Demonstrating Automatic Content Generation in the Galactic Arms Race Video Game

Erin J. Hastings, Ratan K. Guha, and Kenneth O. Stanley
School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL 32816
{hastings, guha, kstanley}@eecs.ucf.edu

Abstract

In most modern video games, content (e.g. models, levels, weapons, etc.) shipped with the game is static and unchanging, or at best, randomized within a narrow set of parameters. However, if game content could be constantly renewed, players would remain engaged longer. To realize this ambition, the *content-generating NeuroEvolution of Augmenting Topologies* (cgNEAT) algorithm automatically evolves novel game content based on player preferences, as the game is played. To demonstrate this approach, the Galactic Arms Race (GAR) video game, which incorporates cgNEAT, will be presented. In GAR, players pilot space ships and fight enemies to acquire novel particle system weapons that are evolved by the game. The live demo will show how GAR players can discover a wide variety of weapons that are not only novel, but also based on and extended from previous content that they preferred in the past. The implication of cgNEAT is that it is now possible to create games that generate their own content, potentially significantly reducing the cost of content creation and increasing the replay value of games.

Creating the models, levels, textures, and other content that players encounter and possess in games is time consuming and expensive (Irwin 2008). In part to address this problem and to provide additional replay value, it is increasingly popular for developers to distribute tools that enable players to create their own content (e.g. level editors) or to randomize content (e.g. random map generators). However, content creation tools usually require significant effort to master and specialized knowledge beyond that of most players. Moreover, randomization only works if it is tightly constrained to avoid generating undesirable content, and provides no means to deduce the kind of content that players prefer. Thus a more intriguing potential solution is to automatically generate content as the game is played, based on actual player behavior.

To show that such content generation is genuinely possible, this demonstration will introduce the *content-generating NeuroEvolution of Augmenting Topologies* (cgNEAT) algorithm, which automatically generates content for video games. The cgNEAT method, which is an extension of the NEAT algorithm (Stanley and Miikkulainen 2002), creates new game content based on the content players liked in the past in real time through an evolutionary algorithm.

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

To demonstrate the potential of cgNEAT, it is implemented in a video game called Galactic Arms Race (GAR), created at the Evolutionary Complexity Research Group at the University of Central Florida. In GAR, *compositional pattern producing networks* (CPPNs; Stanley 2007), which are a variant of artificial neural networks, genetically encode and control particle system weapons. The CPPNs evolve and increase in complexity through cgNEAT, which tracks which weapons the player fires the most. During the game, weapon behavior becomes increasingly sophisticated while consistently evolving to suit player tastes. In this way, it is the *player* rather than the designer who ultimately implicitly determines what kind of content will populate the game.

Figure 1 shows two lineages of related weapons that evolved during actual GAR game play. It is important to point out that it does not take long for players to begin to find compelling weapons. As figure 2 shows, desirable weapons often arise within the first ten generations and continue to elaborate into novel forms over dozens of generations.

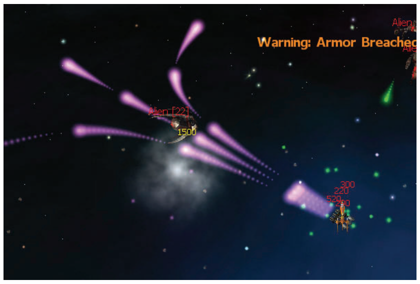
The main result is that players discover a wide variety of content that is not only novel, but also based on and extended from previous content that they liked in the past. While the evolved content in GAR is the weapons, in principle cgNEAT can evolve any class of content in the same way, opening up an exciting new direction in video game design. The implication is that it is possible to produce games and simulations that create their own content to satisfy users, impacting both the production cost and longevity of future such games. GAR will be freely available in spring 2009 at `http://gar.eecs.ucf.edu`.

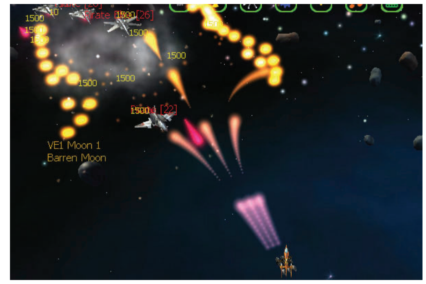
References

Irwin, M. J. 2008. Game developers' trade off. *Forbes.com*.

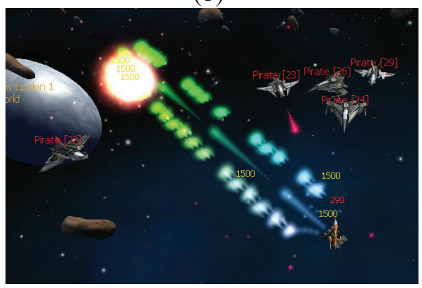
Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10:99–127.

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



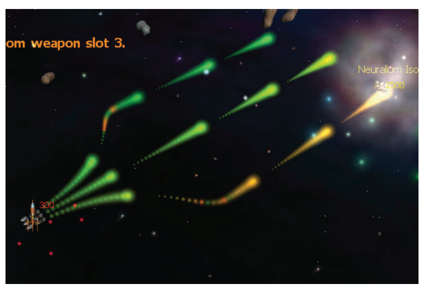


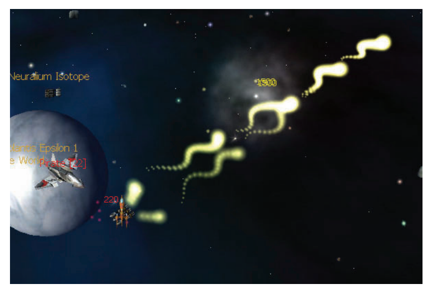
(a) (b) (c)



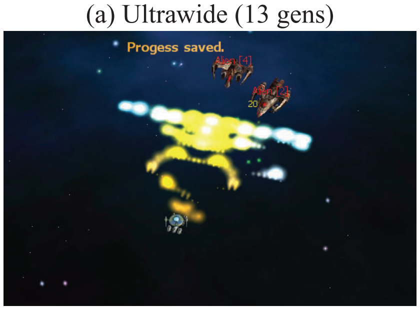
(d) (e) (f)

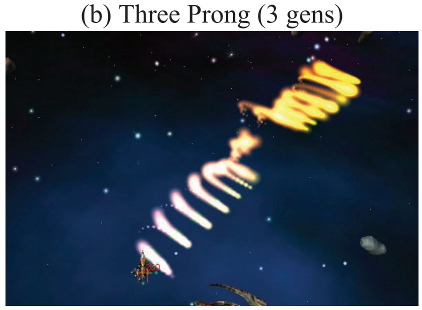
Figure 1: **Weapon Evolution Examples**. As weapons evolve in GAR, players are likely to find weapons with qualities similar to those they favored in the past. In this example from actual gameplay, the player often fired a spread weapon (a). Later in the game, new spread gun variations (b,c) evolved. Another interesting spread gun (d) fires two slower-firing outer projectiles and a fast inner projectile. Later descendants of this weapon (e,f) exaggerated the speed difference between the inner and outer projectiles, diversified the color pattern, and modified the spread width. These examples illustrate how cgNEAT evolves novel content based on past player preferences.

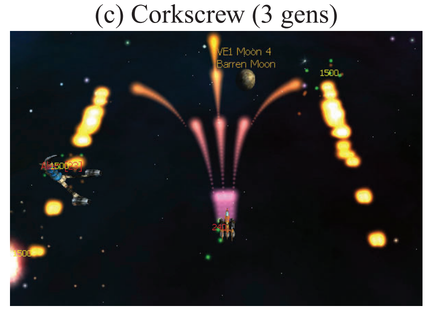




(a) Ultrawide (13 gens) (b) Three Prong (3 gens) (c) Corkscrew (3 gens)







(d) Wallmaker (14 gens) (e) Ladder (35 gens) (f) Tunnelmaker (2 gens)

Figure 2: **Weapons Evolved During Gameplay**. GAR players discovered many original weapons, all invented by the game itself. The number of generations of reproduction taken to evolve each weapon is shown next to its name. The *ultrawide* (a) and *three prong* (b) emit wide particle patterns that are effective for fighting many enemies at once. The *corkscrew* (c) emits a pattern that is initially wide, for blocking, but later converges for concentrated damage at a distance. The *wallmaker* (d) literally creates a defensive wall of particles in front the player. The *ladder gun* (e) fires a wide wave-like pattern that can swivel around obstacles like asteroids. The *tunnelmaker* (f) creates a defensive line of particles on both sides of the player, yielding a defensive sheath. These results demonstrate the ability of cgNEAT to generate a tactically and aesthetically diverse and genuinely useful array of weapons.