BotPrint: Casual Robotic Evolution

Kate Compton, Dylan Lederle-Ensign, Zhongpeng Lin, Joe Mazeika,
Afshin Mobramaein, Johnathan Pagnutti, Huascar Sanchez, Jim Whitehead
University of California, Santa Cruz

Abstract

This paper describes BotPrint, a system that allows novice users to design simple robots. Botprint combines both automated and interactive evolutionary algorithms to assist the players in designing a bot that is optimized by the system yet customized by the user. Players can switch between watching bots compete in a simulated arena, breeding new bots, and selecting and editing any bot. Modifications to a bot can be inserted back into the genetic algorithm, so hand-edited robots can be further evolved and optimized into new variants.

Introduction

Robots are hard to program. Robots with unusual morphologies are much harder to program. How do we help novice players create interesting new robot morphologies, and design new control schemes for them?

This project began with the idea of creating a web app to help kids design a laser-cuttable robot chassis, while helping them with the component layout with automated design techniques. Since a custom chassis is really only interesting if it's an unusual layout, we wanted to encourage users to try out bot layouts other than the familiar two-eyes/two-wheels design. Our tool supports a wide possibility space of robot layouts, so how do we encourage the player to explore it?

Encouraging a fast and fun exploration of a possibility space is the definition of a casual creator, so we can use some of the casual creator design patterns to support this work. (Compton and Mateas 2015 pending) One way to get the user to consider the whole design space is to generate and show them new solutions, a technique called 'mutant shopping'. This technique pairs well with genetic algorithms. The player can see and pick new solutions, and then is presented with more nearby solutions to fine tune their choice. We also use a simulated arena to show off the behavior of the bots while the player is picking them, and editor and breeding modes to allow customization.

BotPrint draws from frameworks for creativity empowering software. We provide a mixed initiative, co-creative interface—one where both the computer and the user show initiative and "take turns generating or modifying the artifacts. Similar interfaces, such as the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013), have been shown to foster creativity.

Related Work

Genetic algorithms are a common solution for procedural content generation. They are an attractive technique which allows the system to sample from vastly different points in the space of potential generative artifacts, but also optimize those points to local maxima.

GAs ability to explore a design space is useful in developing new physical robot designs, such as Iscen, et al's work with evolving tensegrity robots (Iscen et al. 2013). and the modular robots of Hornby et al (Hornby, Lipson, and Pollack 2001) These algorithms can not only discover interesting new robot forms, but can evolve control schema to make them move efficiently. If the heuristic can be evaluated computationally, these algorithms can evolve good solutions very rapidly.

These algorithms can be used for entertainment as well as engineering. BoxCar2D is a free web application, where users can tweak the starting parameters to a GA, and then let it evolve various vehicles. The vehicles success along a generated track is used as the fitness function. Each trial is simulated, so users can watch the vehicles slowly improve their performance and get further and further down the track.

Interactive Evolutionary Computational (IEC) (Lim and Thalmann 1999) allows humans to provide the heuristic evaluation, for situations where the heuristic is not computable, like "I like it" or "it looks good". Dryad (Talton et al. 2009) and Picbreeder (Secretan et al. 2008) use this technique to evolve user-selected trees and abstract images.

BotPrint

BotPrint starts with a genetic algorithm that runs for a limited number of generations before presenting the resulting simple robots to the user. The user can then modify any of the attributes that the genetic algorithm can, including the shape, color, and attachments. These user modifications can range from small tweaks to completely changing the bot. The user's changes are translated back into the generative representation used by the GA, and re-run through evolution. The system and the user take turns evolving and modifying the bot until the users satisfied with the result.

By allowing manual editing of the bots, the genetic algorithm only needs to get close to what a user might consider a highly fit and successful robot—the users can make the final

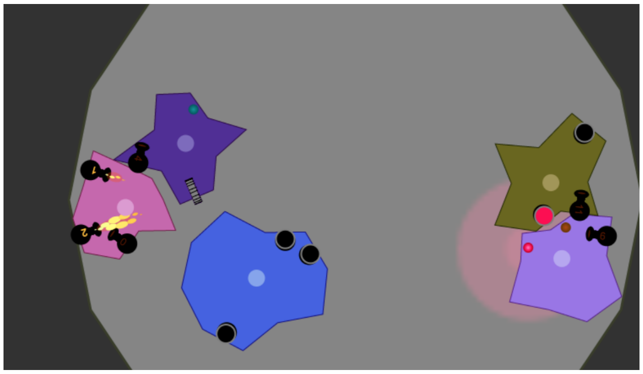

Figure 1: Bots in the Arena

round of tweaks themselves. However, to release the revised bots back into the gene pool, we need reversible phenotype-to-genotype translation, in addition to usual genotype to phenotype (Lim and Thalmann 1999). This was a tricky requirement, but it enables playfulness and experimentation by the user.

System Overview

BotPrint is a multi-layer tool, focused around two main panes: the simulation mode and the editor mode.

The Arena

IEC is typically done in domains where humans can very quickly recognize fitness, such as visual compositions (Secretan et al. 2008). Users must perform at-a-glance evaluation, which can be challenging if they need to imagine an artifact in action. Our solution to this is the Arena, which simulates physics and simple bot behavior. By watching how robots perform in a simulated environment, users can be surprised and enchanted by another robots behavior, and change their mind on what constitutes a 'good' robot.

The bots are modeled as center parts, sensors, and actuators. Center parts include critical parts such as the microcontroller and battery pack, and these are placed in the center of the chassis. When fabricated, these two parts sit on top of the chassis. While users have some degree of control over the placement of these pieces, they are critical to the final robots design and cannot be removed.

The actuators and sensors impact how the robot behaves and simulates. For BotPrint demos, we had a small number of various sensors and servos that could be attached to the side of the robot, although one can imagine a great many more, from various touch and IR sensors to a laser pointer and a holder for a user to stick a pen in so the robot can draw while it drives around.

The Editor

On a double click, a user can 'zoom in' on any robot and switch over to the editor mode. In this mode, we reveal all the vertices that the robot uses to figure out its chassis, as well as let the user add and remove various edge parts. The vertices can be shifted around, pulled out or pushed in, allowing a user to deform the chassis until they find a shape that they enjoy or has whichever properties that they are searching for.

Genetics Mode

Once the user has modified a given robot to their liking, they can return to the arena, where the newly-modified robot can be watched in action. The user then has the option of modifying the robot further - or selecting other robots to edit in an identical fashion. After all of the changes that the user wants to make have been finished, the user can then examine how the robot is performing across a variety of different metrics - from the interesting (max speed or ability to traverse an obstacle course) to the more trivial (color matching). The user selects one of these, and then with a press of a button, the robots are evolved forward ten generations, using their performance along the selected metric as their evaluation. Additionally, the user has the option to choose individual robots to use as the parents for the next generations - in which case, the chosen metric will be ignored.

Conclusions

In conclusion, we think that this system, or others like it, that combine simulation, genetic evolution and real time player interaction will help players come up with new designs.

References

Compton, K., and Mateas, M. 2015 (pending). Casual creators. In *Proc. of the Intl. Conference on Computational Creativity*.

Hornby, G. S.; Lipson, H.; and Pollack, J. B. 2001. Evolution of generative design systems for modular physical robots. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 4, 4146–4151. IEEE.

Iscen, A.; Agogino, A.; SunSpiral, V.; and Tumer, K. 2013. Controlling tensegrity robots through evolution. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 1293–1300. ACM.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *FDG*, 213–220.

Lim, I. S., and Thalmann, D. 1999. *Pro-actively interactive evolution for computer animation*. Springer.

Secretan, J.; Beato, N.; D Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; and Stanley, K. O. 2008. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1759–1768. ACM.

Talton, J. O.; Gibson, D.; Yang, L.; Hanrahan, P.; and Koltun, V. 2009. Exploratory modeling with collaborative design spaces. *ACM Transactions on Graphics-TOG* 28(5):167.