

Applying Genetic Algorithms Based on Abstract Data Type (GAADT) for Adaptive Evolution of Electronic Games' Scenarios

Leonardo Filipe Batista Silva de Carvalho¹, Helio Cavalcante Silva Neto¹, Roberta Vilhena Vieira Lopes¹,
Fábio Paraguaçu¹

¹ Masters Degrees in Computational Modeling of Knowledge - Federal University of Alagoas (UFAL)

Postcode 57072-970 – Maceió – AL – Brazil

Abstract

Users' requirements for games with a more realistic behavior have reinforced the importance of artificial intelligence techniques in electronic games. Within the variety of existent game genres, one that stands out as an excellent candidate for the application of such techniques is the RPG genre whose games provide a rich and interactive environment where interesting events are simultaneously taking place, in a similar manner to what happens in the real world. Using this genre as a background, this paper aims to demonstrate the application of a genetic algorithm that can optimize the build of game scenarios, by seeking a desired characteristic for a game map and employing its best efforts to make the transition to such characteristic as gradual as possible, giving as result a cohesive map that has its geographic characteristics as an extra factor to be considered by the players society.

Keywords: RPG, Games, Genetic Algorithms, GAADT.

Author' Contact:

{lfilipebsc, helio.hx, fabioparagua2000}
@gmail.com, rv21@hotmail.com

1. Introduction

The RPG (*Role Playing Game*) genre consists of games in which the players assume the role of a character and create stories in a collaborative way [Adams 2004]. The MMORPG (*Massive Multiplayer On-line Role Playing Game*) is a kind of RPG that, allows the coexistence of thousands of players at the game world simultaneously. The realism and feeling of immersion of the scenarios are of vital importance to the virtual world of a MMORPG, which is experienced as sets of environments, and also, as the players' feeling of being part of something that they perceive as an environment [Christian Anthony et al. 2007].

This paper takes advantage of the complex relationship that players and a MMORPG game scenario share and uses it to test and validate the application of genetic algorithms based on abstract data type (GAADT) [Lopes 2003] over a two dimensional (2D) MMORPG game scenario to change its characteristics.

Now, this paper will focus on explaining GAADT's model and how it was used for the problem of modifying the features of the environment represented in a 2D MMORPG game map. For this, the next section, section 2, will show the basic concepts of procedures and terminologies concerning the application. Section 3 will discuss GAADT's concepts and how they were adapted to work with the problem proposed here. The results of this application and the considerations drew from this will be presented at the last section, section.

2. The Application

This paper aims to acquire the necessary knowledge to manipulate and apply changes to 2D game maps and to evaluate them in order to assert whether or not their use on a MMORPG would be appropriated. Thus, this paper offers an initial exploration through criterions that need constant refined to completely meet the demand for innovation from assiduous game players [Schwab 2004].

For this purpose, the Tiled [2004] application was chose as it consists of a free and open source general purpose map editor that, allows the addition of modules necessary to supply the shortage of data and to implement a GAADT instance to autonomously work the game maps.

2.1. Concepts Inherited from The Application

The application embraces several concepts that can be consulted at its web site [Tiled 2004]. The relevant ones for the proper comprehension of the paper are depicted in Figure 1 and are defined as:

- **Map:** a mesh of x width and y height, with $x \leq y$, or $y \leq x$, and l layers that represents different levels of the map mesh, each of them, with a maximum of n cells that create the a area of the map. In this paper we'll work only with rectangular maps, so that $a = n * l$ and $n = x * y$;
- **Tile:** the entity that occupies the cells of the map mesh and contains an image that can be assigned to one or more of the map cells. In order to keep a coherent map, a tile must have a $(i * w) \times (j * h)$

dimension, with w and h respectively the width and the height of a map cell, and i and j constants so that $1 \leq i \leq x$ and $1 \leq j \leq y$;

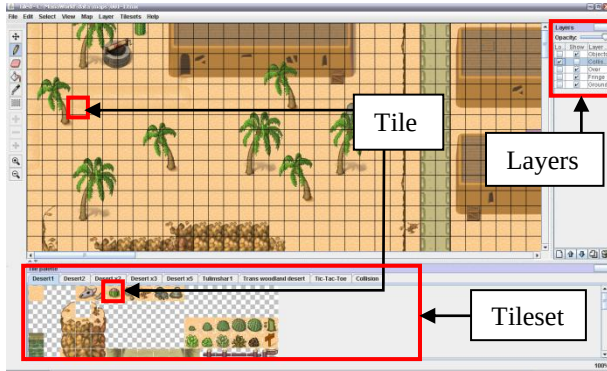


Figure 1: Tiled application, user's graphic interface.

- **Tileset:** It's the image with the set of figures that can be sorted into tiles;
- **Layer:** Level of the mesh in which the map elements, tiles or other objects, are superimposed, allowing them to be "stacked" on top of each other. A map can have a maximum of l layers.

2.2. Map tiles descriptions storage module

As the application's tile object lacks information that identify the contents of each cell of the map mesh, having solely an identification inside the tileset to which it belongs, **id**, and an identification inside the set of tilesets to which it belongs, **gid** (global id); at first, it was necessary to modify the application in order to store into the map the new data that allows it to tell apart the various roles a tile can assume.

Therefore, it was developed the module show in Figure 2, in which a human user feeds descriptions for the tiles, or sets of tiles, supplying the needed data and enabling the application to identify the specific group of a tile (for example, water, tree, grass, etc.), the layer it belongs to and other characteristics unique to the tiles of its group.



Figure 2: Description insertion module

2.3. Map Evolution Module

Once the needed data is provided for the system, the development focused on changing the map characteristics using GAADT. For understanding how this module works its necessary to properly understand

the GAADT. This paper will now focus on explaining how GAADT works, and show how its concepts were adjusted to this kind of model, being such explanations the focus of section 3.

3. Genetic Algorithms (GAADT)

GAADT uses a metaphor regarding its basic genetic material. Thus, the terms basis, gene and chromosome are employed here in representing the inputs for the problem that this paper proposes [Lopes 2003]. Therefore, it should be understood that in this paper, genes are sets over an Ω alphabet satisfying the rules defined by a set of requirements called Axioms for Formation of Genes (AFG). Similarly, chromosomes are sets of genes that satisfy the rules defined by the set of requirements called Axiom for Formation of Chromosomes (AFC).

This paper will now explain the definitions of biological terms that will be in use during the course of this paper, linking them with the problem of representing a 2D MMORPG game map built in the inherited application that was presented in the previous section.

DEF 1. Basis – a basis is defined as $BUDescUb_\lambda$, where B corresponds to a set of terms representing a three dimensional element $\mathbf{b} = (x, y, z)$ corresponding to a point at the x column, y row and z layer of the map, so that $\mathbf{b} \in \mathbb{N}$ with $0 \leq x < 117$, $0 \leq y < 75$ and $0 \leq z < 5$; $Desc$ is the set of descriptions fed into the system using the module of Figure 2 and b_λ is the innocuous basis.

DEF 2. Gene – a gene is a section of the map satisfying the $AFG = \{[AFG_1 = (x_1 < x_2)], [AFG_2 = (y_1 < y_2)], [AFG_3 = (z_1 \leq z_2)], [AFG_4 = (d_1 \in Dub_\lambda)], [AFG_5 = (d_2 \in Dub_\lambda)]\}$. It is made of a set of elements of the same type as the bases, having the form $\mathbf{g} = (b_1, b_2)$, with $b_1 = (x_1, y_1, z_1, d_1)$ and $b_2 = (x_2, y_2, z_2, d_2)$. The gene representation assumes that the b_1 components x_1 , y_1 and z_1 delimit the beginning of the section, and that the x_2 , y_2 , z_2 components from b_2 delimit its end. In addition, d_1 and d_2 are those genes respective descriptions, as fed into the system by the module of Figure 2. A gene having $d = b_\lambda$ is called innocuous gene and represents an empty space or section of the map.

DEF 3. Chromosome – a chromosome is a set of genes that satisfy the $AFC = \{<AFC_1 = \forall a, b \in \{1 \dots n\} [(\forall x' \in [x_{1,a}, x_{2,a}] \nexists x'' \in [x_{1,b}, x_{2,b}] | (x' = x'')] \wedge (\forall y' \in [y_{1,a}, y_{2,a}] \nexists y'' \in [y_{1,b}, y_{2,b}] | (y' = y'')] \wedge (\forall z' \in [z_{1,a}, z_{2,a}] \nexists z'' \in [z_{1,b}, z_{2,b}] | (z' = z'')] \}>, <AFC_2 = (\forall x' \in [0, 117] \exists a \in \{1 \dots n\} | x' \in [x_{1,a}, x_{2,a}]) \wedge (\forall y' \in [0, 75] \exists a \in \{1 \dots n\} | y' \in [y_{1,a}, y_{2,a}]) \wedge (\forall z' \in [0, 5] \exists a \in \{1 \dots n\} | z' \in [z_{1,a}, z_{2,a}]) \}>$; thus, a chromosome is the aggregation of the sections of map defined by its genes. This paper assumes the chromosome as $\mathbf{c} = \langle g_1, g_2, \dots, g_n \rangle$. As presented in the previous definition,

each g_i gene is a set by itself, so in order to distinguish the elements of each gene, a chromosome stores each of them as $g_i = \langle b_{1,i}, b_{2,i} \rangle$, or more extensively, $g_i = \langle (x_{1,i}, y_{1,i}, z_{1,i}, d_{1,i}), (x_{2,i}, y_{2,i}, z_{2,i}, d_{2,i}) \rangle$.

DEF 4. Degree of adaptation of the gene – a function that returns a value corresponding to the quality of a gene in respect to what is expected to achieve as a solution for the problem. Is written as *degree*: $G \rightarrow \mathcal{N}$.

For the particular problem of this paper, depending on what is happening to the scenario the map represents, this function will look for specific characteristics on a gene and is the existence of those characteristics on it that will indicate the degree of its adaptation.

For example, consider that the region represented by the map is currently undergoing a period of drought. As such, it can be assumed that the solution expects to find a higher frequency of sand on the map. In this case, the function will be a function $sand \rightarrow \mathcal{N}$ that corresponds to Equation (1) which is recursively executed for each row and each column of the gene, so that, $\forall g \in G \text{ degree}(g) = sand(x_1, g)$.

It is important to say that, for the problem of looking for specific characteristics on the game map, upon facing a change on the game map other than a drought, the function for the adaptation of the gene will be analogous to the sand function, differing only on the characteristic that will be sought.

$$sand(x, g_2) = \begin{cases} \alpha_1 \text{ if } x_{1,2} < x_{2,2} \text{ and } y_{1,2} < y_{2,2} \text{ and } d_{1,2} \subset sand \\ \beta_1 \text{ if } x_{1,2} < x_{2,2} \text{ and } y_{1,2} < y_{2,2} \text{ and not}(d_{1,2} \subset sand) \\ \alpha_2 \text{ if } x_{1,2} = x_{2,2} \text{ and } y_{1,2} < y_{2,2} \text{ and } d_{1,2} \subset sand \\ \beta_2 \text{ if } x_{1,2} = x_{2,2} \text{ and } y_{1,2} < y_{2,2} \text{ and not}(d_{1,2} \subset sand) \\ 1 \text{ if } x_{1,2} = x_{2,2} \text{ and } y_{1,2} = y_{2,2} \text{ and } d_{1,2} \subset sand \\ 0 \text{ if } x_{1,2} = x_{2,2} \text{ and } y_{1,2} = y_{2,2} \text{ and not}(d_{1,2} \subset sand) \end{cases} \quad (1)$$

$$\begin{aligned} \alpha_1 &= 1 + sand(x, \overbrace{(x_{1,2} + 1, y_{1,2}, z_{1,2}), (x_{2,2}, y_{2,2}, z_{2,2}), d_{1,2}}^{g_2}) \\ \beta_1 &= 0 + sand(x, \overbrace{(x_{1,2} + 1, y_{1,2}, z_{1,2}), (x_{2,2}, y_{2,2}, z_{2,2}), d_{1,2}}^{g_2}) \\ \alpha_2 &= 1 + sand(x, \overbrace{(x, y_{1,2} + 1, z_{1,2}), (x_{2,2}, y_{2,2}, z_{2,2}), d_{1,2}}^{g_2}) \\ \beta_2 &= 0 + sand(x, \overbrace{(x, y_{1,2} + 1, z_{1,2}), (x_{2,2}, y_{2,2}, z_{2,2}), d_{1,2}}^{g_2}) \end{aligned}$$

DEF 5. Adaptation of the chromosome – is a function that measures the chromosome's quality as a result for a problem. It is given by the sum of the adaptation of its genes, i.e., $adapt(c) = \sum_{g \in c} degree(g)$;

DEF 6. Population – a set of chromosomes;

DEF 7. Selection – an operator that corresponds to a set of criteria used in determining which of the chromosomes of the current population will have their genetic material reused during the search for a better result. It works by selecting from the current population chromosomes that satisfy a requirement r that belongs to an Rq set of requirements. This is done using a *sel* predicate of first order logic so that $sel_{r \subseteq \wp(P)}$, having $sel_r(\{c_1, c_2, \dots, c_n\}) = \{c_i \mid \forall i (1 \leq i \leq n) r(c_i)\}$, where $r(c_i)$ indicates that the c_i chromosome satisfies the $r \in Rq$ requirement.

DEF 8. Mutation – is a predicate $mut \subseteq \wp(P)$ that works by applying a function called *change* over a chromosome of the population. The change function gives as result a chromosome c that is the input chromosome for the function having one of its g genes changed by a g' gene, resulting in a new chromosome. This paper express the mutation operator as $mut(c) = \{c' \mid \exists g, g' \in G (c' = chang(c, g, g') \wedge (c' \in p_{cut}(P)))\}$, where p_{cut} is a criterion of acceptance for the chromosomes of an A environment. The change (*chang*) function is so that $chang: P \times G \times G \rightarrow P$ and is assumed for this paper as Equation (2).

$$chang(c, g, g') = \begin{cases} crom((gen(C) - \{g\}) \cup \{g'\}) & \text{if } g \notin gen(c) \\ c & \text{otherwise} \end{cases} \quad (2)$$

The comprehension of these concepts is crucial in understanding GAADT, as the algorithm works by evolving a population made of chromosomes in accordance with the characteristics of an A environment that corresponds to a 8-tuple defined as $\langle P, \wp(P), Tx(P), \Sigma, Rq, AFG, AFC, P_0 \rangle$, where P stands for the current population, that must have at least one chromosome. The other components are the power set of P given as $\wp(P)$; Tx the taxonomy (species and family) of the chromosomes of P , corresponding to the register of the genealogic history of its chromosomes; Σ the set of genetic operators acting over the population P ; as previously defined in this paper, AFG and AFC are respectively the axioms for formation of genes and of chromosomes of the population; the last component P_0 , is a subpopulation of P know as initial population that is unique in $\wp(P)$.

GAADT's last concept regards how the algorithm establishes a criterion that directs the selection of chromosomes of P , cutting out of it the chromosomes that are not supposed to be present in the next population. This criterion must be defined according to the problem, for this problem it is assumed that the average adaptation should not exceed a third of the total area of the map, as to not face players with abrupt changes. For this reason, this value was attributed to the algorithm's p_{cut} criterion, a concept that stands as a cut point defined as a unary predicate acting over the P population chromosomes.

DEF 9. GAADT – once the A environment is assumed, an *AGTAD* genetic algorithm is established as a GAADT function so that $GAADT: A \rightarrow A$ as show by Equation (3). The equation defines the next population, P_{n+1} , as $P_{n+1} = mut(c) \cup Rq(p_{cut})$. It also defines a satisfaction criterion t as the value required by the environment for accepting chromosomes in P_n ; and defines a value k as the criterion standing for the number of interactions

$$GAADT(P_n) = \begin{cases} P_{otm} & \text{if } P_{otm} = \left\{ c / c \in P_{n+1} \wedge \right. \\ & \left. adapt(c) \leq t \neq \emptyset \right\} \\ P_{n+1} & \text{if } n+2 = k \\ GAADT(P_{n+1}) & \text{otherwise} \end{cases} \quad (3)$$

4. Results and Conclusions

The approach discussed in this paper, aims to demonstrate the use of adaptive computation in adding new features to preexistent 2D MMORPG game maps in order to change their scenarios and, thus, generate new maps.

The first results obtained show how the input map of Figure 3 had new features added to it and was consequently changed into the map of Figure 4, in this case, beginning to change from a forest into a desert. An achievement that was possible in reason that enough data about the map was supplied to the algorithm so it could properly handle it. A process that is always susceptible to improvements and innovations, as new perspectives or interpretations are identified and added into the environment.



Figure 3: Map before the GAADT's action.



Figure 4: Map after the GAADT's action.

These initial results show that certain level of refinement is still needed in order for human users properly accept the new scenarios, even though the maps are satisfactory from a strictly functional standpoint. Another improvement would be, to allow the geophysics changes to happen in real time, something not possible at this moment, since the map modifications take place outside the game server, due the fact that the game server would require a great amount of reimplementation to be able to work with

elements that the map editor was specifically created to work with.

Other than geophysics improvements, the NPCs interactions with each other could also be enhanced by linking the scenario changes to the action that triggers them, instead of linking them to the agents responsible for their execution. Such approach would allow the effect of the action to be reflected in the game scenario regardless of whether the execution was the result of the interaction of two or more NPCs to each other, players, or even the interaction of these two kinds of individuals.

References

- Adams, J. 2004. *Programming role playing games: with DirectX Game Development*. Series: The Premier Press game development series. Thomson/Course Technology.
- Christian Anthony, L. Go, Tristan, B., Won-Hyung Lee. 2007. *MMORPG Map Evaluation Using Pedestrian Agents*. Publisher: Springer Berlin / Heidelberg. Springer from: <http://www.springerlink.com/content/j489145xl6up8660/>. [Accessed Saturday, December 08].
- Lopes, R. V. V. 2003. *A Genetic Algorithm Based on Abstract Data Type and its Specification in Z* – Computer Center(Centro de Informática – CI), Federal University of Pernambuco (Universidade Federal de Pernambuco – UFPE).
- Schwab, B. 2004. *AI Game Engine Programming*, 1st Edition, Charles River Media.
- Tiled, Tiled. 2004. a generic tile map editor. Available from: <http://mapeditor.org/>, [Accessed december 14, 2009].