

Project Proposal

Evolution Nine

Evolving Creatures in a 2D Platform Game

Lecturer: Mick Grierson
Year 2 - Creative Projects

Student: Igor Corrêa
Reg. N°: 33271183

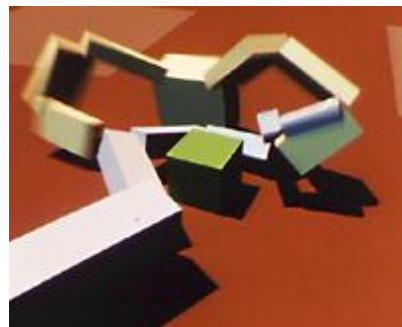
Project description

Evolution Nine and consists in a 2D platform game that utilizes concepts of genetic algorithms. In the game the player watches nine creatures move and jump randomly, one at a time, in a level containing platforms, obstacles and possibly other objects. The level will also contain an object representing the goal, so that when the creature touches this object the level will be cleared and the player will be taken to the next level. The player's part is to select the creatures whose movements he/she thinks takes it closer to the preferred path towards the goal object. After the player selects some creatures, another generation is bred from the ones selected and they move around again, one by one. This process will repeat until the player manages to make the creatures' movements converge in the desired way so that the goal object is reached.

Context

"In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems." – Wikipedia

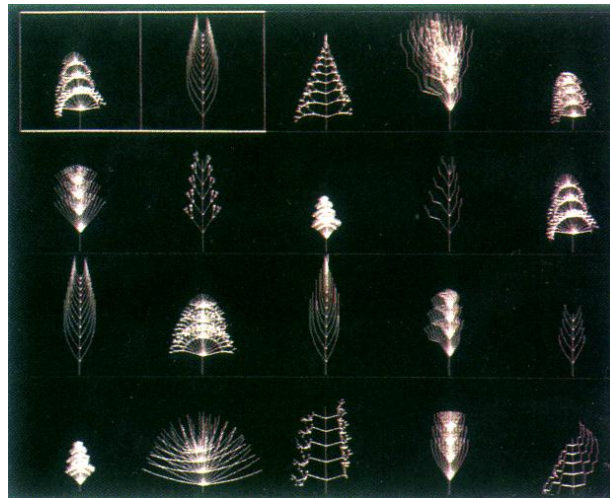
Genetic algorithms can be used in a great variety of applications [1] ranging from very practical uses in fields such as Economics [2] to game applications as Natural Motion's Euphoria [3] which, by using neural networks and genetic algorithms managed to create incredibly realistic animations for the human body dynamically. This last application can be mentioned a descendant of Karl Sims' research where he also used genetic algorithms and neural networks to evolve virtual creatures to crawl, swim, fight, etc. [4][5]



Karl Sims' creatures competing for a green cube

The usual implementation of a genetic algorithm contains a population of individuals that have their fitness tested, that is, a function tests how well they did in the presented conditions. Then, to create the next generation, new individuals are bred from the best fitted. The new individuals have their fitness tested and the process repeats until a pre-determined fitness is achieved or until a maximum number of generations are generated. Even though the automated fitness test is usually part of genetic algorithms, sometimes it is adequate to make the user of the application choose directly the ancestors of the next generation. One example of this approach is the use of genetic algorithms to

create complex structures and textures proposed by Karl Sims, in which a human observer is responsible for attributing the fitness to generated images. [6]



Karl Sims' example of mating plants structures

The figure above shows many plant structures, the first two being the parents of the others. The user is supposed to combine the structures to achieve a desired form.

The idea

The initial inspiration for the proposed project came from Internet applications such as BoxCar 2D [7], which is a program that evolves randomly generated cars based on their performance, and Cambrian Explosion [8] which is another program that evolves creatures into learning how to walk. Both programs use genetic algorithms and caused me great interest. Also, I was amazed by how entertaining just watching those cars and creatures evolving could be, and wondered if that could be turned into a game where the person watching could participate in a more active way on the evolution process, besides just setting parameters before the beginning of the simulation.

How it will work

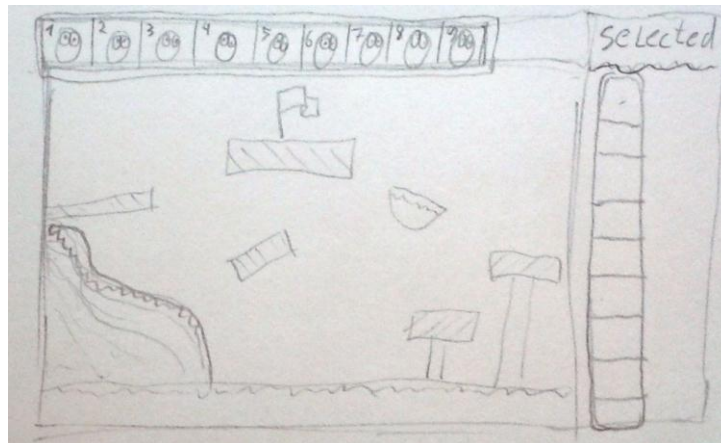
Here I will brake down the game flow into a summarized set of simple steps in order to make it clear how it will work:

- 1 - A level is loaded;
- 2 - Nine creatures are randomly generated;
- 3 - The creatures' trial takes place:
 - a) The first creature jumps around the level a certain number of times. Let's say five times;
 - b) All the other creatures, one by one, do the same;
- 4 - The player chooses the creatures whose movements he/she liked the most;
- 5 - From the creatures chosen by the player nine new creatures are bred, thus forming the new generation;
- 6 - This process is repeated until one of the creatures reaches the goal object;

7 - The next level is loaded and we go back to step 2.

Here are some images to roughly illustrate what the game screen could look like during some of the steps listed above:

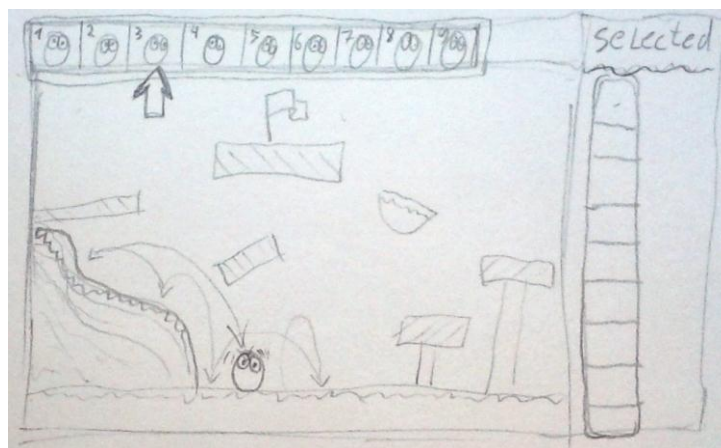
Steps 1 and 2



Level and initial creatures

Here the level can be seen and the generated creatures are represented inside squares on a row at the top left of the drawing. Each creature is identified by a number.

Step 3



Third creature after its trial

Here a creature is shown in the level after jumping five times. It is the third one, as the arrow on the top left is showing. Thin arrows illustrate the trajectory of each random jump previously performed by the creature.

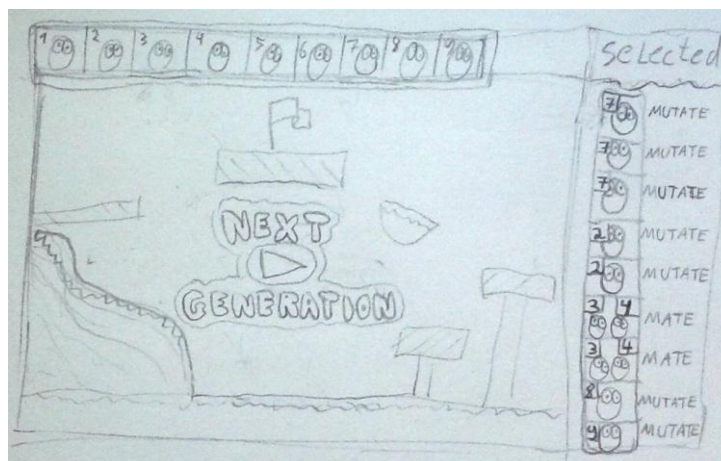
Step 4



The player chooses the winning creatures

Here the player drags creature number 7 to a slot on the column to the right, where the selected creatures will be. As shown on the next drawing, the player can choose the same creature several times by dragging it to more than one slot (if the player could only choose any given creature once, the selected set would be the same as the original set).

Step 5



All slots filled and ready for the next generation

With all the slots filled, the player has the option of going to the next generation. The next generation will be formed by nine new creatures, each one based on the corresponding creature on the slots to the right. Taking the drawing above as example, the new creature on slot 1 on the generation to come will be a possibly mutated version of creature 7, because creature 7 is the one occupying the first slot on the column to the right.

Here we should note that the player was able to drag two creatures into one single slot, instead of just one (on the column to the right containing the selected creatures, notice the slots that have the word "mate" in front of them). In this case, the two creatures will mate in order to make the new creature, and then the

new creature will be subject to mutation just as the ones bred from one single creature.

Proposed method

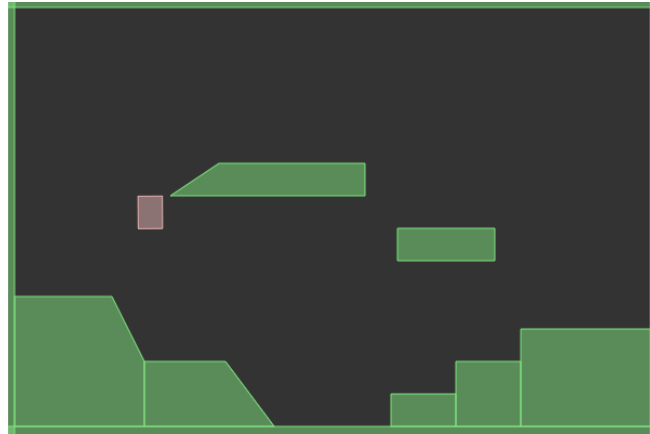
During the development of the idea, openframeworks was used to make a simple prototype using genetic algorithms (the add-on ofxDna, by André Sier, was modified and used [9]) and initially I thought of using openframeworks for the final project. The problem was that on the final project an accurate physics simulation was required and for this I wanted to use Box2D. The add-ons wrapping Box2D using openframeworks are still not quite done yet and I was not satisfied, so at this point I abandoned openframeworks.

My next approach was using the original C++ implementation of Box2D directly and it was working out well (using xcode, I created an additional test to the Testbed application that comes with Box2D and built a basic version of the game's first level on this new test). But at this point I decided I wanted more visibility (people playing the game) than an executable C++ program could offer, so I shifted the project to JavaScript with Box2dWeb using HTML5, so that the game could be played from any modern web browser. Box2dWeb is a port of Box2DFlash 2.1a to JavaScript. Other ports, like Box2DJS, were considered but discarded for reasons such as being a port of an older version of Box2d and not being a self-contained JavaScript file, like Box2dWeb is. [10]

It was also considered developing the game for a mobile platform (either iOS or Android), but in the end this idea was left out, at least as a priority. Even though part of the audience will not be reached due to this decision, the straightforwardness brought by the fact that the game will be playable directly from the browser is intended to compensate it.

The final decision is to implement the project using JavaScript and Box2dWeb in HTML5. At the last stage of the project I will be in a better position to evaluate whether or not I will need some external libraries to help me draw better graphics and put a few simple sound effects on the game.

Firstly I intend to finish the prototype that is being made using JavaScript and Box2dWeb, not worrying about drawing better graphics, applying sound effects or implementing a sophisticated user interface. At the present moment the prototype consists of a small level containing a rectangle representing the jumping creature and some polygons that the creature is supposed to jump on. A two dimensional vector pointing upwards is randomly generated and then used to apply an impulse to the creature's body, thus making it "jump" to a random direction.



Screenshot of the prototype

ToDo's

Phase 1

What has to be done now is the implementation of a class to represent the creature, and more importantly a class to represent the creature's DNA (it should be noted that despite the used of the term "class", what will be actually coded will be a structure that can be somewhat equivalent to an object oriented class, since JavaScript doesn't support classes). Even though I learned a lot by reading and modifying the openframeworks add-on ofxDna, this DNA class will be written from scratch. The idea is to have the DNA data (chromosomes), as an array describing each movement of the creature, which on this stage, will be only jumps, so each node of this array will have a two dimensional vector that describes the angle and magnitude of the jump. The class will also have to provide methods to mate and mutate the creatures. As for a fitness function, it will not be necessary, for the whole point of the game is that the selection is made by the player, and not "naturally" (automatically measuring the fitness), as it is usually the case with genetic algorithms.

Phase 2

The following step will be setting up a data structure to hold the creatures on the current generation and then organizing the loop that will play the simulation of each one moving around the level. After that, a simple interface for the player to choose the creatures that will be ancestors for the next generation will be created. After confirmation from the user the mutation and mating methods will be applied to the selected creatures and the data structure holding the current generation will be updated with the newly created creatures.

Phase 3

Having all this working, it will be time to improve the user interface by providing intuitive drag and drop functionality and menus. Also at this stage the way the graphics are drawn will be redone, and a few sound effects and music will be added.

Milestones

Jan 14	Deadline for setting up the initial posts and layout on the game blog
Jan 21	Deadline for completion of Phase 1
Jan 22	Beginning of Phase 2
Feb 11	Deadline for completion of Phase 2
Feb 12	Beginning of Phase 3
Mar 18	Deadline for Phase 3
Mar 19	Beginning of revision and improvements on the project (including the addition of another level)
Apr 25	End of revision and improvements
Apr 26	Publication of the project online
May 3	Deadline for writing the report on the feedback acquired
May 3	Project deadline

Proposed audience

The target audience is generally casual gamers, for the final software is supposed to be a game with several small levels that shouldn't take too long to go through, thus being suitable for playing during short periods of time.

People interested on genetic simulation, as those who enjoy watching the simulations that take place on the aforementioned BoxCar2D and Cambrian Explosion applications, should also be mentioned as a target audience.

As an example of the target audience I could mention Angry Birds and specially Bad Piggies players. Both these games are fairly casual and physics-based. Another similarity between Bad Piggies and the proposed game is that in Bad Piggies usually most of the interaction between the player and what happens on the game takes place before the action (in it, the player assembles a vehicle that should be capable of, once the simulation is started, crossing the level and reaching an object representing the goal). [11]

The game will be posted on social networks and forums in order to get the desired visibility.

Self-evaluation method

I intend to write a short report commenting on whether the game mechanics proposed work well to make an interesting game. In order to be able to write about this I intend to have at least a couple of levels finished and the game looking polished enough to be shown to the proposed audience. Even though ideally the game should contain many levels and possibly new elements not yet devised, a short demo version of it would possibly be enough to test how interesting the game can be and whether or not it has potential to do relatively well in the casual games scene or arouse some interest on those keen on genetic simulation applications. The aforementioned report will be posted at evolutionnine.wordpress.com, along with all other documents related to the project.

Documenting and archiving the work

The project documentation will be in form of a blog at <http://evolutionnine.wordpress.com>. This blog will work as a journal describing the project development including new ideas, insights and changes that may happen on the way. All the sketches, artwork, code, documents (like this PDF proposal) and other related media will also be posted there.

Online Press Release

FOR IMMEDIATE RELEASE



GENETIC SIMULATION PLATFORM GAME RELEASE DATE Can You Help Them Evolve And Reach the Goal?

London, UK, January 8th, 2013 – A game mixing concepts of genetic algorithms and physics-based games like Rovio's Bad Piggies is being developed by Igor Corrêa as part of university project on Creative Computing. The game is due to be released on April 26 on the project blog: evolutionNine.wordpress.com, where much more information on the project will also be available.

Inspired by genetic simulation applications like the ones found at CambrianExplosion.com and BoxCar2D.com, Evolution Nine will feature nine creatures that jump around aimlessly in a level full of platforms. It is up to the player to help them evolve in order to reach the goal. By selecting the best creatures to form the next generation, the player will gradually make the evolved creatures jump towards the goal.

Mating and mutating

Creatures will not only randomly mutate from one generation to another, they will also be able to mate and create a new creature that has the combined behaviour of their parents.

For more information, check out the game blog at:

<http://evolutionnine.wordpress.com>

Contact: Igor Corrêa

Tel: 07708003783

Email: igorbuenocorrea@gmail.com

###

Biography

Igor Bueno Corrêa was born in Brazil and is keen on computers ever since he first found out about them. After finishing secondary school in 2004, which he did in a small town on the state of Minas Gerais, he made a one and a half year technical course on computing. For the following two years he worked at the taxes department of an accounting office in São Paulo. He learned a lot from this experience, but feeling that his life was not quite on the right path, he left his job and his life in the big city to start a graduation course on Computer Science at PUC - Pontifical Catholic University, back in Minas Gerais. As risky as this move was, it soon proved to be the right decision for Igor. After the first six months on the course he got, due to his grade on a nationwide exam, a full scholarship valid until the end of his course, and also started a paid undergraduate research project on Neural Networks. This project lasted one year and was followed by another one, this time on the computer games area. In late 2011 he started the demanding and tiresome application process to a scholarship programme which led him to cross the Atlantic to study computing at Goldsmiths, University of London, where he will stay until September 2013.

References

- [1] "List_of_genetic_algorithm_applications." Internet:
http://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications
- [2] Leigh Tesfatsion . "How Economists Can Get Alife: Abbreviated Version."
Internet: <http://www2.econ.iastate.edu/tesfatsi/getalife.htm>
- [3] Alex J. Champanard. "NaturalMotion's euphoria Technology: The Honeymoon is Over." Internet:
<http://aigamedev.com/open/editorial/naturalmotion-euphoria/>
- [4] Alex J. Champanard. "Evolving Virtual Creatures: The Definitive Guide."
Internet: <http://aigamedev.com/open/article/evolving-virtual-creatures/>
- [5] Karl Sims. "Artificial Evolution for Computer Graphics." Internet:
<http://www.karlsims.com/papers/siggraph94.pdf>
- [6] Karl Sims. "Evolving Virtual Creatures." Internet:
<http://www.karlsims.com/papers/siggraph91.html>
- [7] "Computation Intelligence Car Evolution Using Box2D Physics." Internet:
<http://boxcar2d.com/>
- [8] Dimitri Bilenkin. "Cambrian Explosion." Internet:
<http://www.cambrianexplosion.com/>
- [9] André Sier. "ofxDna." Internet: <http://ofxaddons.com/repos/1658>
- [10] "box2dweb – A Box2DFlash port to javascript." Internet:
<http://code.google.com/p/box2dweb/>
- [11] Justin Davis. "Vehicular Amusements Abound." Internet:
<http://uk.ign.com/articles/2012/10/02/bad-piggies-review>