

Introduction

Flappy bird, a small game but made a great success, earned 4 million a year. But this game is really hard to play. I had this game in my phone and I played it for a while. At first I can only get a score like 4 or five. After I found some tricks I can get a score above 10, but my record is just 15. Let me describe how to play this game. This is a 2D game, and there is a bird flying towards the right, the bird is motionless horizontally in the screen, but the background is moving backwards, it makes us feel the bird is flying towards the right. The background is composed by the infinite many pipes, two pipes as a group, one is at the top, another is at the bottom. Because the length of pipes are random, so the gap between them is also random. But the length is not totally random, it must be chosen from a specified range, otherwise it's possible to appear gap equals 0 and totally no gaps. We were talking about the gap between pipes in one group. However the gap from one group to another group is constant. Let's come back to the bird. We have only one choice to control the bird that is tap the screen. Whenever we tap the screen, the bird will jump upwards a little bit, otherwise the bird will drop to the ground very fast. And the another way to lose is crash to the pipe. Therefore, in order to keep the game going, we need to tap the screen at a precise time, so the bird could fly through the gap and avoid crash to the pipe. Here is question, why this game is so hard to play for the most of people? In my opinion, there are many reasons. First the height the bird could jump does not depend on you. And the drop rate is really high, once you made a bad deci-

sion, you die, no chance for making up. Therefore, I came up with an idea, let the computer make the decision, because in this game, whenever you pass through a pair of pipes, there is always a time you should tap the screen to let the bird live, human just can not always tap at the right time because we are human.

Literature review

In paper "Reinforcement Learning- Flappy Bird", the author made an AI technique which lets agents to adapt and learn by a reward-punishment system. I think this is useful for my project and the reward-punishment idea is a good way to achieve the automatic control. In paper "Exploring Game Space Using Survival Analysis" and paper "Obstacles Avoidance with Machine Learning Control Methods in Flappy Birds Setting", they calculated the function how bird fly. When you tap the screen, what height the bird will jump. The distance between pipes horizontally. The gravity and many kinds of velocities. And they also discussed how to avoid pipes based on those numbers. They are essential for my program, I will use these data a lot. In paper "'Tap, tap, flap, flap.' Ludic Seriality, Digitality, and the Finger", it talks about how the finger of human hand related to games such as flappy bird. This paper may not help me for the coding part of my project. But it gives people a better understanding of the differences of playing game between human and computer itself. paper "Dynamic obstacle-avoiding path plan for robots based on a new artificial potential field function", it has a

bigger topic, it is about obstacle avoiding. The author creates a new artificial potential field function. And use repulsive function to achieve flexible control of distance between robot and obstacles. In paper "Flappy Bird AI". The author discussed about what it takes to develop a game that includes an Artificial Intelligence. In his view, it is hard to build an AI for its own game. And the best solution was to try to imagine the best possible path and then make an agent that would be able to follow that path. This paper explore my idea, I was thinking all about calculation, but now, maybe I may just find a good path for my bird in my program. Paper "Adaptive game AI with dynamic scripting" and "Improving adaptive game AI with evolutionary learning" They have a bigger topic that is game AI, they talks a lot about algorithms of game AI and many improved version algorithms. And the Machine Learning in Modern Computer Games. I may not go such deep for this project but they represent many better, more efficient ways to achieve my project.(Try to use bibtex, but I don't know why it does not work, so as you see, I used a complicated way)

Approach

In order to write the AI for flappy bird. First of all, I need the source code of flappy bird. After search on websites, I found a public source code of flappy bird written by python. I haven't use python for a while, when I try to run this code. Python console just flashed for 0.1 sec. After couple hours research, I realised that I need some packages for

python to run a 2D game like this. Packages are pip, six, Pyglet, and Cocos2d. Installing Cocos2d is the most difficult part, at first I downloaded python 3.4 , and I just can not install cocos2d. After search on website and read many answers. I download python2.7, and finally I installed cocos2d. Then I successfully ran the game, press R to restart, press space to jump, as I expected, the game is difficult. After played a while, I try to understand the detail of the source code. There are two main part of the code. The first part are the objects, pipes, bird and ground. Each object has their properties and pictures. The second part included lots of content. Such as how to count score, how to place pipes, reset function, key-press function, and the most important update function, it defines what is collision in this game.

Experiment Design

After understand how the source code work, its time to design my Flappy bird AI. From the source I know that whenever I press space, the program will call the jump function which is to let the bird jump a little bit. So I know all my Flappy bird AI need to do is to call the function : jump. The question I need to solve is when should I call jump. At first, I just generate a random number from 1 to 10, and if the number is smaller than 5 then call jump. And runs each 0.2 second. When I run my code, I found that my bird cannot even pass the first gap. Then I realized that this method is stupid, I need to be smarter to help the bird pass through gaps. So, my idea is first, let the bird try to

fly like a straight line, even bird will fly like a wave but, in general, it is a straight line. To achieve this, we need to know the drop rate and the jump height of the bird. From the code I found that the drop rate is 50 per 0.2 second, and each jump action is 50 height unit. So, I just let the bird jump per 0.2 second, and the bird will fly in a straight wave. And the next step I need to think about is how to make change when the environment is changing. I set the top of bottom pipe as the point I could just pass, because if I just pass the bottom pipe, then I must pass the gap, as I mentioned before, the bird could pass every single gap, it is all about timing. It is impossible that the bird just passes the bottom pipe but still hits the top pipe. And another thing I need to mention is that the top of the bottom pipe is not a point, its an area. So in order to reach that area, I need to adjust the ordinary jump frequency, and once the bird reach the area of top of the bottom pipe, call the jump function.

Result and Analysis

The result for random jump AI is awful, it can barely pass the first gap. But the second AI is much better, normally, my bird could pass 50th pipe, and sometimes could reach 100. The reason I think why my bird can't keep flying is because there are still many cases that I did not consider. In my program, I let my bird jump when it reach the top of bottom pipe, but the motion when bird reach that area could be various, some extreme angle I may not consider. In order to achieve a better

Flappy bird AI, we need to consider more cases, add more useful if statements of jumping. These two results also let me think about what is AI. We cannot deny that random jump program is not an AI. But it is a really bad AI, and my second AI is better, even though when compares with some other Flappy bird AI which could get 2000 scores, mine is still not good enough. So in my opinion, a good AI is a program which consider as many cases as possible. Back to my random jump AI, this one is the AI which consider 0 case.

Conclusion

In general, this project makes me learn a lot and let me think a lot about AI. To finally get my AI work, it is a long way to go. First I found the source code of Flappy bird. And I need to download all the environment such as python2.7, pyglet, cocos2d. It takes a lot of time, I never thought just let the source code run would take such amount of time. Then I read these code, try to understand how these code work, and at the end. I could write my AI. And the reason I say this project let me think a lot is because the reinforcement learning. I know the reinforcement learning from the peer-reviewed article. Based on the reinforcement learning algorithm, keep running it on flappy bird couple hours, flappy bird could almost keep flying and never crash on pipes. But reinforcement learning takes lots of time to learn, at first, reinforcement learning AI is hard to pass the first gap. But if I use my second AI, flappy bird could pass 50th pipe easily. This is the competition between human intelligence and reinforcement learning. The advantage

of human intelligence is learning fast. But may not close to the optimal. However, once we give time for reinforcement learning, even though it may have a slow beginning, in a long run, reinforcement learning will closer to the optimal than human. That's why AlphaGo won.

References

Reinforcement Learning-Flappy Bird
Gatt, Brian
Goldsmiths University of London
2013

Exploring game space using survival analysis
Isaksen, Aaron and Gopstein, Dan and Nealen, Andy
Foundations of Digital Games
2015

Obstacles Avoidance with Machine Learning Control Methods in Flappy Birds Setting
Shu, Yi and Sun, Ludong and Yan, Miao and Zhu, Zhijie
2014

“Tap, tap, flap, flap.” Ludic Seriality, Digitality, and the Finger
Heilmann, Till A
Eludamos. Journal for Computer Game Culture
pages 33–46
2014

Dynamic obstacle-avoiding path plan for robots based on a new artificial potential field function.
Fan, Xiao-Ping and Li, Shuang-Yan and Chen, Te-Fang
Kongzhi Lilun yu Yingyong/Control Theory
2005

Flappy Bird AI
James, Dalton and Begonis, Michael

Adaptive game AI with dynamic scripting
Spronck, Pieter and Ponsen, Marc and Sprinkhuizen-Kuyper, Ida and Postma,
Eric
Machine Learning
pages 217–248
2006

Improving adaptive game AI with evolutionary learning
Ponsen, Marc
2004
school :Citeseer