Takenoko report - Artificial intelligence

Team Members :

- Jia FU
- Oualid BENAZZOUZ
- Andrei ZABOLOTNÎI
- Cheikh El Wali MALAININE

Supervised by :

- Arnaud MALAPERT
- Philippe RENEVIER

The Scope

The goal of the TER was to continue the work being done on the game Takenoko during the GL/PD course, precisely taking the game into more advanced level.

In this report we will explain how we could achieved an AI robot that tries to accomplish and finish the game in a more smarter and intelligent way, also the methods, performances of our AI.

The game version (PD/GL)

To develop a steady and good AI, we have chosen to modify the latest version of the PD/GL game. The code we are working with have better performances and written in a good, clean, professionnel way then what we had before.

In term of game restrictions, we did some changes as for example :

- we removed the weather and the improvements.
- we decided not to put in play the Objectives of type pattern, but in the other hand we focused more on objectives of type gardener and panda and we did succeed to implement them all.
- the game finishes when a player accomplish six objectives.

And for the game functionalities, we have the essentials to run the game as the original one :

- Game board
- Panda, Gardener
- Plots
- Bamboo
- Objectives (type Gardener and Pada)
- Actions (Moving the panda, moving the gardener, placing a plot, choosing an objective)
- Emperor

AI guaranteed :

We started developing our AI robot by creating two sub AI robots, one will try to do the gardener objectives and the other one will try to do the panda objectives, and at the end we merged them into one AI which is our final robot.

1. Algorithms used :

a. Objectives ranking

The main issue in developing the artificial intelligence in our game was to identify the best solutions in order to find out the best player's action at each state of the game. Each executed action must try to approach the achievement of the objectives more quickly.

To have the opportunity to follow the progress of player's objectives, we decided to implement a method for objectives ranking.

At every game state this method will calculate the ratio of objectives in relation with game's characters positions, plots placed on the game's board and bamboos.

As we have two types of objectives (panda objectives and gardener objectives) we tried to find a common approach to calculate the ratio for both types of objectives. We know that each type of objective requires manipulating the bamboos sections present in the game.

More specific:

To accomplish Panda Objectives, it is necessary to move panda character on the game board, so that it eats the bamboos' sections that are indicated on the objective card.

To accomplish Gardener Objectives, the player has to move the gardener on the game board, so that he grows bamboos' sections on the hexagons in the same combination that is indicated on the objective card.

First we had to find out the value that will increase the ratio of our objectives after each action (eat or grow bamboo's section), so that it realizes a part of conditions requested by the objective card.

We know that one objective requires to eat or to grow $n\ section\ of\ bamboos$. We can deduce that at each time when the players eat or grow the required bamboos sections, we increase objective ratio with value $\frac{1}{n}$.
To achieve one objective of $n\ bamboos\ sections$, the player has to do $n\ successful\ actions \Rightarrow \overbrace{1/n+1/n+...+1/n}^{n}$

We note:

$H -$ *the set of all hexagons that were already placed on the game board.*
$P \subset H -$ *the set of hexagons directly accessible by Panda.*
$G \subset H -$ *the set of hexagons directly accessible by Gardener.*

Panda objectives ratio:

We have established that each time when Panda eats one required bamboo's section, player's objective ratio increases with $\frac{1}{n}$. After we had to add a value that will indicate that from current Panda position he can directly go and eat one bamboo's section. If $\exists h_i \in P,\ i = 0...card(P) - 1$ and on hexagon $h_i\ number\ of\ bamboos > 0\ and\ h_i$ color corresponds with objective color, we increase objective ratio with the value of $\frac{1}{2 * n}$.

In the case when first condition is not true, ranking method check the condition $\exists h_k \in P \cap G,\ k = 0...card(P \cap G) - 1$ and h_k color corresponds with objective color, we increase objective ratio with the value of $\frac{1}{4 * n}$.

We consider $r -$ *number of bamboos' sections required and already eaten by Panda.*
Hence, our ratio formula will have the next form $Ratio = \frac{r}{n} + \frac{1}{i*n} + \frac{score}{100},\ i = 2, 4.$

Gardener objectives ratio:

Also we had to add a value that will indicate that from current Gardener position he can directly go and grow one required bamboo's section.

$n -$ *number of sections required by the objective.*
$t -$ *number of hexagons on which to grow bamboos*, $t = \overline{1,4}$
$H' \subset H -$ *the set of hexagons with the objective color and number of bamboos* $< \frac{n}{t}$
If $\exists h_i \in H',\ i = 0...card(H') - 1\ AND\ h_i \in G$,we increase the objective ratio with the value of $\frac{1}{2 * n}$.

If the first condition is not true, our ranking method will check if $\exists h_k \in G,\ k = 0...card(G) - 1$ and if we move gardener on h_k, we will have access to the required hexagon $h_i \in H' \ AND \ h_i \in G$. In this case we increase objective ratio with the value of $\frac{1}{4 * n}$.

We consider $r - number\ of\ bamboo\ section\ required\ and\ already\ grown.$
So our ratio formula will have the next form $Ratio = \frac{r}{n} + \frac{1}{i*n} + \frac{score}{100},\ i = 2, 4$

b. Simulation

Beside the objective ranking algorithms we used, we are also simulating all possible actions. In order to figure out which action the intelligent bot need to do, we need to find out which action increased the most our average success ratio for all the objectives. So for each action we do a simulation in our board to see what could be the outcomes.

Simulation the panda/Gardener mouvement :

- we move the panda to all possible plots he can visit, and at each plots we calculate the average of the objectives success ratios.
- we keep all the average recorded so later on we choose the plots that got us the maximum average to go to.

Simulation the placing the plots :

- we draw three plots and the try to put each one of the in all free slots in our board, and keep recording all the success ratios averages, so that the plot X that was putted in free slot Y which has got us the highest average ratio of the objectives is the one we are going to take in the real game.

After simulating all the actions, we have the maximum average success ratio at each action, and we simply the action with the highest average is the one the bot will do.

AI performance :

I. Our AI vs random robot

A random robot chooses the actions randomly, after choosing the action, if it's to moving the panda or the gardener he will move them in a random place, if it's to put a plot he will put it in a random free slot, which means it takes time and it's hard for him to accomplish his objectives. On the other side, AI robot doesn't choose action to carry out randomly but plays in a much smarter way and gets to the aim, each step he calculate what's the benefit in term of approaching to accomplish an objective. this is why we see below that the AdvancedAIStrategy is winning by a whopping 97% when they played 1000 game.

	WON	LOST	DRAW	Average score
Player One: RandomStrategy	25 2,50 %	971 97,10 %	4 0,40 %	11
Player Two: AdvancedAIStrategy	971 97,10 %	25 2,50 %	4 0,40 %	34

II. Our AI vs PD/GL "robot"

The PD/GL robot plays its actions randomly but when it's for the action, he tries to move the panda to plots where their is bamboo, if there isn't one he choose a random plot, and for the gardener action he moves the gardener to irrigated plots.

AI robot instead never plays randomly, any action is played with a specific purpose to be attained, this is the PD/GL robot loses the game most of the times, but a little bit better than the Random Strategy in term of losing percentage.

	WON	LOST	DRAW	Average score
Player One: GL/PD Bot	98 9.80%	892 89.20%	10 1.00%	15
Player Two: AdvancedAIStrategy	892 89.20%	98 9.80%	10 1.00%	30

III. Our AI against itself

When AI plays against itself, it's comparable to two human beings playing against each other as their intellectual capabilities are the same. one might outdo the other based on the situation and conditions it finds itself in. this explain the small difference in game won and lost for both players.

	WON		LOST		DRAW		Average score
Player One: AdvancedAIStrategy	515	51,50 %	468	46,80 %	17	1,70 %	26
Player Two: AdvancedAIStrategy	468	46,80 %	515	51,50 %	17	1,70 %	25

Ambitious AI

What we were able to accomplish for our AI compared to the ambitious AI is represented by the fact that the player has become smarter in choosing the actions to carry out and doesn't choose them randomly, but thinks about them in a faster and more intelligent way, the player indeed is able to accomplish the objectives in a very short time and taking the lowest number of actions possible.

The step we still haven't reached is the player ability to counterattack the adversary and stop him from scoring nor does he obstruct him in accomplishing objectives.

In the ambitious AI of course it help evolving the player intelligence, as he will think also about the opponent actions before taking an action.

Ambitious AI performance :

I. Ambitious AI vs random robot

Our aim is to make sure our ambitious AI Bot would never be beaten by a random bot and also make the ambitious AI bot score significantly higher than the random bot.

II. Ambitious AI vs PD/GL "robot"

The goal here is to have a better performance when the current AI played against the PD/GL robot, and just try to finish the game faster and more smarter. while taking into consideration the other robot actions.

III. Ambitious AI against itself

we are expecting that when the ambitious AI plays against itself they finish their games in shorter periods, and using the minimum number of actions to accomplish the objectives.

However for the number of games won and lost by each ambitious AI, it would be similar to our current AI when he played against himself.

Approaches and developpements

During the work on the development of the AI for our version of Takenoko we went through several steps, looking for a more efficient and intelligent way to manage the choice of actions at each move of the game.

As Takenoko is part of a game with incomplete information we did not have the opportunity to use the techniques and algorithms learned in the AI Game and Resolution courses.

We started with simpler methods, the one that did not have a mathematical basis based only on the obvious data of each state of the game such as : the positions of

the panda and gardener, the hexagons in the board and the bamboo sections present.

But these methods didn't give us a chance to really justify the order of actions performed in each round of the game.

This is why we tried to implement the objective ranking system, this allowed us to implement, at each move of the game, a simulation of the actions and calculate the variation of the average of the objectives compared to each simulated action.

Thanks to this simulation we have the possibility to make a more intelligent choice of actions and to approach the achievement of the objectives more quickly, If we talk about results we consider that we have succeeded in developing an AI that really has the highest performance compared to the Random and GL robot.

However we are aware that our AI is not perfect for the following reasons:

1. The AI does not take into account the actions performed by the opponent and analyzes the states of the game based only on its own information.
2. We have not been able to establish the property that will tell us the best time to choose a new objective.
3. At each round of the game the AI simulates one move in advance but to really find the best action we have to simulate several moves in advance.

By achieving the above points we can improve our AI and get closer to a more ambitious AI.