Cooperative Serendipitous Discoveries with Interactive Evolutionary Computation: a case study with 3D models

Jordy van Miltenburg jordyvanmiltenburg@gmail.com
Leiden Institute of Advanced Computer Science, Leiden University

Maarten Lamers m.h.lamers@liacs.leidenuniv.nl
Leiden Institute of Advanced Computer Science, Leiden University

Tim van den Bosch t.i.vandenbosch@hhs.nl
The Hague University of Applied Sciences

Abstract

Finding inspiration during the creation of a 3D model can be a difficult task. The designer can get stuck in the process. This study aims to find a way to stimulate serendipitous discovery in the process of creating a 3D model with the use of an interactive evolutionary algorithm. In this context, serendipitous discovery is the moment you find new inspiration from something you are not looking for. An interactive evolutionary algorithm is used to generate new 3D models based on the initial model created by the designer. To test if this is possible, an experimental tool is built. Reflected upon by the builders and four external evaluators from within the field. The reflection shows a mixed result from the builder and external evaluators. Concluding that serendipitous discovery can be stimulated depending on the state of the design process. At the beginning of the process, it is more likely for the tool the stimulate the user in having a serendipitous discovery.

Keywords

Interactive evolutionary computation, Collaborative interactive evolution, Genetic algorithm, Evolutionary algorithm, Serendipitous discovery, 3D modeling, Processing.

1 Introduction

As a designer, there is the everlasting struggle to find inspiration and think of the greatest ideas that are most suitable for the task at hand. In this technological age, every task is done with the use of a computer to try and improve the work of a human or assist in any way. But can this also be done for creativity? Creativity is a very human thing and a hard to understand concept, especially for a computer. But what if we see the computer as a tool to help the human or to guide the human. Programs like Photoshop or Illustrator have been around for a long time now and have helped many people to express their creativity. With creativity also comes the moment that you are stuck and can not think of a new idea to continue your work. This raises the question if a computer can help you during this phase and help you to continue your work.

During this research, the main focus is on how the computer can assist the designer in order to come to new insights or ideas. Finding a new idea or insight in an unexpected way is called a serendipitous discovery. This can happen at any moment, but can it also be sparked with the use of a computer? The core of the system will rely on an interactive evolutionary algorithm that will generate new variations of the designers work. The user is able to evolve the 3D model to different shapes that may spark new ideas for the next iteration of the 3D model. There is no example of such a system in the form where the user is in full control of the algorithm.

This paper will have a research approach called ”research through design”. This means that the team will reflect on their own work and the process of building the prototype. In combination with the reflection of some external people. The two reflections combined will give an overview of this system and check if it can help a designer during the creation of a 3D model. Or can be the starting point for future work in this field.

In the remainder of this paper, the reader will encounter related work in section 2, and an overview of the Research Approach in section 3. Subsequently, the design consideration will be found in section 4 followed by the implementation in section 5. The last two sections will contain the reflection in section 6 and the final section 7 is the conclusion.

2 Related Work

A computer as a creative companion to solve a problem you face. In the last years, the research towards a cooperative AI working together with a human got more attention. We can see the computer in four different ways according to Lubart (2005). 1) computer as nanny, where the computer is monitoring the human to enhance the creative process. 2) computer as pen-pal which is helpful to exchange ideas during a brainstorm session. 3) computer as coach to provide relevant information at the right time as an expert on the subject. 4) computer as colleague in which the human and computer work hand in hand to accomplish a goal. The human starts with the initial idea, that the computer can modify in a random or heuristic way. Starting a cycle back and forth between the human and the computer until the user or outside judges are satisfied. This is a great point to start for further thinking about how computers can guide or support humans.

2.1 Interactive Evolutionary Computation

To create a setting where the human and computer work together to develop new and novel ideas. We first want to introduce interactive evolutionary computation (IEC)(Takagi, 2001) or in other words aesthetic selection where the fitness function is not known. Therefore a human is used as a fitness function to evaluate different individuals within the population guiding the artificial evolution. These systems can produce a variety of different digital artifacts such as images (Sims, 1991), 3D models (Nishino et al., 2001).

A more recent example is the creation of new terrain in a realtime strategy (RTS) game. This process can be very demanding on the designer or programmer to develop novel ideas that are playable and thus do not contradict the rules. The Sentient Sketchbook (Liapis et al., 2013) is an example of a genetic algorithm (GA) working alongside the human designer. The system gives realtime feedback on the work of the human. The exciting thing about this project is the novelty search. With this approach, the optimization toward a quantifiable objective is replaced with the optimization of the population diversity.

Moreover, the search space is explored more and will result in unforeseen high-quality solutions. In a standard GA, the bad or the infeasible solutions are killed off so the better individuals produce the next generation. The Sentient Sketchbook, instead of only reproducing the good individuals also evolves the “bad” ones to find the search space's borders to increase the likelihood to find more feasible individuals. From a creative standpoint, this would mean a more significant variation of the original idea. It is called exploration in GA terms, to find new solutions instead of exploitation where you use the existing solution and change it to have a better fitness. Exploring more could potentially spark creativity for the user.

2.2 Collaborative Interactive Evolution

This section reviews the overlap of interactive evolutionary computation with collaborative interactive evolution (CIE). The difference is the number of people used in the system. In an IEC system, only one user is interacting with the system in the case of a CIE system there can be multiple people interacting with the system.

To explore the field of CIE Secretan et al. (2011) proposed an experiment. With this experiment, they attempted to improve upon IEC by using multiple users during the interaction. The system they developed is called Picbreeder, a system that generates a picture and can be evolved by the user. The images start with low complexity and get more complex over time with the use of neuroevolution of augmenting topologies (NEAT) algorithms (Stanley and Miikkulainen, 2002; Miikkulainen and Stanley, 2004). If the user wants to stop evolving the picture, they can save the image in a database. The database full of pictures is used to provide starting points for other users, so they do not have to start from scratch.

In this way, the system is both optimizing and exploring at the same time. Furthermore, the users who start from scratch explore the search space and the users that continue a branch are optimizing towards something. The best part about the system is the serendipitous discoveries made by the users. In some cases, the users were aiming to evolve an alien but in the end, evolved the alien into a nice car. By interacting with Picbreeder, the users or artists are helped in the creative process to design novel ideas. At the end of the experiment, the database was filled with over 7000 images and 140000 combined generations. Clune and Lipson (2011) extended upon picbreeder and created a 3D version, thus instead of using images they used 3D models to evolve in the network.

2.3 Agent Turn-based Co-creativity

Another approach for cooperative design with an AI is a turn-based system. Instead of the simultaneous design mentioned in the previous examples Guzdial et al. (2017) proposed a turn-based system. The human and the AI take turns to create a new map in Super Mario Maker — a level editor where users can create their levels for Super Mario — other users can play. To reduce user frustration, the AI can only make additions to the map and cannot delete tiles created by the user.

To accomplish this goal Guzdial and Riedl (2019) introduced a new framework where the human and AI can take different actions. 1) Artifact actions that directly impact the artifact that is to be designed. 2) Other actions that not directly affect the artifact, such as the AI explaining its actions. 3) Non-turn actions, so what can be done outside of the turn. For example, trying to learn from the human user or have the ability to observe the artifact. This new framework gives a lot of flexibility to explore different settings between humans and AI, in comparison with other frameworks (Guzdial et al., 2018).

This form of interaction comes the closest to cooperation between a human and an AI. In the Sentient Sketchbook, the AI is constantly working to find new configurations. But in the end, the human can ignore all the things suggested by the AI.

2.4 Serendipitous Discovery

As mentioned above, a serendipitous discovery can be made with the help of an AI. This serendipitous discovery is a discovery not foreseen by the human Quéau and Dagognet (1986) would describe it as ”The art of finding what we are not looking for by looking for what we are not finding”. This is very relatable for creativity, the so-called writers block happens to people working in the creative field. Struggling for days to come with a novel idea or to find inspiration for your next design. Then out of the blue something sparks the creativity or you have a moment of clarity to continue your design. This moment is a serendipitous discovery.

But how can we be certain if this so-called serendipitous discovery actually is a serendipitous discovery and not just conventional creativity? When working with creativity the initial step is to diverge from the original starting point to broaden your horizon. After the divergences of the idea we can start to converge back to a specific idea. In conventional creativity the convergence leads to a novel solution for the problem at hand. A serendipitous discovery usually leads to something different, thus an unexpected discovery is something completely different compared to what you started with (Campos and Figueiredo, 2002).

2.5 Key Findings

In this section we summarize the main findings of our literature review.

1. Different levels of cooperation exist between a human and AI, from nanny to equal.

2. An interactive system is able to work with one or multiple humans at the same time.

3. Taking turns gives a feeling of equality, this is also the case between human and AI.

4. Serendipitous discovery will lead to a more novel idea compared to conventional creativity.

3 Research Approach

In this section of the paper the research approach is explained and the question that we aim to answer.

This study will follow a method called research by design or in other words research through design. This method of conveying research is focused on creating knowledge through reflection. Thus the researcher will reflect on the created product to come to a conclusion (Laurel, 2003). This is unlike other research where a quantitative approach is used to come to a conclusion. To back up the findings of the researcher a small group of people is asked to use the product and answer some questions. But not nearly as many as in typical quantitative research.

The experimental tool that is built for this research will not focus on the experience of the user. So the ease of use or the appearance of the tool will not be taken into account. Moreover, the focus is on the tool’s functionalities and whether it can bring us closer to answering the question.

3.1 Research Question

With the use of an interactive evolutionary algorithm can we create a tool to stimulate serendipitous discoveries during the creation of a 3D model?

4 Design Considerations

In the following section all the different parts of the system will be discussed, to consider all the different options possible within the system as seen in figure 1.

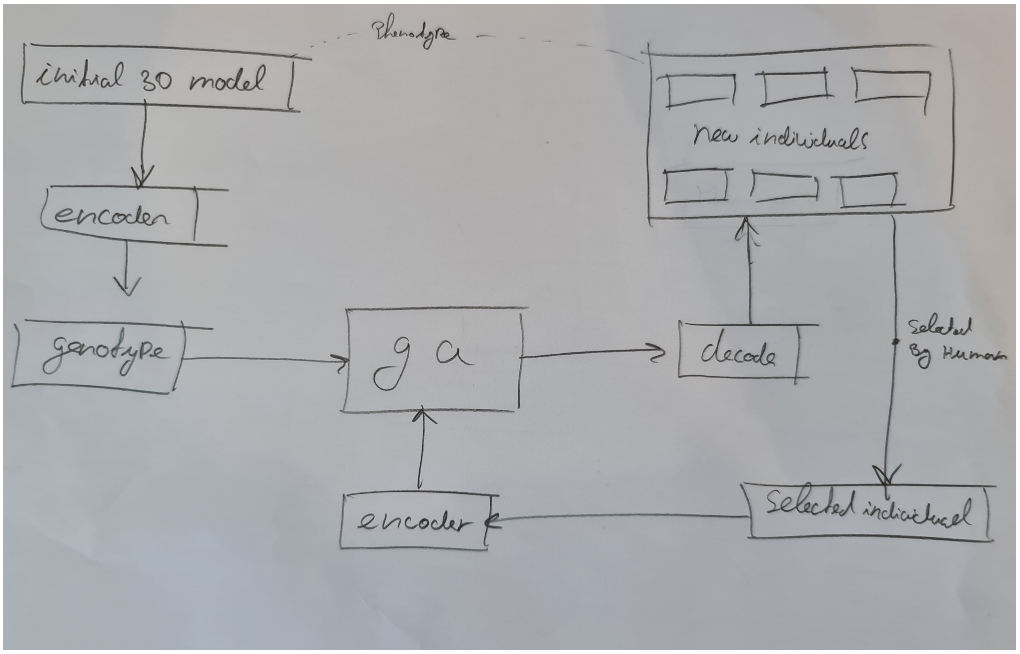


Figure 1: A rough overview of all the different parts necessary to create the interactive evolutionary algorithm.

4.1 Genotype

To start working with an evolutionary algorithm (EA) the first step is to define how we store the problem and the possible solution so the computer can manipulate it. The object that is forming, thus the possible solutions we call the phenotype. Their encoding is called the genotype, this is used within the computer to evolve and adjust (Eiben and Smith, 2015). The genotype is an essential part of an EA because it contains the individuals' DNA and can be represented in many different ways. In the following sections some representations will be presented to give an insight into the different options.

4.1.1 Binary

This classical representation is widely used in the world of EA and consists of binary vectors that are often referred to as bitstrings of binary strings. The bitstring has a fixed length and is build using 1's and 0's. Such a representation works well for pseudo-boolean optimization problems (Bäck et al., 1997). A binary representation can also indicate integers, but instead of seeing every single bit as a separate number, substrings are used to indicate a number. Moreover, if the substring contains 8 bits the numbers that can be represented are 0 to 255.

4.1.2 Real-Value

Instead of only using 1's and 0's, real-value representation strings consist of float values. When using this kind of representation, the algorithms benefit from a mutation-based approach, searching through the search space by adding a multivariate zero-mean Gaussian random variable to each number in the string. If crossover is the algorithm's primary focus, then the problem is often represented in a binary representation, such as mentioned above.

4.1.3 General Tree

In the previous representations, the genotype is encoded using a string or an array with linear order. The next representation is also represented using a string, but it is not a linear one. The general tree as the name suggests is formed like a tree built with nodes and edges. The nodes are the bytes in the array but the edges determine the linkage between the nodes. So for example if we have a simple tree consisted of nodes ABCDEF where A is the start node connected to B and C. Continuing with D and E connected to C the string would look like this (A,(B,C,(D,E(F)))). In this way we can take a part of the DNA and use this in the crossover operator to preserve larger portions of the individual.

4.1.4 Lindenmayer System

Another tree-like system is the Lindenmayer system (L-system) (Lindenmayer, 1968). An L-system is centered around the notion of rewriting. In principle, a way to define complex objects by successively rewriting parts of the starting object. It has a simple structure initially but gets more complicated further along in the process. Such a system is created using two simple rules, A - AB and B - A where the - is a connection to the next layer in the tree. If we start with B the next layer is an A which results in the next layer to A B. If we continue this process the result will be a large tree consisting of A's and B's in a specified pattern. This procedure will not give the opportunity to change individual parts of the system. The only thing that can be changed is the rule that determines how the structure is built.

4.1.5 Directed Graph

Compared to a tree structure that goes in only one direction, a directed graph (DG) is a network that also contains nodes and edges. The different notes can also interconnect, with in addition a given direction in which the edges are connected. This can result in nodes influencing only specific other nodes through the directional edges. This intricate system of nodes and edges contains more data and can recreate a data flow, like neurons sending information in the human body (Sims, 1994). The DG can be seen as a building block for the individual, which may contain multiple DG's. All this extra information is not necessary and will overcomplicate the genotype.

4.1.6 Genotype Evaluation

Finally to create a 3D model and convert it to a genotype, different elements need to be used. The first is the ability to evolve to a more complex structure over time to create a more significant search space for the algorithm. So representations with a fixed-length are not suitable for adding more complexity down the line. The binary and real-value representation both cannot implement a function like this.

The second requirement is the speed at which the individuals evolve. If the speed of evolution from the start is too slow the users may lose interest and causes human fatigue. To tackle this problem the individuals can be created using building blocks, so larger parts of the DNA are used in the crossover. Thus giving more lavish sets of genes to the offspring and may result in more significant changes between generations. The general tree gives the option to take a specific part of the tree that can be added to an individual. Moreover, the L-system gets more complex over time but only with the predefined rules that don't change.

At last the system can modify the initial individual in a more drastic way, the users may serendipitously discover new insights to help them in their creative process. Individuals based on DG's are able to accomplish big changes through the generations. But the DG's as a building block also contain a lot of redundant information for the application at hand.

4.1.7 Choice of individual representation

For this project the most optimal genotype representation is the general tree. Moreover the general tree is flexible for growth and large parts of the genotype can be interchanged to give lots of variation. The way the genotype is structured complements the creation of hierarchical models, this due to the way a general tree is structured.

4.2 Phenotype

As mentioned above the phenotype is the solution to the problem, or in this case the 3D model visible for the user. So a visual representation of the genotype can be seen in figure 2. To form the initial phenotype the user can build a simple 3D model using cubes, which can be manipulated to the user's needs. Moreover the position, shape and rotation can create the initial 3D model that will be used to form the next generation. The project's goal is to inspire the user and not create the best or most beautiful 3D model. By only using simple cubes the core of the project stays in focus. With the different sized cubes there are already many possibilities for the user to create the initial 3D model. Therefore the choice is made to only work with cubes and not use any other primitives like a sphere, cylinder or pyramid.

4.3 Encoding & Decoding

After creating the initial 3D model the shape needs to be converted to the genotype so the algorithm can use this individual to generate the population. In order to do this the phenotype needs to be encoded to a genotype, which contains all the information about the different parts of the model. Such as the position, scale and rotation of the individual cubes. After the algorithm has done its thing the new generation of individuals has to be decoded back to a 3D model so the user can see how the new models look.

4.4 Mutation & Crossover

The algorithm will generate new individuals from the selected individual. With the use of mutation they will be slightly modified compared to the selected individual. This mutation will happen using a step size, so a predetermined amount modifies the original genes. By using a step size the new generations will have more resemblance with the selected individual. For more drastic changes crossover will be used. In this process more significant parts of the selected individual will be selected and copy and pasted into a new individual. In this way, particular features can be interchanged between different individuals.

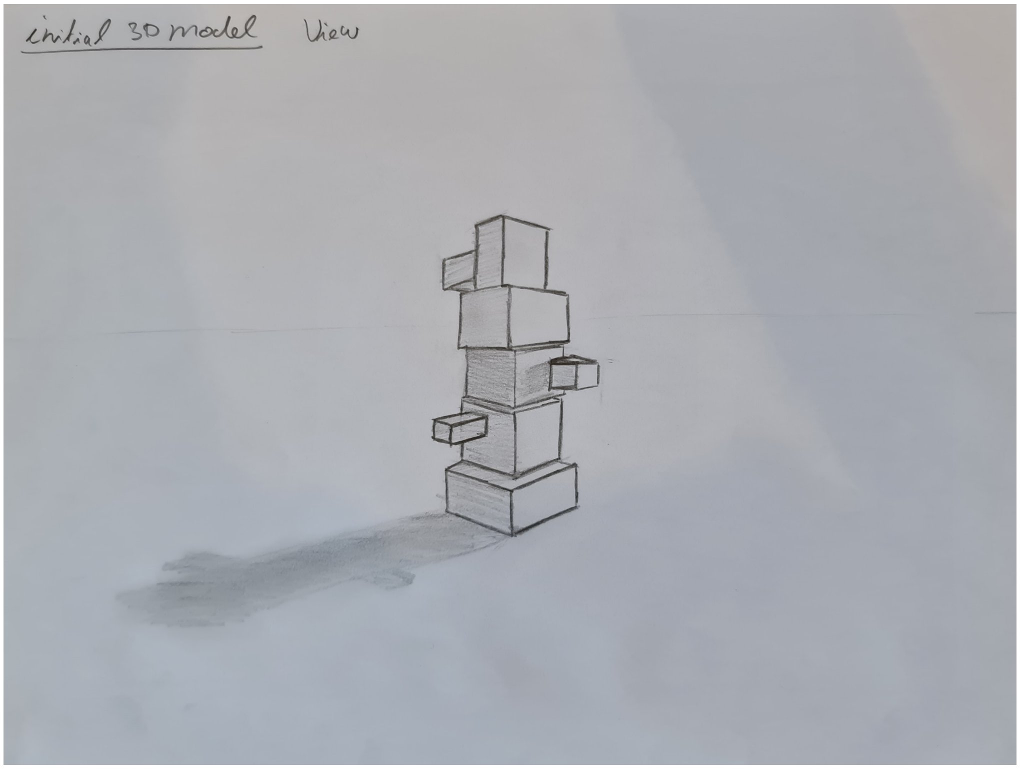


Figure 2: This is a sketch of how the initial 3D model would look created by the user, so the algorithm can start evolving.

4.5 Selection

In typical evolutionary algorithms, the selection of the parent is done with a selection operator. This operator is using the different fitness levels of the individuals. High fitness will result in a higher chance of being selected for the next generation. In interactive evolutionary algorithms, there is no fitness function to evaluate each individual. The user of the system is the fitness function, thus selecting the individual that is most suited for being the parent of the next generation. In this system the user can only choose one individual used for crossover instead of ranking the whole population. This so that the user will not get fatigued during the process. The selection will be made with the use of an interface where the possible parents will be displayed. Figure 3 gives a rough idea of how such an interface is going to look.

4.6 Environment & Tools

To create the system some considerations have to be made. In which environment is the program running or in which language. In the following paragraphs we explain the different advantages and disadvantages of some of the considered programs.

4.6.1 Processing

Processing is an accessible environment to start with and has some abilities to work with 3D objects. But processing is not created to be a 3D modeling software, so many

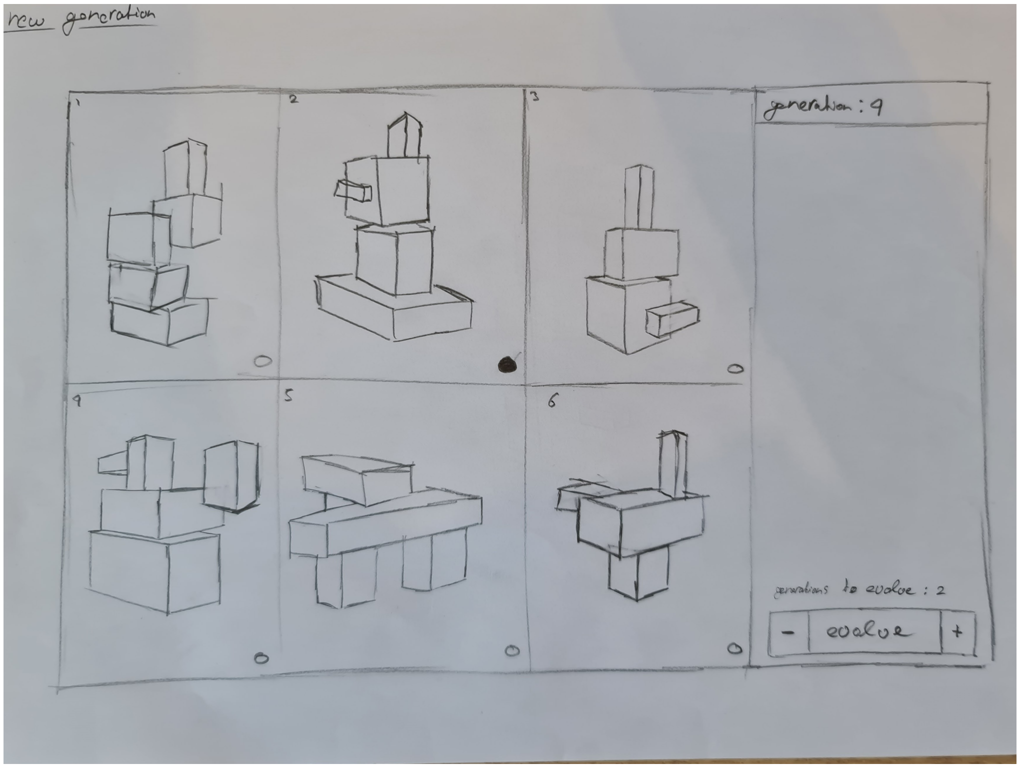


Figure 3: The interface which is used by the user to select the best individual to evolve the next generation. This is a mock-up version and will later be changed in the tool.

essential features must be created first. Think about placing primitive shapes or manipulating them in space.

4.6.2 Cinema 4D

The next program is a 3D model and animation software called Cinema 4D. Within the program there is the option to code in python. All the actions that can be performed with the mouse and keyboard can also be programmed using python. The benefit of using a program like this is the strong features within the software. Unfortunately, programming in Cinema 4D is not optimized to create looping code, which is necessary to run the algorithm.

4.6.3 Unity

In the game making industry Unity is widely used due to its versatility; all aspects that you may need are available in Unity. However this program uses C# as the main language to script, which is not the preferred language of the team.

4.6.4 Rhino

Rhino is a 3D software primarily based around free-form surface modeling. But the more interesting part is a plugin named grasshopper. This plugin allows for visual programming in Rhino, this is especially useful for parametric modeling or generative art. Furthermore, one of the grasshopper components enables the user to write python code to give more flexibility and control. The lack of programming options makes it not the preferred choice for this application.

4.6.5 Conclusion

In the end the choice is made to create the tool in different modules. The main part of the tool is built using processing, thus the Evolver and Interactor. More about the modules in the next section of the paper. The decision is made to work with processing because of the ease of use and the team's experience with this framework.

5 Implementation

In the previous section different programs and environments were discussed to determine the best one for this experiment. After some testing with the different programs the decision is made to split the experiment into separate modules. The first is the Model Maker; in this program, the user can create a 3D model used in the tool. The second module is the Evolver; this houses the evolutionary algorithm (EA) with all of the functions necessary to run the system. The last module is the Interactor; this module adds the interactive component to the EA so that the user can manipulate the 3D model.

5.1 Model Maker

At first the user will create the initial 3D model, which is the starting point for the EA. The Model Maker makes it easy to convert a 3D model to a genotype that is readable for the EA. So the user can create a simple 3D model built out of only cubes, which is an abstract representation of the desired 3D model.

5.2 Evolver

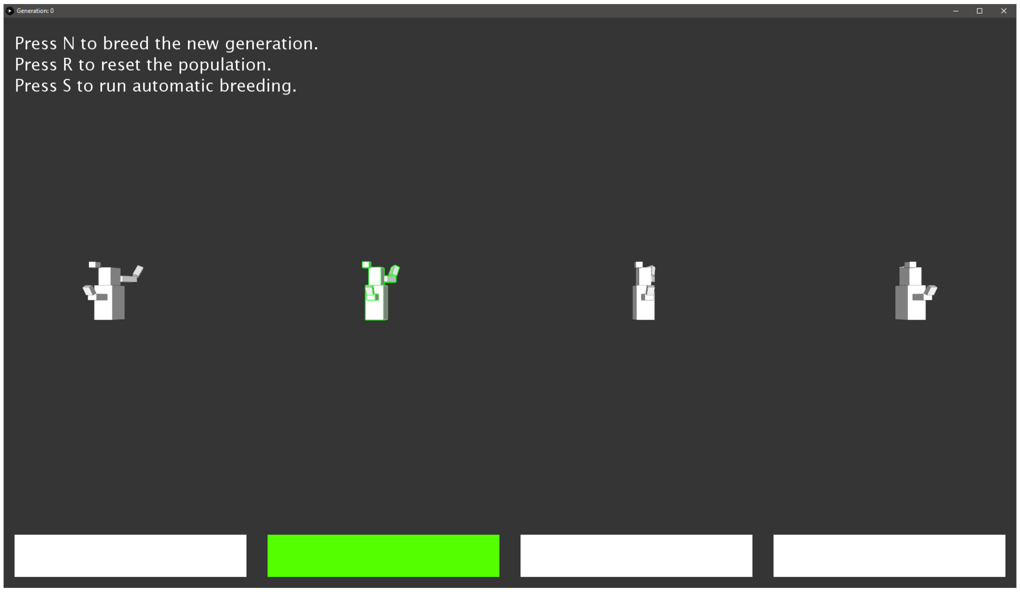
The second part of the system is the Evolver. This module contains the heart of the tool. All the different operations needed for an EA are done in this module. The genotype is decoded and converted back to a phenotype so the user is able to see the initial 3D model. In the first generation all the individuals in the population are a copy of the initial model. To display the 3D model each cube is converted to a node in the tree. Each node contains the position, rotation and scale of the cube. Moreover the data about the parent and the children is also saved in the node. All the nodes combined form a tree. The nodes in the tree have a relative position from one another to ensure that nodes will not be overlapping because they have the same position.

To create the offspring for the next generation crossover is applied to the current population. The selection for the crossover is as follows. The user selects the first parent and the second parent is selected at random. If the user does not select a parent both of them are selected at random. Crossover happens with two randomly selected crossover points. These crossover points are nodes in the tree that will be used to create the offspring. The user-selected parent is always the base for the offspring. Thus the user-selected parent is copied as the base before the subtrees are switched. The selected node from the random parent will be replacing the selected node of the user-selected parent. Therefore the user-selected parent is more prominent in the offspring. The other case is where the random selected parent is the base with the addition of the subtree from the user-selected parent that is not used in the offspring.

After the crossover it is time for the mutation of the offspring. The mutation will give the new population more diversity or new features that may be interesting for the user. Novelty search as mentioned before, is not going to be implemented into the system. It will result in a more diverse population, which may influence the user's control over the system.

5.3 Interactor

The last part of the system is the interface that will allow the user to control the tool. First, the option to rotate, scale and move the 3D models so the user can view the models from different angles. This is done with the use of the mouse. There are as many buttons in the lower section of the screen as there are 3D models, thus the population size. These buttons are used to select a model for the next generation. Both the button and the corresponding 3D model will turn green to indicate which model is selected. Different keys on the keyboard are used to continue to the next generation, run the algorithm automatically or start over with a fresh generation. The automatic breeding is just an option to let the algorithm run randomly. The interface can be seen in figure 4.



Press N to breed the new generation.
Press R to reset the population.
Press S to run automatic breeding.

Figure 4: The interface used in the final prototype. To interact with the tool different keyboard presses are used. The mouse is used to rotate all the models around the center axis. The four buttons at the bottom are used to select the preferred model.

6 Reflection

This section of the paper is dedicated to reflecting on the tool from both the team and some external evaluators, to give a better foundation for the claims that will be made in the conclusions. The first part of the reflection is about building the prototype and reflecting on some of the difficulties. The second part is the reflection of the team that created the tool. They can give more insight into the project from the builder's perspective. The last part of the reflection will contain the opinion of users from outside of the team. They have a nonbiased opinion about the tool and may see things from a different perspective. The personal reflection is written before the external reflection, so the external evaluators' comments would not influence the team. This could depreciate the comparison between them.

6.1 Process

The process of building the prototype brought some difficulties to the table. They were ranging from the data structure to some internal limitation within processing.

At first the data was saved in a long string to represent the 3D model. Quickly the conclusion arose that the string did not contain the underlying connection between the nodes. Another structure is used, a class called node where all the data for one cube is saved and the relationship between them. This resulted in a 3D model with a hierarchical structure because the nodes had a parent and children.

The second hurdle emerged during the manipulation of the tree. Processing uses Java as its primary language and within Java, referencing to data was holding us back. Different parts of the model would have a reference to the data from the parent. This resulted in an offspring with individuals containing the same references to specific data. Furthermore, this would interfere when a manipulation happened to one individual and the rest of the population would also change. This was most visible during the crossover to create the offspring for the next generation. Moreover the different subtrees would have the same reference, all from the same parent. To resolve this issue, a copy is made with every manipulation so all the nodes would get their own reference.

Another problem that arose was linked to the hierarchical structure of the 3D models. At first, the 3D models were built using an absolute position, so each node had a position in the global space. This resulted in overlapping nodes after the crossover. To resolve this issue, a relative position is used so the nodes would position themselves according to the position of the parent node. Unfortunately this brought an even bigger problem to the table that can't be resolved within processing. All the nodes drawn on the screen are using a relative position. The position is saved on a matrix stack with a max depth of 32. Thus when the 3D models get bigger more positions need to be saved on the matrix stack. If the matrix stack is full, the program will crash. To stop the program from crashing the user needs to keep the 3D models a bit smaller and not go for the biggest model each time.

6.2 Personal Reflection on Research Aims

The following part of the reflection will contain the remarks and thoughts from the builders of the tool. Different aspects of the prototype will be discussed, such as the user's amount of control and how this tool can help the serendipitous discoveries for creating a 3D model. And at last, some more remarks found during the building of the prototype.

First of all, during the interaction with the system the user stays in full control of the EA. In this way, the 3D model remains true to the initial model and the user's further choices. Moreover, the selection of the user will always have a more prominent place in the next generation. The selected model is used for the creation of the offspring. So the amount of control reduces large deviations from what the user prefers.

Furthermore the deviation from the original model is necessary in order to find new ideas, thus have a serendipitous discovery. If the change is not big enough the gradual nature of the EA will slowly change the model to a better or worse variation, this will not spark the serendipitous discovery moment. i.e., where the user realizes to have found a new and interesting model. On the contrary, if the tool gives a too random model or without any resemblance to the previous generation, the tool seems not to consider the user's input. This would go against our idea of letting the user guide the evolutionary algorithm.

To answer the research question from our point of view. "With the use of an interactive evolutionary algorithm can we create a tool to stimulate serendipitous discoveries during the creation of a 3D model?". In our experience, no it didn't spark serendipitous discoveries during the creation of a 3D model. The level of abstraction is too high. Thus the model used in the tool contains too little information to display the intended model. It evolves too fast to what seems to be a random shape, even with some creative thinking in mind. If you are explicitly creating, for example, a cactus, the new shape made by the tool is something completely different.

But if the object that you are design does not have to match any mental model of a specific object, the tool can give you a lot of new shapes that can be a starting point for the designer. The settings can be adjusted so the algorithm makes smaller changes to the model. This can be used to generate different variations of a model. Moreover, if you have a forest with trees consisting of only one model the tool can generate slightly modified trees so they all do not look the same.

6.3 External Reflection on Research Aims

This part of the reflection is based on the remarks from some external evaluators. To give more foundation to the arguments, the team asked some people with 3D modeling experience. They can be designers, developers or anyone who has some experience. All the participants for this evaluation come from within the network of the team. To conduct the test, one team member will be present during the evaluation. The evaluation will be performed in an informal setting so the subject can ask questions and give remarks on the tool. After playing around with the tool a set of questions will be presented to the evaluator, so we get a baseline throughout the different evaluators. The questions will be open so the users will not be limited in a specific direction and can answer according to their own experience. The walkthrough sheet can be found in appendix A. This contains the questions and also the important information that is provided during the evaluation. The evaluators' names will be kept out of the paper due to privacy reasons but can be requested from the first author. Due to the Covid-19 pandemic most participants are sourced from within Leiden University. Contact has been made with Grafisch Lyceum Utrecht to find students from the game artist and game development program, but making appointments to convey the evaluation unfortunately was too much of a hassle in the current pandemic.

6.3.1 Designer A

The first participant is a student following the master Media Technology at Leiden University, who has followed some 3D modeling courses at the Technical University Delft. His preferred software is Autodesk Maya but he also played around with other programs. He would rate his experience in 3D modeling between 6 and 7. Initially, he never heard of serendipitous discovery, but after the explanation, particular moments arose. This is especially true during generative design, usually based on just trying and seeing if something works. The following part is paraphrased to better encapsulate the thoughts of the evaluator.

Most of the times the initial shape stays present during the interaction but there is a point when this completely changes into something else. This can be a good thing to find new ideas and better understand how to continue your design. Moreover, if you need to design something with more exact dimensions, this change will not be helpful for that specific design. I would like to *see how this tool works with a "real" 3D model and not with the abstract version or resemblance of the model. By using the cubes you lose the essence of the model and the impression is entirely different. But for now, I think this tool can help you in an inspirational way and help you rethink the model when done with the interaction.*

Other applications I see for this tool lay in the creative field, especially the generative function. When texturing an abstract model, the same principle can be used, start with one and let the algorithm find new variations for you. This would only work if the texture does not resemble the object or is specific to the object.

6.3.2 Developer B

The next evaluator graduated from the master Media Technology at Leiden University, currently applying for a Ph.D. position regarding socially embedded AI. He is more focused on the development side with experience in evolutionary algorithms and neural networks. Nevertheless, 3D design is not new for him because of a course he took learning blender. Rating himself with a 5 on a scale from 0 till 10 as it comes to experience with 3D modeling. Serendipitous discovery was new until he heard the explanation and mentioned some moments where this had happened to him. The following part is paraphrased to better encapsulate the thoughts of the evaluator.

The first interaction resulted in a shape with different branches, which resulted in wanting to get more branches. The aim or goal changed during the interaction. At first I did not have any aim or direction I wanted to go for, but after a couple of generations the branches came, and I aim toward it. The next model I wanted to create was something without tall structures. During the interaction, a sort of ball formed and this was my next focus. Each time there is an aim or focus, but the control is not enough to keep that aim. The exploration you get is great, but if my purpose is to find different variations that wouldn't happen right now. Sometimes the algorithm just deletes parts that I like and want to keep. The option to lock certain parts of the model would be a big addition. So I can give more guidance after the exploration. If the tool changes the 3D model I'm adapting with the tool to explore the possibilities further.

Something I would see fit for this tool is one of those mirrors where you can see the outfit digitally projected on your body. Showing you different combinations of clothing and colors to ultimately arrive at your perfect outfit.

6.3.3 Developer C

The second developer is currently graduating from the master's program Media Technology at Leiden University. Familiarized himself with artificial intelligence by taking some courses on evolutionary algorithms, reinforcement learning, and neural networks. 3D design is something he started doing 1.5 months ago, mostly for the development of game assets. For him, serendipitous discovery is not something new. He mentioned an interesting example. "Online libraries are lacking serendipity because you can't see the book next to the one you are actually looking for." The following part is paraphrased to better encapsulate the thoughts of the evaluator.

In the beginning almost all shapes would get tall and only go in one direction. If I want to get a split in the shape, so there are more endpoints or leaves, this is possible. The algorithm is still controllable until a certain point. Sometimes a particular feature I like will disappear from all the individuals in the population and is gone forever. During the interaction with the tool, I stay in control with a small amount of randomness. You never know what it is going to do. At first I will be picking the model I like the most, but after a couple of generations I also start to guess or predict what will happen, thus trying to understand the algorithm and choose my actions accordingly. When getting to the result, there is a resemblance to the initial model, and parts stay present throughout the generations. Sometimes you want something completely *different. Also, the data structure is visible in the model. Almost all the models have a root or base as a starting point for the model. I would have liked to see models without the base and relative positioning.*

With some imagination you can definitely see different shapes in an abstract form that later can be used as inspiration during 3D modeling, just like how we see shapes in clouds or the Rorschach test with the ink patterns. At one point during the interaction with the tool, I saw something that looked like a low poly explosion. Coincidentally I was working on a game where I need an explosion like that. So now I would revisit the design of the explosion after using the tool.

What comes to mind is using an image classifier to recognize something in the 3D models and try to create a tree. Thus keep the algorithm running until the image classifier thinks the 3D model is a tree. Furthermore, it would be an excellent tool for generative art if there was more control within the algorithm. Actually every creative sector could use a tool like this. When you get stuck and need more inspiration, it could help you in a new direction.

6.3.4 Designer D

The last external evaluator and the second designer works as a graphical designer and content creator, with some 3D modeling experience using Cinema 4D. Anything related to evolutionary algorithms is new for him. Serendipitous discovery on the other hand, is something very familiar. During the study Communication and Multimedia Design at Avans University of Applied Sciences in Den Bosch, one project revolved around this subject. They had to give visitors to an exhibition a serendipitous discovery. The following part is paraphrased to better encapsulate the thoughts of the evaluator.

From the first interaction I intended to get a big structure with big pieces. In the end, this worked out just as I intended. The algorithm is definitely controllable by my actions. If I now keep selecting small cubes, it should stay lower and be more widespread. Also, by selecting the smaller object I get the feeling the overall structure changes. So sometimes, the initial model remains more prominent in the 3D model than other times. This gives a great variety of options to take into account during the designing of the model. If I compare the result with the starting object, it is a more mature version of itself. Usually a bit more chaotic, but I like that. The shape becomes more enjoyable throughout the generations and I made conscious choices to get to this point. Without the tool I would have never thought of the variety that the tool provides. Usually if I start designing something, the start stays visible during the process. The tool helps to takeout this repetition.

As of this moment I am doing a lot of marketing and making commercials. In most cases we tend to go for the "standard" commercials to keep it safe, but with a tool like this I am curious to see what could happen. What other surprises we might find.

6.3.5 Overall

The next section will give an overview of all the input provided by the external evaluators. In most cases the initial shape stays present during the interaction, which provides you with more insight into the current shape. On the other hand, the system can give you something completely different. This really helps to explore and see the current project from another perspective. During the interaction the goal you have in mind can change by some new feature that will arise in the 3D model. This can be something you expected or predicted, and something completely random that will spark some new inspiration. Imagination is a big part of this system because the representation is abstract. By using cubes to resemble the actual 3D model you lose the essence of the 3D model. The user needs to see the bigger picture and try to find inspiration in the models. This is just as seeing shapes in clouds and the Rorschach test.

7 Conclusions

With the rise of artificial intelligence, more tasks are being optimized or enhanced using a computer. Most designers have experienced it a serendipitous discovery, a certain spark of inspiration. But is it possible to help out this process with the use of an interactive evolutionary algorithm?

The user can control the algorithm and push the 3D model in a specific direction until a certain point because of the algorithm's random nature. This point is an opportunity for serendipitous discovery, where you can find new insights or inspiration for the model. If you are creating something specific, it can be challenging to keep the overall shape. The algorithm can produce something completely different. This may not be very helpful in finding new inspiration for your 3D model. Thus not allowing the user the have a serendipitous discovery. It depends on what stage of the design process you are. At the beginning of the process you are more likely to be exploring the different possibilities for your 3D model. In this stage, the tool helps the designer and the changes to the model can be bigger. Further along in the design process, the changes to the model need to be smaller in order to keep the shape of the existing model. Moreover, bigger changes will more easily result in a serendipitous discovery than small changes.

To restate the research question, "With the use of an interactive evolutionary algorithm can we create a tool to stimulate serendipitous discoveries during the creation of a 3D model?" Yes, the tool created in this research can stimulate serendipitous discoveries when creating a 3D model but not in every circumstance. The more specific the 3D model needs to be, the harder it gets to use this tool for inspiration. Preferably the tool is used at the beginning of the design process when there are more possibilities for change. Will the tool help all of its users? No, this tool will not guarantee a serendipitous discovery, it only helps to stimulate the user to get one.

References

Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). *Handbook of evolutionary computation*. CRC Press.

Campos, J. and Figueiredo, A. D. d. (2002). Programming for serendipity. In *Proceedings of the 2002 AAAI Fall Symposium on Chance Discovery-the Discovery and Management of Chance Events, AAAI Technical Report FS-02-01*.

Clune, J. and Lipson, H. (2011). Evolving 3D objects with a generative encoding inspired by developmental biology. *ACM SIGEVOlution*, 5(4):2–12.

Eiben, A. E. and Smith, J. E. (2015). What is an evolutionary algorithm? In *Introduction to Evolutionary Computing*, pages 25–48. Springer.

Guzdial, M., Chen, J., Chen, S.-Y., and Riedl, M. O. (2017). A General Level Design Editor for Co-creative Level Design. *AAAI Workshop - Technical Report*, pages 2013–2015.

Guzdial, M., Liao, N., and Riedl, M. (2018). Co-creative level design via machine learning. *CEUR Workshop Proceedings*, 2282.

Guzdial, M. and Riedl, M. (2019). An Interaction Framework for Studying Co-Creative AI.

Laurel, B. (2003). Design research: Methods and perspectives. page 82.

Liapis, A., Yannakakis, G. N., and Togelius, J. (2013). Sentient sketchbook: Computer-aided game level authoring. *Proceedings of the 8th International Conference on the Foundations of Digital Games (FDG 2013)*, pages 213–220.

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of theoretical biology*, 18(3):300–315.

Lubart, T. (2005). How can computers be partners in the creative process: Classification and commentary on the Special Issue. *International Journal of Human Computer Studies*, 63(4-5 SPEC. ISS.):365–369.

Miikkulainen, R. and Stanley, K. O. (2004). Competitive Coevolution through Evolutionary Complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100.

Nishino, H., Takagi, H., Cho, S.-B., and Utsumiya, K. (2001). A 3d modeling system for creative design. pages 479 – 486.

Quéau, P. and Dagognet, F. (1986). *Éloge de la simulation*, volume 16. Editions Champ Vallon.

Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403.

Sims, K. (1991). Artificial evolution for computer graphics. *SIGGRAPH Comput. Graph.*, 25(4):319–328.

Sims, K. (1994). Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 15–22.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.

Appendix

A External Evaluator Plan

Introduction
Introduce myself
My name is Jordy van Miltenburg studying Media Technology at Leiden University currently working on my thesis.

Information about the external evaluator
What do you do?
What is your experience level with 3D design or with EA?

Introduce subject
This is an experimental tool that is made to help a 3D designer in the process of designing a 3D model. The tool uses an interactive evolutionary algorithm in order to give new suggestions to the user.
Explain what an interactive evolutionary algorithm is.
Explain how the interface works.

Explicitly say that it is not about the looks of the interface or the working. Also the final result, so the last 3D model is not supposed to be nice and beautiful. We only look if the tool could help the designer in the process of designing.

Research question:
With the use of an interactive evolutionary algorithm can we create a tool to stimulate serendipitous discoveries during the creation of a 3D model?

Do you know what a serendipitous discovery is? Explain if they don't know. Have you ever experienced such a moment?

Task
Start the experiment.
Just let the user play with the tool so they can figure out how everything works and help them if they don't know.
Create a shape that you like from the initial 3D model with as many generations that you like.

Why do you like this shape?
Is this your intended shape from the beginning? Or did it came to you during the interaction?

Conclusion
What do you think of the end result compared to the initial model?
If the tool helped you in what form?
What new insights did you find during the interaction?
What other application do you see for this tool?

Thanks for participating.
If you want I can send you the paper when finished.
If you have questions you can always ask.