this"*.

While the playtests would have ideally included a larger number of participants, even the tests with few players suggest that it is possible to create interesting and novel gameplay by evolving 3D objects in a sandbox video game.

V. DISCUSSION AND FUTURE WORK

This paper presented a novel PCG-based game, which allows players to evolve 3D objects and use those objects to build larger physical structures. The results indicate that players enjoyed creating unique objects and were curious about the process of evolving them in an open world environment shared with others. The novel game mechanics in Artefacts allowed for some emergent gameplay, with players building structures individually and collaboratively. Because players share the same physical space, they were able to collaboratively evolve artefacts and extend the lineages of artefacts evolved by others.

While the game concept shows promise, we imagine a variety of further studies and improvements that would make it more engaging in the future. Since our playtests were performed with a rather small number of players, an important next step is a larger multiplayer experiment. What type of objects could be evolved by thousands of players collaborating and what type of physical structures could they build? An important question in this context is if a game like Artefacts could allow players to express their creativity in ways similar to a game such as Minecraft. A step towards answering these questions is the creation of a dedicated Artefacts server that enables a persistent virtual world, allowing many players to join at the same time.

Based on the players' questionnaire answers and observations during the playtests it became obvious that some would have enjoyed the addition of more gameplay elements. We imagine that in the future the game could have competitive elements that reward players for the unique structures they build or the objects they evolve. Additionally, the game could benefit from a resource-based system in which artefacts are limited and seeds have to be traded to get different variations. Furthermore, being able to interact in a more meaningful way with other players (e.g. talking to other players, trading artefacts etc.) and adding more physical rules (e.g. bouncing, springs etc.) could provide the player with a larger set of affordances. Giving players the means to share or sell the objects they evolved, similar to how players sell flowers in the marketplace in the Petalz video game [16], could not only allow the artefacts to create economic value but also increase the level of social interaction between players.

The current version of the game has a number of technical limitations. For example, the artefacts evolved in the game generally look very abstract and do not always resemble familiar shapes. In the future it might be possible to blend handmade content with generated artefacts. For instance, textures could be applied to the artefacts to create a variety of more natural looks. However, the biggest current limitation in the game is the lack of a persistent world that players could join at any point. With the current implementation of the game such a world was not computationally feasible. The most expensive operation was the querying of the CPPNs to generate the 3D objects, which lead to too long waiting times when a player wanted to join a server with many existing objects. In the future, this process could be accelerated by incrementally querying the objects closest to the player or by executing the Marching Cubes algorithm on the GPU instead of the CPU.

VI. Conclusion

Artefacts, a novel sandbox video game, allows players to interactively and collaboratively breed an endless variety of 3D objects. Importantly, players can build larger structures together with others by combining evolved objects. An iterative development approach was chosen, in which a total of three different game versions were tested. While the first physics-based iteration allowed some interesting gameplay to emerge, the final version in which physics and gravity were disabled, enabled players to build the greatest variety of different structures. Even though only a small number of people participated in the playtests, their feedback suggests potential for the game concept, and search-based PCG games in general. In the future it will be interesting to see what types of objects many players can evolve together in a persistent Artefacts world, and what structures they might build.

Acknowledgment

Special thanks to the Artefacts testers.

References

[1] P. Avery, J. Togelius, E. Alistar, and R. van Leeuwen. Computational intelligence and tower defence games. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1084–1091. IEEE, 2011.

[2] N. Cheney, R. MacCurdy, J. Clune, and H. Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2013)*, New York, NY, 2013. ACM Press.

[3] J. Clune and H. Lipson. Evolving 3D objects with a generative encoding inspired by developmental biology. In *Proceedings of the European Conference on Artificial Life (Alife-2011)*, volume 5, pages 2–12, New York, NY, USA, Nov. 2011. ACM.

[4] K. Compton and M. Mateas. Casual creators. In *Proceedings of the Sixth International Conference on Computational Creativity June*, volume 228, 2015.

[5] M. Cook and S. Colton. Multi-faceted evolution of simple arcade games. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pages 289–296. IEEE, 2011.

[6] E. Hastings, R. Guha, and K. O. Stanley. Evolving content in the galactic arms race video game. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG-09)*, Piscataway, NJ, 2009. IEEE Press.

[7] E. J. Hastings, R. K. Guha, and K. O. Stanley. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games*, 1(4):245–263, 2009.

[8] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 9(1):1, 2013.

[9] A. K. Hoover, P. A. Szerlip, and K. O. Stanley. Generating a complete multipart musical composition from a single monophonic melody with functional scaffolding. In M. L. Maher, K. Hammond, A. Pease, R. P. Y. Perez, D. Ventura, and G. Wiggins, editors, *Proceedings of the 3rd International Conference on Computational Creativity (ICCC-2012)*, 2012.

[10] A. Liapis, G. N. Yannakakis, and J. Togelius. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(3):213–228, Sept 2012. ISSN 1943-068X. doi: 10.1109/TCIAIG.2012.2192438.

[11] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[12] B. Mark, T. Berechet, T. Mahlmann, and J. Togelius. Procedural generation of 3d caves for games on the gpu. In *Foundations of Digital Games*, 2015.

[13] P. T. Olsted, B. Ma, and S. Risi. Interactive evolution of levels for a competitive multiplayer fps. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 1527–1534. IEEE, 2015.

[14] S. Risi and J. Togelius. Neuroevolution in games: State of the art and open challenges. *IEEE Transactions on Computational Intelligence and AI in Games*, PP(99):1–1, 2015. ISSN 1943-068X. doi: 10.1109/TCIAIG.2015.2494596.

[15] S. Risi, D. Cellucci, and H. Lipson. Ribosomal robots: Evolved designs inspired by protein folding. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2013)*, New York, NY, 2013. ACM.

[16] S. Risi, J. Lehman, D. D'Ambrosio, R. Hall, and K. Stanley. Petalz: Search-based procedural content generation for the casual gamer. *IEEE Transactions on Computational Intelligence and AI in Games*, PP(99):1–1, 2015. ISSN 1943-068X. doi: 10.1109/TCIAIG.2015.2416206.

[17] J. Secretan, N. Beato, D. D'Ambrosio, A. Rodriguez, A. Campbell, J. Folsom-Kovarik, and K. Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

[18] N. Shaker, G. N. Yannakakis, J. Togelius, M. Nicolau, and M. O'Neill. Evolving personalized content for Super Mario Bros using grammatical evolution. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2012)*, Menlo Park, CA, 2012. AAAI Press.

[19] N. Shaker, J. Togelius, and M. J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015.

[20] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines Special Issue on Developmental Systems*, 8(2):131–162, 2007.

[21] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10: 99–127, 2002.

[22] K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Real-time neuroevolution in the NERO video game. *IEEE Transactions on Evolutionary Computation*, 9(6):653–668, December 2005.

[23] H. Takagi. Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, 2001.

[24] J. Togelius and J. Schmidhuber. An experiment in automatic game design. In *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, pages 111–118. IEEE, 2008.

[25] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.