Fuzzy logic and A* algorithm implementation on goat foraging games

P Harsani*, I Mulyana, D Zakaria

Computer Science Study Program, Pakuan University, Jl. Pakuan PO Box 452 Bogor 16143, Jawa Barat, Indonesia

*Email: prihastuti.harsani@unpak.ac.id

Abstract. Goat foraging is one of the games that apply the search techniques within the scope of artificial intelligence. This game involves several actors including players and enemies. The method used in this research is fuzzy logic and Algorithm A*. Fuzzy logic is used to determine enemy behaviour. The A* algorithm is used to search for the shortest path. There are two input variables: the distance between the player and the enemy and the anger level of the goat. The output variable that has been defined is the enemy behaviour. The A* algorithm is used to determine the closest path between the player and the enemy and define the enemy's escape path to avoid the player. There are 4 types of enemies namely farmers, planters, farmers and sellers of plants. Players are goats that aims to find a meal that is a plant. In this game goats aim to spend grass in the garden in the form of a maze while avoiding the enemy. The game provides an application of artificial intelligence and is made in four difficulty levels.

1. Introduction

Artificial intelligence is often applied in a variety of games that aim to make the game more interesting, one of which is on enemy behaviour aimed at defeating players. In the AI game there is a goal to calculate the optimization of actors’ behaviour against players. The result must be trustworthy and fun [1]. In the AI game there are two main actions that must be done, namely pathfinding and steering. Pathfinding is how actors can find routes by searching methods in AI. While steering is the method of directing the action based on the environment encountered. One of the games that implement search techniques is Goat Foraging. Game Goat Foraging can be said as a pacman game with the labyrinth arena where the labyrinth is a garden or rice fields and players as a goat that can walk in the labyrinth path. Players in this game have a goal to eat all the grass that is in every lane and also besides it must avoid enemies who try to chase the player, where the enemy in this game is human.

The fuzzy method in the game Goat Foraging is a determinant of the behaviour of the enemy whether to chase players, wait or get away from the players. There are 3 characters to be determined fuzzy variables. The characters are breeders, planters, farmer and seller. The breeder will determine the shortest distance with the goat. Variables used are distance, anger and behaviour. There are four rules to describe the relationship between input and output. The specified variables are distance and level of anger. The method used for calculating the distance between enemies and farmers is the Manhattan method.

There are previous studies related to the use of fuzzy logic and the shortest route search algorithm in the game. Based on previous study [2,3,4] the using of fuzzy logic in the game...
allows game developers to express actor behaviour in a more natural form and ease of
programming because it does not require elaborate math formulation. Application of the A*
algorithm in game applications is also applied in artificial intelligence games. One of them is
the game for pathfinding and steering. Other studies had constructed games with behavioural-
reading approach and A* algorithm implementation are performed by [5,6]. The game was
built using action-RPG. The behaviour was determined by the fuzzy method. The actor whose
behaviour was determined is the attacker and the archer [3]. The next game was a maze game.
The A* algorithm was applied to determine the shortest path the player must pass. The next
research is to make a maze game. The A* algorithm is applied to determine the shortest path.
The player must go through the path to find the solution. The current study is to combine fuzzy
and A* methods to produce a goat foraging game.

2. Methods
This research builds a game that implements fuzzy logic and A* algorithm. Games are built
using a methodology approach to multimedia design. The stages are as follows (Figure 1):

![Figure 1. Steps Games Development](image)

- **Project definition**: In this stage determined the basic elements in the games, namely users,
  content and planning. The game to be built is a game with the type of game platform. The
game platform is a unique game and has a simple game system. Platform games usually require
  gamers to pass through some obstacles to achieve the goal.

- **Creative concept**: This step define the following stages: (i) determination of actors in the
game. The actor in the game is divided into 2, namely the player and the enemy. Players are
  goats that aim to find food, while the enemy consists of farmers, planters, farmers and sellers
  of plants, (ii) game scenario determination. The goat foraging game scenario involves two
  actions to be performed in the AI game, namely steering and pathfinding. Steering is how
  enemy characters behave based on actions performed by players. The behaviours performed
  are presented in Table 1.

  | Enemy Character | Behaviour                                      | Methods            |
  |-----------------|-----------------------------------------------|--------------------|
  | Breeder         | the shortest route of search and the level of anger | Fuzzy rule, A*    |
  | Planter         | Predict the goat route with the level of anger | Fuzzy rule, A*    |
  | Farmer          | the shortest route of search and the level of anger | Fuzzy rule, A*    |
  | Seller          | Behaviours such as breeders and planters       | A*, Randomize      |

2.1. Fuzzy Logic
The three enemy behaviours in this game are catch up, leave and waiting. The scale used is 0-
200. The behaviour will be determined by two inputs. The inputs are distance and anger level.
Distance has a scale of 0 to 50 and the level of anger has a scale of 0 to 100. Fuzzy rule is
different for all enemies. This is based on the behaviour that has been determined on the
scenario. Fuzzy rule for breeders is as follows:

[R1] IF And anger CALM THEN behaviour CATCH UP;
[R2] IF distance AWAY And anger CALM THEN behaviour CATCH UP;
[R3] IF distance CLOSE And anger ANGRY THEN behaviour LEAVE;
[R4] IF distance AWAY And anger ANGRY THEN behaviour WAIT.

Fuzzy rules like these are basically formed with attention to human behaviour. Humans who want to catch a goat will pay attention goat condition. If the goat is angry then people will tend to avoid goats so as not to get hurt, then if the goats are calm then man will try to catch the goat, because that is formed fuzzy rule.

2.1.1. Fuzzyfication. Fuzzyfication is the process of mapping the input values (crisp input) derived from a controlled system (non-fuzzy quantity) into the fuzzy set according to its membership function. The fuzzy set is a fuzzy input that will be processed fuzzy on the next process. To convert input crisp to fuzzy input, first must specify membership function for each input crisp, then fuzzyfication process will take crisp input and compare with existing membership function to generate fuzzy input. Input variable (distance and anger) and output variable (behaviour) are depicted in the fuzzy scale and the membership function is obtained as shown in (1), (2), (3) and (4).

There are 2 input variables, namely distance and anger level. The distance variable consists of 2 fuzzy sets, near and far. For the closest distance is 0, while the farthest distance is 50 meters (1000 pixels). The value of 1000 is the conversion value of many grids to 50 meters if 1 meter is represented by 20 pixels. The value 1000 is the maximum value of the grid size. Distance value 0 means the enemy and the goat are at the same place and farthest is 50 (1000 pixels). The level of anger consists of 2 fuzzy sets, which are angry and calm. The anger level has an interval of 0-100. This value is derived based on human behaviour that will wait before pursuit.

\[
\mu\ CLOSE[x] = \begin{cases} 
1, & x \leq 0 \\
\frac{50-x}{50-0}, & 0 < x \leq 50 \\
0, & x \geq 50 
\end{cases}
\]

\[
\mu\ AWAY[x] = \begin{cases} 
1, & x \geq 50 \\
\frac{x-0}{50-0}, & 0 \leq x \leq 50 \\
0, & x \leq 0 
\end{cases}
\]

\[
\mu\ CALM[y] = \begin{cases} 
1, & y \geq 100 \\
\frac{100-y}{100-0}, & 0 \leq y \leq 100 \\
0, & x \geq 100 
\end{cases}
\]

\[
\mu\ ANGRY[y] = \begin{cases} 
1, & y \geq 100 \\
\frac{y-0}{100-0}, & 0 \leq y \leq 100 \\
0, & y \leq 0 
\end{cases}
\]

The output variable that will be determined membership function is behaviour. This variable consists of three fuzzy sets, leave, wait and catch-up. Three domains are obtained by observing human behaviour in pursuit. There are 3 values, namely 0 for the set of blur, 100 for waiting and 200 to pursue. This value is derived based on human behaviour that will wait before pursuit.

\[
\mu\ LEAVE[z] = \begin{cases} 
1, & z \leq 0 \\
\frac{100-z}{100-0}, & 0 < z \leq 100 \\
0, & z \geq 100 
\end{cases}
\]

\[
\mu\ WAIT[z] = \begin{cases} 
1, & z = 100 \\
\frac{x-0}{100-0}, & 0 < z < 100 \\
0, & 100 < z \leq 200 
\end{cases}
\]

\[
\mu\ CATCHUP[z] = \begin{cases} 
1, & z \geq 200 \\
\frac{z-100}{200-100}, & 100 < z \leq 200 \\
0, & z \leq 100 
\end{cases}
\]
2.1.2. **Fuzzy Implication.** Operator used for fuzzy implication is operator MIN. Every fuzzy rule that has been defined will produce implications. For R1, fuzzy implications are presented in Figure 2. This will apply to all rules.

![Figure 2. Fuzzy implication](image)

2.1.3. **Rule Composition.** The composition of all rules will be generated by applying the MAX method to the implication function (Figure 3)

![Figure 3. Rule Composition](image)

2.1.4. **Defuzzification.** The defuzzification used is the centroid method. The results obtained from fuzzyfication can determine the speed of the enemy chasing the goats for each of its frames [3].

2.2. **A* Algorithm**

The A* algorithm is used to determine the shortest distance between enemies and goats. With the A* algorithm the enemy will run effectively to get to the goats making it difficult for the goats to avoid the enemies who chase them. The A* algorithm is in the process of using nodes that form dots or boxes as determinants of places or paths that can be passed. The A* algorithm is one of the Branch & Bound algorithms, also called an algorithm to search for solutions using additional information (heuristics) in generating the optimal solution [6].

2.3. **Manhattan Heuristic**

Manhattan heuristics are used to determine the distance between enemy and player, the process of calculating the distance using Manhattan is done by summing the horizontal distance of the enemy and the goat with the vertical distance of the enemy and the goat. If the enemy P1 is at the point \((x_1, y_1)\) and the goat P2 is at the point \((x_2, y_2)\), then to calculate the Manhattan distance \(d_M\) between the enemy and the goat is:

\[
d_M = |x_1 - x_2| + |y_1 - y_2|
\]  

Create Storyboard: In this stage is designed things as follows, design a navigation map, define screen layouts, contents and actions, design user interaction controls. This game consist of nine main screen that is opening, menu, menu selection, game, game complete, game ends, help, about and exit. The implementation of this game is a hierarchy.

Application Development: This stage is transforming design into games using Adobe FlashCS6. There are three parts in this stage that is define and implement visual layout of the interface, produce/edit graphics, videos, sound and create controls and insert elements.
3. Result and Discussion

This game has three levels that have a maze and setting different places, the degree of difficulty is distinguished by the types of labyrinth such as shown in Table 2:

| Level  | Branches | Number of Flower | Dead ends |
|--------|----------|------------------|-----------|
| Level 1| 59       | 10 flowers       | -         |
| Level 2| 40       | 8 flowers        | -         |
| Level 3| 34       | 8 flowers, 8 Dead ends |           |

This game has a place in the garden gardens with a fence as a barrier and a light brown path that can be passed by goats and enemies. The player will control the movement of the goat. The enemy will move to pursue the player with the shortest path. Determination of the shortest path using A* algorithm. If players eat flowers then the enemy will run away but this is only temporary because the level of anger will decrease slowly and the enemy will again catch up if the level of anger is up. Players will be deemed victorious if the grass and the existing interest have been exhausted and the player will be deemed lost or game over if the player comes in contact with the enemy. In general, the game structure can be seen in Figure 4.

![Navigation Structure](image)

**Figure 4.** Navigation Structure

Level 1 in this game has a place setting in the garden maze with a fence as a barrier and a light brown path that can be passed by goats and enemies (Figure 5) The game starts when the player presses the direction button on the keyboard, and at that moment the enemy will start to chase the player with the shortest path. If players eat flowers then the enemy will run away but this is only temporary because the level of anger will decrease slowly. The enemy will again catch up if the level of anger is up. Players will be deemed victorious if the grass and the existing interest have been exhausted and the player will be considered defeated. If the player comes into contact with the enemy then the game will be finished. The other levels only have differences in the difficulty levels mentioned in Table 2. Each level will have different settings.

Manhattan distance calculations and A* algorithm implementation use the coordinates as location reference and will use the frame as a character movement. Images and pictures show examples of distance calculations (Figure 6). In the path determination and distance calculation, A* implementation is performed using the grid. (Figure 6). The use of grids in A* implementation is one of the strategies to reduce state space [8]. The larger the size of the grid fed the size of the state space will be greater and further increase the time complexity.
The known coordinates of the enemy are 208,183 and the coordinates of the goats are 887,112. Then the distance can be known as follows:

\[
h(breeder) = |x(breeder) - x(goat)| + |y(breeder) - y(goat)| = |208 - 887| + |183 - 112| = 679 + 71 = 750 \text{ pixels}
\]

Then the distance is 750 pixels, then converted into meters,

\[
\text{distance (meter)} = \frac{750}{20} = 37.5 \text{ meter}
\]

The 1 meter size is assumed to require 20 pixels. This is taking into account the distance on a fuzzy that is limited to 0 to 50 or if in pixels equals 0 to 1000 pixels (50 multiplied by 20). 1000 pixels adjust to the size of the labyrinth of 1152x640 and a maximum resolution of 1280x750.

4. Conclusion
This research builds a platform games. Game rules implemented using fuzzy logic and A* algorithm. Fuzzy logic is used to determine the level of enemy characters and actions performed while the A* algorithm is used to determine the escape path for the enemy and its target. This game is made using Adobe Flash CS6 and action script. This study shows that the games can be made using the concept of artificial intelligence and able to apply the degree of difficulty based on the shape of the path and speed of pursuit by the enemy. To increase game difficulty, fuzzy rule can be added in the determination of enemy characters and target action.

References
[1] Narayek, A. 2004. *AI in Computer Games* Volume 1 Issue 10 Game Development. New York: ACM.
[2] Ha JW, Pyo H and Kim J. 2004. *Fuzzy Logic in Agent-Based Game Design*. IEEE Explore Volume 2. IEEE Annual Meeting of the Fuzzy Information, Processing NAFIPS ‘04. Canada: IEEE.
[3] Pirovano, M. 2012. The use of Fuzzy Logic for Artificial Intelligence in Games.
[4] Hui YC, Prakash EC, and Chaudhari NS. 2004. *Game AI: Artificial Intelligence for 3D Path Finding*, TENCON 2004. IEEE Region 10 Conference, Volume B, 306-309, Thailand: IEEE.
[5] Purba KR, Hasanah, RN, and Muslim, MA. 2013. *Implementasi Logika Fuzzy untuk*
Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG. Jurnal EECCIS, Universitas Brawijaya Vol 7, No 1. Malang: Universitas Brawijaya.

[6] Rengga, D.P., Djuriatno W., and Aswin M. 2013. Pencarian Rute Terdekat pada Labirin Menggunakan Metode A*. Jurnal Mahasiswa TEUB Vol. 1, No. 2. Malang: Universitas Brawijaya.

[7] Simanjuntak, H. and Sigiro, M. 2006. Penerapan Algoritma A* sebagai Salah Satu Algoritma Branch & Bound pada Aplikasi Global Positioning System (GPS). Teknik Informatika STEI ITB. Bandung: Institut Teknologi Bandung.

[8] Hu J, Wan, WG and Yu, X. 2012 Pathfinding Algorithm in Real-Time Strategy Game Based on Unity3D. International Conference on Audio Language and Image Processing 16-18 July 2012 Shanghai. China: IEEE.