Game Development using Panda 3D Game Engine

Vipul Rao
Information Technology.
Shah & Anchor Kutchhi Engineering College, Chembur.
Mumbai,India.
vipulraocr7@yahoo.in

Nilakshi Jain
Information Technology.
Shah & Anchor Kutchhi Engineering College, Chembur.
Mumbai,India.
nilakshijain1986@gmail.com

Prashant Rana
Information Technology.
Shah & Anchor Kutchhi Engineering College, Chembur.
Mumbai,India.
rana.prashant10@gmail.com

Prathamesh Mahagaonkar
Information Technology.
Shah & Anchor Kutchhi Engineering College, Chembur.
Mumbai,India.
pratham_m@outlook.com

Vishal Hanchate
Information Technology.
Shah & Anchor Kutchhi Engineering College, Chembur.
Mumbai,India.
vvh019@gmail.com

Abstract— This paper explores the features of panda 3d game engine and the AI algorithm used in creating games. Here we propose the A* algorithm which is used in game development and explain its merits and demerits with other path finding algorithms. We describe the importance of AI in games and even understand how to A* algorithm works and also how to implement A* algorithm in python.

Keywords- algorithms,*artificial intelligence,computationalintelligence;games,path planning.*

*****

I. Introduction

Games have nowadays been synonymous with AI. Since AI based games have been very popular among kids and teens as well as to many professional gamers. It definitely increases the fun factor of the game as well as helps to boost decision making skills of the player playing the game. Good quality game AI increases the entertainment associated with the game as well as brings a great deal of competition in the gaming market. Traditional games such as chess and solitaire has become outdated since people have found several ways to win the game without any difficulty due to no change in the core basics/features of the game. Nowadays computer programs that can defeat their human opponents have been the main subject of AI game researchers. In AI based games, decision making is given utmost importance which affects the storyline of the game and thereby increasing the interestingness measure of the game. Hence AI based games have always stood out against the old traditional games. Strategy based games such as Warcraft and multiplayer games such as FIFA have been highly ranked by gamers. These games have superior AI which demands creativity from players while playing and has thus won hearts of many gamers. Thus FIFA and Warcraft games have become household names for gamers playing across continents. AI research along with video games has thus formed a lethal combination in gaming industry. The most researched application of AI in games is nothing but creation of non player characters.

Other uses of AI are Interactive storytelling, cooperative behaviors, player control, and content creation.[1]

Interactive Storytelling

It is quite popular yet has very little commercial success. Here we dynamically generate story-lines that are bound to be interesting to the player.

Cooperative Behaviors

It requires a good AI engine for the individual actors in the first place. It is useful for serious games, such as training simulations that involve multiple people coordinating together (e.g. fire drills or police investigations).

Player Control

This area is still under-research and holds lots of promise for improving the player’s experience. The idea is to build the whole player logic with a robust AI system so that the avatar would behave more intelligently and remove the frustrations of traditional reactive behaviors (running into walls, not doing the right thing, etc.)

Content Creation

Next-gen titles need huge amounts of time and money to create. We use similar AI techniques to create more behaviors with less time, and apply them to creating levels. For example, generating cities, buildings, rooms, or any kind of terrain that’s interesting for the player.

II. PANDA 3D GAME ENGINE

Panda3D is a 3D game engine which involves a library of subroutines for 3D rendering and game development. The library is written in C++ along with set of Python bindings. A game can be developed using Panda3D by writing a Python program that controls the Panda3D library. Its design emphasis is on supporting a short learning curve and rapid development. It is considered quite useful whenever deadlines are tight and turnaround time is of the essence.[2]

Panda3D was developed by Disney for their massively multiplayer online game, Toontown. It was released as free software in 2002. Panda3D is now developed jointly by Disney and Carnegie Mellon University's Entertainment Technology Center.[2]

The primary interest of Disney in Panda3D was commercial. They are using it to develop Disney games and amusement-park exhibits. Panda3D is being developed to obtain a fully-featured engine, capable of all the performance and quality one expects in any 'A-grade' commercial title.[2]

Education was the prime reason due to which Entertainment Technology Center supported Disney's Panda 3D.They thought that Panda3D can be well-suited for use in student projects. Bullet proof reliability will be a requirement since students as beginners would certainly cause crashes while operating it. They need to be designed in such a way that students would be able to complete their project in less than a semester and hence a very short learning curve and an engine which has rapid prototyping abilities would help a great deal of students by reducing their burden. [2]

Figure 1. Screenshot of a Flight Simulation Game using Panda 3D game engine.

III. PATH FINDING ALGORITHMS

Path finding algorithm is the most critical aspect of game development since it defines the game play.

This report reflects types of path finding algorithm that are presently used in games. Path finding / Path planning acts as a border between decision making and movement. Path finding can simply be defined as creating paths to reach a certain goal[4].

Path finding algorithms basically use directed non negative weighted graph, a graph where positive weights are given to various paths in graph and every path is associated with a specific direction.

A. A algorithm*

A* algorithm is primarily used for path finding and graph traversal in computer games. A* not only finds path towards the goal but also finds the shortest path towards it. This process of finding the shortest path is done relatively quickly.[6]

A* algorithm works only after the preparation and pre processing of game map. Here the map is broken into different points called nodes that help us in recording the progress of the search.

Each node has 3 attributes-

- Goal which is denoted by g is defined as the cost of reaching the current node from the starting node.
- Heuristic which is denoted by h is defined as an estimate cost from current node to goal node.
- Fitness which is denoted by f is defined as sum of g and h and is also the best estimate cost of the path going through the current node.

Thus the value of f is critical in A* and thus lower the value of f the more efficient the path.

For implementing A* algorithm we need to maintain two lists, an Open list and a Closed list. The open list contains the nodes that are yet to be explored while the closed list contains nodes that have been already explored. Thus nodes represent the state and progress of search.

B. Steps in A algorithm*

The pseudo-code for the A* Algorithm[4] is as follows:

```
1. Let P = starting point.
2. Assign f, g and h values to P.
3. Add P to the Open list. At this point, P is the only node on the Open list.
4. Let B = the best node from the Open list (i.e. the node that has the lowest f-value).
```

```
        a. If B is the goal node, then quit – a path has been found.
        b. If the Open list is empty, then quit – a path cannot be found
  5. Let C = a valid node connected to B.
  a. Assign f, g, and h values to C.
  b. Check whether C is on the Open or Closed list.
            i. If so, check whether the new path is more efficient (i.e. has a lower f-value).
  1. If so update the path.
  ii. Else, add C to the Open list.
  c. Repeat step 5 for all valid children of B.
 6. Repeat from step 4.
```

Since A* algorithm requires large amount of CPU resources thus a large map with too many nodes might cause a slight delay in the game. Thus game designers overcome this problem by tweaking the game so as to avoid the situations.

IV. CONCLUSION

In this paper we have described the features of Panda 3D.Here we have given a brief description of the importance of AI in games. Thus AI plays a crucial role in establishment of present generation video games. AI helps in decision making and thus improves the quality of the game thereby helping the developer create a convivial game for gamers.

Path finding algorithms such as A* algorithm turn out to be propitious while implementing AI based games. We have even formulated steps for implementing A* algorithm in this paper. Hence a perfect blend of Panda3D game engine and A* algorithm can help us to create AI based games.

REFERENCES

[1] Alex J.Champandard ,"Research Opportunities in Game AI", July.28, 2007.[Online].
Available:http://aigamedev.com/open/article/research-opportunities/ [Accessed Dec.21,2014].

[2] Carnegie Mellon Entertainment Technology Center ,"Panda 3D Manual" ,2010-2015.[Online]
Available:https://www.panda3d.org/manual/index.php/Main_Page [Accessed Jan.24,2015].

[3] Flatcoder, "MyGameFast Magazine", 2010-2015.[Online]
Available: www.mygamefast.com [Accessed Jan 25,2015].

[4] Ian Millington,Artificial Intelligence For Games,2nd ed. Reading.[E-book] Available: Amazon

[5] David Brian Mathews, Panda3D 1.6 game engine beginner's guide,1st ed.Reading.[E-book] Available: PacktPub

[6] Ross Graham, Hugh McCabe and Stephen Sheridan School of Informatics and Engineering, Institute of Technology, Blanchardstown, "Path finding algorithm in game development",published in IBT journal

[7] Simon.M.Lucas, "Computational Intelligence and AI in games", published at IEEE Transactions On Computational Intelligence And AI,2009.

[8] Istvan Szita, Marc Ponsen, and Pieter Spronck," Effective and diverse adaptive game AI ", published at IEEE Transactions On Computational Intelligence And AI In Games,Vol 1. No 1.,2009.

[9] Cameron Browne," Elegance in game design", published at IEEE Transactions On Computational Intelligence And AI In Games,Vol 4. No 3., September 2012