EvoWild: a demo-simulator about wild life

Macarena Mey Rodríguez, Eduardo Palacio Gayoso, *Member, ImaI - OpenLab*

Abstract — **During the last years we can see how AI (Artificial Intelligence) is reappearing because of technological improvements. These improvements make possible the management of large groups of information with acceptable reply times.**

On the other hand, cost reductions in technology make possible that an investigation field like AI becomes to an inversion field closer to scale economies, that's why it'll be economically profitable to invert in this type of applications.

One of the fastest consequences is the AI implantation in a big amount of devices of our environment, cell telephones, palms and of course, in the video game industry.

This is the reason that took us to develop EvoWild, a simulation about wild life that has video game format and tools but at the same time implements AI algorithms like genetic algorithms and reasoning based in cases.

Keywords— **AI (Artificial Intelligence), EvoWild, Genetic Algorithms (GAs), Reasoning Based in Cases (RBC).**

I. Introduction

In the middle of the 70's the first video games appeared, even thought they never were in the market until almost the 80's. The first machines that were sold were very simple, they used to present black and white graphics and little details, all its states were based in rules. Soon, this new way of entertainment became popular**[1][2]**; being such a great success years after when the first consoles came out, they showed an easier manner to play compared to the other machines.

In each generation, video games got better in spectacular graphics and in the artificial intelligence of the enemies, so much that at present they reach great heights of realism in both fields. Moreover, during these years of evolution, we could see a clear division between these two markets: Personal computers and Consoles. Curiously nowadays it seems like both worlds come together again since there are coming up very many video games for them.

Because of today we have access to machines with very acceptable performance to face AI oriented programming tasks and this fact increases the possibility of developing demonstrations cases of real life through graphic simulations, this project will focus on these points and also in developing and applying AI algorithms to simulate life in a wild environment.

EvoWild product is the result of this project.

II. Work Study

One of the first challenges of this project was to make contact with knowledge already forgotten that at the same time don't belong to the activities of a Computer, but as always in this profession, we must learn from other disciplines to do our work. In this case there is a clear sample as we'll see based on parallel studies conduct.

We must be clear: the classification, behaviour, forms of communication and learning of living creatures. We also should study how the natural selection intervenes in evolution and the effects that it produces, as well as the role of mutations in the way they evolve from animals in the real world.

The selected tool among all those studied is XNA and we are going to work on C# as programming language.

XNA is an API devised by Microsoft to develop games, both PC and Xbox360. Technically it is a work environment based on .Net Framework 2.0.

XNA, with freeware versions for non-profit developers, was born with the intention of simplifying the process of creating games and bringing it to all those programmers who always had been overwhelmed by the complexity of existing tools.

From these work studies we will draw conclusions that will help us to develop EvoWild with a reasonable approximation to reality.

III. EvoWild

EvoWild is based on a series of studies to establish a firm basis of the principles of the game.

Among these studies**[3][4][5][6][7]**, we highlight the following:

A. Animal Classification

In this game, it's very important that all the relations and all the species resemble as closely as possible to reality. As we'll see, this will allow us to have a solid base of entities and classes of animals that will take part in the game (see Figure 1).

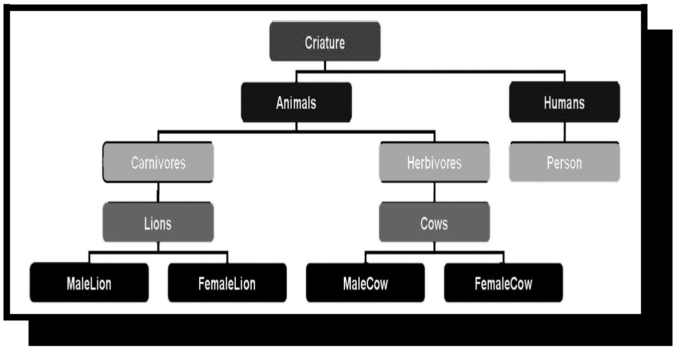

Figure 1 Animals Tree from EvoWild

B. Behaviour

Establish the individuals and families behaviours will make us easier to understand and extend their conducts since they are classified and categorized.

As a conclusion, the animals from EvoWild can have three different states of conduct: aggressive, mild and scared. From our work study we draw the following conclusions for EvoWild:

- Carnivorous:
    - The carnivorous pursues only when hungry and when it gets closer to the victim, then, it attacks.

- Herbivorous:
    - If an herbivorous see a carnivorous that pursues it (running), then, it flees.
    - If an herbivorous see a carnivorous that attacks it, then, it fights trying to defend itself.

C.Communication

One of the most important facets of every living being is communication with their same species and with the environment. The main communication systems to bear in mind are: visual, chemical, hearing, electric and touch.

In this version of the game, we'll only have one communication system and we've chosen the visual one since it's the most representative of all the senses. Our animals will have a certain range of vision that will use to discover the world at the time they walk.

D.Apprenticeship

This is other of the goals that we want EvoWild to have in this version, this way animals that take part in the game will be able to learn in order to provide solutions to their needs.

In EvoWild, animals have a limited ability to remember the world where they are as well as to learn from actions previously performed and its consequences, for example, they can remember if they have drunk certain water from a certain lake and what they have got in return.

To give a learning system of these characteristics to the animals we'll use Reasoning Based in Cases (RBC) described in .

IV.The game

Is focusing on the actions undertaken by animals that are on the board game, that's why it's a game demo format that allows the player to interact with animals selecting and receiving information from the actions and decisions that are taken during the game.

One of the features that make EvoWild to be a game and not a simple demo or a simulation is that it satisfies the requirements of the loop game (see Figure 2)**[8]**. The characteristic of this loop is that it's thought for the player to interact in real time with the game so the loop is oriented to reading the entry that provides the player, while the simulation programs or demonstration are geared to the process be as perfect as possible. EvoWild tries to align the two features in order to offer a game that reflects a certain degree of accuracy about the process of real life.

EvoWild contains the following elements typical of a game:

- Board game: Where all the action takes place in the game, it's composed of the following elements**[8]**:
    - Map game: It defines the space for the development of action.
    - Map Objects: It contains all the objects that interact with the creatures during the game; it contains the inanimate objects of the game.

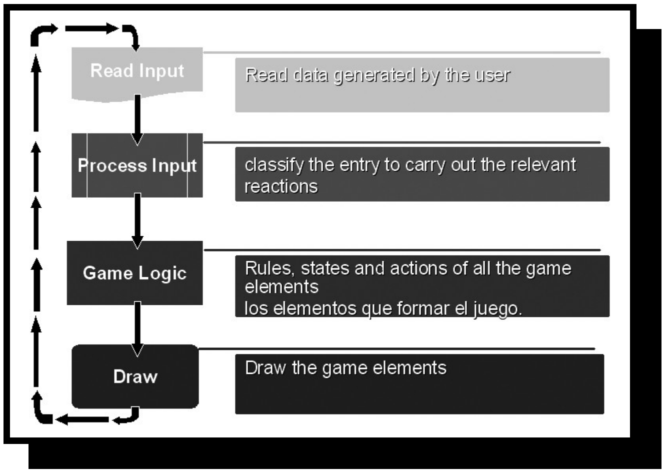
Read Input
Read data generated by the user
Process Input
classify the entry to carry out the relevant reactions
Game Logic
Rules, states and actions of all the game elements
los elementos que formar el juego.
Draw
Draw the game elements

Figure 2 Loop Game

-
    - Map not-pass areas: It delimits areas of the map that may or may not be circumvented by the creatures in the game, as well as the edges of objects such as lakes, rocks, etc…
    - Information to the player area: Zone where the player can consult requesting information by using the mouse.

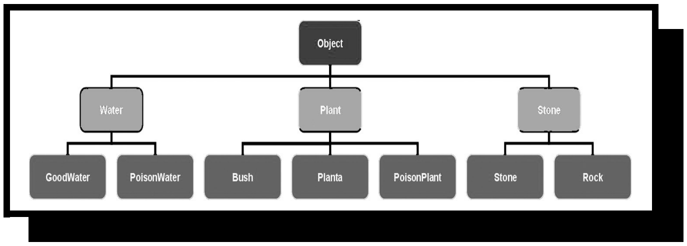

Figure 3 Objects

- Inanimate objects: In this category there are all the objects that belong to EvoWild and do not have the ability to move so they have no graphic animation but they may be alive, like plants for example.

Creatures: They are the main class of EvoWild and they develop all the activity of the game. Under this class we can find all the animals that have been considered in the animal classification (see Figure 1).

V.Creatures from evowild

Without doubt, from the whole game creatures are the elements that have received all the attention during the development and on which we've done the most complex studies because they are the centre on which all the action revolves.

As a result of the studies mentioned in paragraph , creatures have a learning module based on experiences (see), a module that allows their reproduction and creates offspring with common characteristics to the parents (see) like the possibility of moving. All this is present at the "Creature

Loop" that represents how the creatures have to behave and act accordingly (see Figure 4).

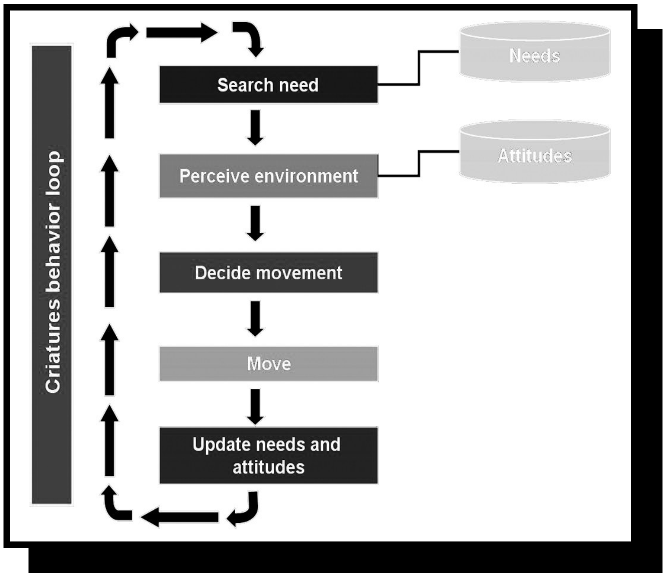

Figure 4 Creature Loop

The “Creature Loop” is divided into five very different functions and, on the other hand, builds and maintains information on the needs and attitudes of the creatures.

We say that the creatures are led by the need since, as we can see in the loop, it is the first function they do and once a creature detects which is its most important need, it’ll be trying to find somewhere in its memory that meets this need. We will call this point the final destination point. The creature will address this point walking from the point where it is, but it won’t do it in a "robotic" way, it will be able to take new solutions and even change the final destination point if on its way it discovers that it can meet the need with a closer object than the one it had chosen previously.

This way the creature will be choosing among different options that perceived using the function of charging the environment, which gives it more benefit. If during the perception of the environment detects a predator or prey, depending on the case and on its status needs, it may take a decision to flee, pursue or continue to its destination without bearing in mind animals in the environment according to detect the attitude of the creatures that surrounded it.

Once explored the environment and with all the information about its attitudes and the needs that it can satisfy, the creature takes a decision that may be any of both: to change its attitude or to satisfy some of its needs, for which it sets the direction where to go calling “moverse()”.

The function “moverse()” gives the creature toward the direction which should turn, we must remember that the creature moves searching to satisfy the need with highest priority. This is the direction that it should take, but on its way, it can recalculate its address short-term (that is the one that its field of perception marks) going to meet needs, trying to escape or attacking, depending on the case.

Once it has walked a certain space (calculated in steps), its needs and its state have got to be recalculated, because depending on how far or near objects that satisfy it needs are, these may change completely. An example of this is that every 50 steps thirst increases, so that if its main need was eating and walking enough without finding what to drink, its priority can change from eating to drinking, and therefore it could require remembering the closest point where sometime earlier it drank.

The idea that the creatures are led by necessity is interesting because very rarely they will be entirely satisfied and therefore downtime will be small.

VI. AI IN EVOWILD

EvoWild is inspired by the original Game of Life**[9]** designed by the British mathematician John Horton Conway in 1970.

This game represents a cellular automaton formed by a board of infinite cells or grilles where there are cells surrounded by eight other cells that evolves according to the so-called "laws of the evolution of Conway", which set conditions for Survival, Birth and Death, depending on the number of cells surrounding a given one.

This game really does not require players during its execution, the actions of the player are limited to select some initial conditions, like the configuration of cells, to start the game. From this moment, the game evolves according to the laws mentioned above until it detects the total extinction of the cells, stabilization total of the cells among a finite set of well-located states or a constant occupation of cells that is not foreseeable but that continues to shape constant.

EvoWild goes one step further and aims to represent the interactions among a finite set of populations of individuals based on the laws of natural selection postulated by Darwin.

A. Genetic Algorithms

Genetic Algorithms (GAs)**[10][11]** are selection algorithms based on methods adaptive.

The basic principles of Genetic Algorithms were established by Holland (1975) and are well described in various texts: Goldberg (1989), Davis (1991), Michalewicz (1992) and Reeves (1993).

They are usually used to find optimal solutions of searching problems. As its name suggests they are based on selecting the best among the possible solutions that exist in a certain moment during the development of finding a solution to the problem, which is very similar to the theory of evolution, Darwin (1859), based on the idea of selection of the most powerful.

This way in the genetic algorithm model we can identify the same words that Darwin used for his model. We have, on one hand, a population of individuals who are interrelated in a given environment. The interrelationships within the population are a way to select or discriminate from the solution the individuals who don’t bring significant improvements to this solution against those who do it.

There are different types of interactions among populations of these individuals; moreover they interact with an environment that has a certain set of qualities which conditions the evolution of a particular kind of individuals. Broadly speaking we could have these sets of populations:

- A population of predators who are responsible for eating a population of prey.

- Another population of preys that require a certain quantity of food provided by the environment in which they live.

The quantity of food is a very important variable in the evolution since the lack leads directly to extinction unless the species in question adapts themselves to another kind of food.

The relationships that will occur in these conditions are often competing for resources in several ways.

- Within a population of individuals, each one is competing for the limited resources, such as food and water that finds within its environment.
- Also within the population, the individuals see the others, to some extent, as finite resources to struggle for, since to mix themselves they look for the best animals, which often form a small group.

This repeated mixture generation after generation and the fact that weaker individuals are prey to predators is what makes to improve the species more and more tending to a solution of individuals becoming more perfect and therefore much closer to the final solution.

For the development we are talking about, we have taken into account the simply genetic algorithm because it is ideally suited to the needs of our game and it's the closest to the actual genetic transfer in living things.

B.Simple Genetic Algorithm

It is an algorithm that simulates very convincingly any phase from the creation of an individual from two parents. Individuals newly created are part of the new generation.

This algorithm explicitly incorporates the theory of the animal selection, developed by Darwin, since during the combination of both parents they choose themselves from the set of the best, as it's shown in Figure 5.

The first action is to select the pair of parents with greater capacity for adaptation to the environment and therefore with more chance of success in their survival. This is a very important factor in the game since it is designed to simulate the adaptation and coexistence in an environment of different animals and see their relations and chances of survival.

1)The code

The genetic algorithms are mainly based on the existence of intrinsic information of an individual's population, this information is the genome.

A genome is a set of grouped values known as chromosomes; this set of heats contains all information of what the individual is and, so far, it doesn't change from his birth to his death.

Within the nomenclature used by biologists, they refer to the genome in two terms: one is the phenotype or genome that is used to combine at the stage of reproduction and the other is the genome that has a mature individual. Generally they should match but because of the interaction with the environment they can cause alterations in the genotype or phenotype that make the individual to show altered characteristics compared to the rest of their kind. These alterations were tested in 1901 by Hugo de Vries in a study work about the offspring of a plant called Oenothera Lamarckiana when he realized that there were giant individuals. These alterations are known as mutations and represent a sudden change in a part of the genome.

```
BEGIN /* Simple Genetic Algorithm */
  Generate initial population.
Compute the individual's evaluation function.
WHILE NOT Finished DO
 BEGIN /* Produce new generation */
  FOR Population size/2 DO
     BEGIN /* Reproductive cycle */
  Select two individuals from the previous
generation, for crossing (probability of selection
proportional to the evaluation function of the
individual).
  Muto two descendants with some probability.
  Compute the evaluation function of the two
mutated descendants.     Insert the two mutated
descendants in the new generation.
    END
IF population has converged THEN
  Finished:= TRUE
  END
END
```

Figure 5 Simple Genetic Algorithm

The changes that occur by mutations in the genome are transmitted only if they are in the phenotype since it is the one that the parents combine.

A change in the genome by a mutation is a change that will be manifested throughout generations and generally they won't be shown from one generation to another.

A genome is represented by a chromosome that the standard model of Simply Genetic Algorithm shows as series of ones and zeros but for us it will be a set of real values from 0 to 100, these values are consistent with the values that animals have and also measures their level of alignment with the current environment.

As we see the Simple Genetic Algorithm is based on series of functions (see Figure 5) that have been adapted to create EvoWild as follows.

In EvoWild, we have isolated the functions of the reproductive cycle and we have implemented them at the stage of the animal reproduction, this way these animals will have a conduct as close as possible to conventional animals.

When an animal decides to reproduce, it looks for a couple in its environment and in the case that there are more than one it chooses the best suited one. It knows this information because when you create new animals you give them a score in the "evaluacionCriatura" attribute.

Once the couple is selected, we call "reproducirse()", this function uses the genome of the couple to call "parir()" where all functions of Simple Genetic Algorithm are performed.

In EvoWild "parir()" (see Figure 6) is the main function of the genetic algorithm because it creates two new genomes for the two offspring (in this version only two offspring are created), the genome of the mother and the genome of the father are mixed, later with little likelihood the genomes of the offspring mutate and finally, it's left to chance the election of the sex and the genome of each animal during the creation of new animals that are inserted into the game.

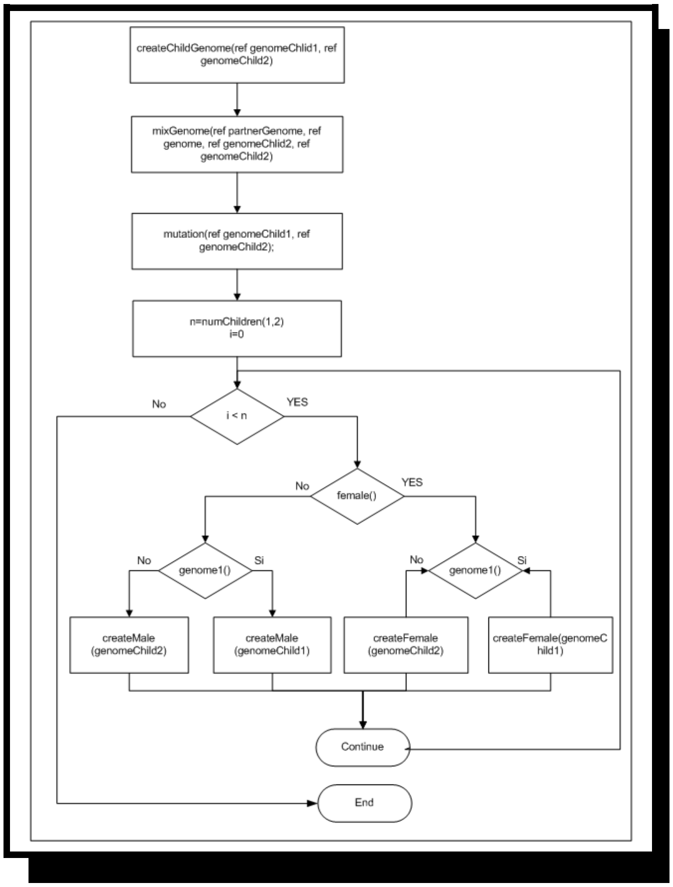
createChildGenome(ref genomeChlid1, ref genomeChild2)
mixGenome(ref partnerGenome, ref genome, ref genomeChlid2, ref genomeChild2)
mutation(ref genomeChild1, ref genomeChild2);
n=numChildren(1,2)
i=0
No
i < n
YES
No
female()
YES
No
genome1()
Si
No
genome1()
Si
createMale (genomeChild2)
createMale (genomeChild1)
createFemale (genomeChild2)
createFemale(genomeChild1)
Continue
End

Figure 6 giveBirth() function

In this function, "parir()", as we have mentioned before the genomes mix. In order to get that, it follows the procedure below.

- The genome is a list of numbers that a parent brings.
- To mix them we choose a position of that list at random, this position divides the list into two sub-lists. One of them will be the head of the genome of one of the sons and the other will be the tail of the list of the genome of the other.
- With the same position, the genome from the other parent is divided into two sub-lists, first and second. The first list will form the top of the list of the genome of a child who earlier got the tail and the second one will be the tail of the child that got the head on the step before.

We already have two strings of genomes that combine the characteristics of their parents. Now, for features to change a little bit and only occasionally, we apply a mutation to the genomes of the two children. This mutation will be applied for each item in the list, but only when they draw a probability over 95%.

C.Reasoning Based on Cases (RBC)

It represents the way of thinking and reasoning that animals have against different problems throughout their lives. RBC systems are based on solving problems using experiences from the past; the overall process of a RBC system is represented in Figure 7. The key concepts of any RBC system are:

- Problem: An adverse situation for which we must give a solution.
- Context: It's the set of circumstances in which the problem takes part.
- Case: It's a problem including its context.

The process that RBC uses to find the solution is:

1. It describes the problem and its context.
2. It recovers from memory a similar problem using the context to select it.
3. If there is a similar case, its solution is proposed as a solution to the problem.
4. If there is no similar case, it tries to give a new solution.
5. In either case, the solution is reviewed in order to see if it was the expected one.
6. Whether it is or it is not the solution, it is stored to be used as a case for subsequent experiences, it becomes a case learned.

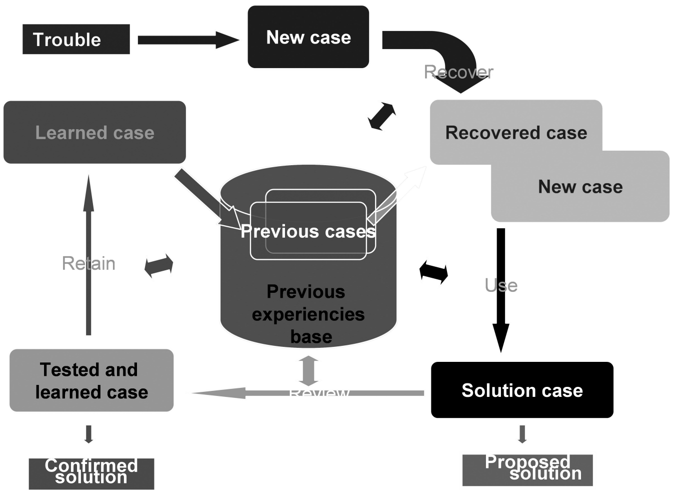

Figure 7 RBC

As we can seen, the most important point about RBC algorithms is to identify a case and its context because the context is what helps us to apply the algorithm in similar situations. Since EvoWild has very well-defined the RBC contexts it's an easy way to give the creatures an apparent form of thinking and reasoning.

The creatures that live in EvoWild have the peculiarity of being able to remember and somehow to think their next action, leaving little room for decision-making by chance.

Creatures of the game move according to a motivation. In the case of EvoWild the main motivation is the need, either to meet shortfalls as thirst or hunger or to flee looking forward to survive when they receive attacks from other creatures, in both cases they use learned behaviours from their past.

To prevent the movement of the creatures by the board are the product of chance or random as we have implemented RBC algorithm reasoning and making some decisions of the creatures, such as when a creature detected several objects in their field of vision RBC used to identify who they are and make a decision that gives you more benefit, as seen in the algorithm decision see title .

The proposed solution for the CBS in EvoWild is the next:

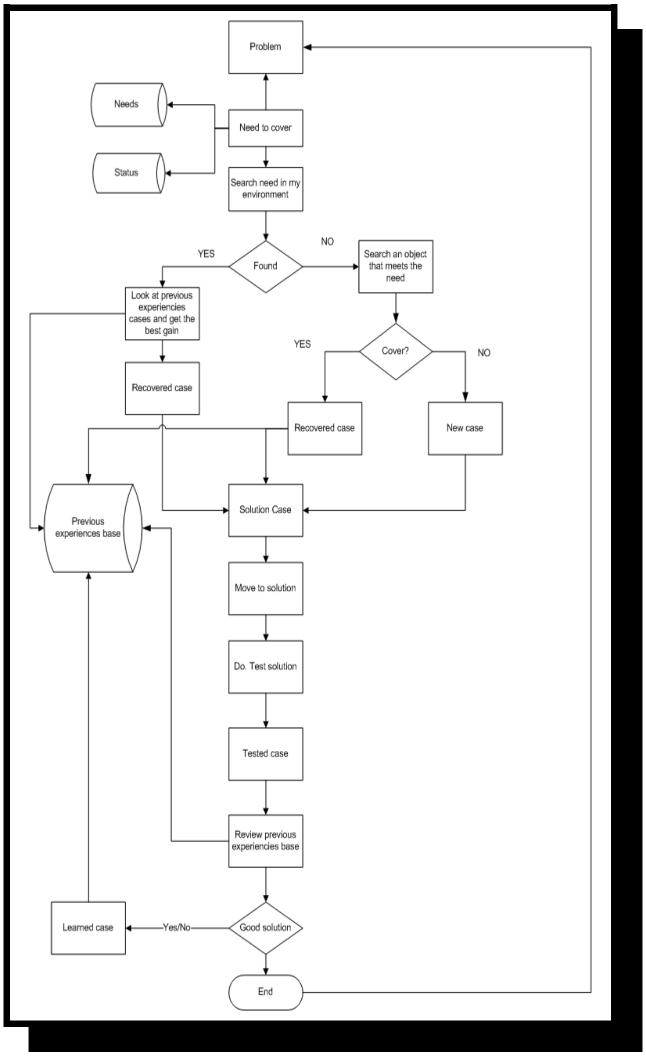
Problem
Needs
Need to cover
Status
Search need in my environment
YES
NO
Found
Search an object that meets the need
Look at previous experiences cases and get the best gain
YES
Cover?
NO
Recovered case
Recovered case
New case
Previous experiences base
Solution Case
Move to solution
Do. Test solution
Tested case
Review previous experiencies base
Learned case
Yes/No
Good solution
End

Figure 8 RBC in EvoWild

VII.CONCLUSION

In EvoWild we have done an approaching task from real-world to computer-simulated-world with a good grade of accuracy; this has been possible because we have used two of the most typical algorithms in AI, even though we have implemented simplified versions of both.

The main conclusion we can draw is the power that this programming technology is able to provide for resolving everyday selection problems and everyday searches with a reasonable cost.

Taking a look at the long history of AI and potholes that have been mainly due to lack of technological resources, another conclusion is that, it seems that nowadays and not long ago, it is the right moment to begin a profitable deployment and development in most of the fields in which it can be used any type of resource related to new technologies.

VIII.FUTHER WORK

Perhaps one of the most interesting areas for development in the future would be to try to have a set of models of ecosystem and to be able to test each of the developments under same conditions.

Another pending point would be to study different types of solutions after making some changes in the model. It could be used as a preventive system of what can happen IF.

It also could be very interesting to make creatures able to adapt themselves to local conditions, changes in eating habits, temperature, radiation…

IX.Thanks

We would like to thank Jesus Soto, Professor of the subject of AI in the Computer Science Faculty at the Pontifical University of Salamanca in Madrid, the help and advices he gave us, with whom it has been possible the adventure of to creating a game that incorporates AI.

Relevant links

[1]ADeSe (2007), Anuario 2006, pp 17-18, http://www.adese.es/pdf/anuario-memoria-2006.pdf

[2]Soaring Nintendo flies into Japan's top three, Leo Lewis, Times Online, http://business.timesonline.co.uk/tol/business/industry_sectors/technology/article2661308.ece, October 15th, 2007

[3]El cuarto Blanco, http://www.educa.aragob.es/iescarin/depart/biogeo/varios/BiologiaCurtis/Indice%20de%20secciones.htm

[4]Biología, La Dinámica De La Vida, Biggs, Alton Editorial: Mc Graw Hill, Año: 2000.

[5]Proyecto Biosfera, Ministerio de Educación y Ciencia, http://recursos.cnice.mec.es/biosfera/

[6]Enciclopedia Encarta, Microsoft Corporation, http://es.encarta.msn.com/

[7]Enciclopedia Libre Universal, http://enciclopedia.us.es/index.php/Mutaci%C3%B3n

[8]Game Architecture and Design: a new Edition, Andrew Rollings, Dave Morris, New Riders, 2004.

[9]Historias de la ciencia: juego de la vida, http://www.historiasdelaciencia.com/?p=126

[10]Algoritmos genéticos y computación evolutiva, Adam Marczyk, 2004, http://the-geek.org/docs/algen/

[11]Algoritmos genéticos, Universidad País Vasco, http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf