Boids Swarm – A Game Mechanism

Anneli Nilsson[1]

Abstract

Boids flocking is a convenient way of using artificial intelligence, also known as AI, to create life-like animated movements in both movies and games. In this project a boids swarm was implemented to act as a game mechanism by creating additional behaviours and features. It resulted in a game mechanic with a modifiable boids flock along with features connected to the AI, for example patrol and attack behaviours. Environment based features were also implemented like obstacle avoidance and swarm areas.

Source code: https://github.com/Fedelmid/TNM095-AI_Project

Author
[1] *Media Technology Student at Linköping University, annni204@student.liu.se*

Keywords: Boids Flock — Artificial intelligence — Game Development

Contents

1. Introduction

Artificial intelligence has been used diligently in video games in various situations ranging from enemy behaviour to animations [1]. Swarm intelligence is a method to create more life-like movements for multiple agents, for example crowd simulation, which is when simulating human behaviours in crowds [2], or task allocations for agents [3]. A *Boids* flock is another example of swarm intelligence, and was first proposed by Raynolds [4] in the 80s. A boid is meant to simulate a bird-like creature, but could also act as different creatures that also has flocking behaviours like a school of fish or a flock of sheep.

The aspect of resusing AI in game development helps developers to conveniently create games with more complex game structure, by using previously created AI assets and tweak accordingly [5]. In this project a Boids flock was implemented and expanded in order to create a dynamic game mechanic.

2. Method

The classic version of a Boids flock from Reynolds abides three different rules that creates the characteristic movements of the flock. These rules are: cohesion, alignment and separation (also known as avoidance). Each boid, an agent of a flock, also have a field-of-view, or FOV, in order to detect were local boids are, for example a narrow FOV will result in the flock moving in a line [6]. The angle, θ, between boids is calculated with (1), where the arguments are two different boids' x- and y-coordinates.

$$\theta = atan2((x_{boid\,A} - x_{boid\,B}), (y_{boid\,A} - y_{boid\,B})) \qquad (1)$$

Cohesion is the rule that steers boids close to each other based on an average position of local boids. See figure 1.

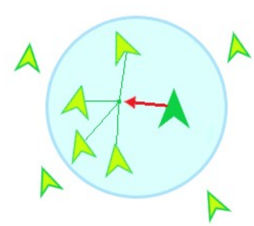

Figure 1. Cohesion flock behaviour.

Alignment changes the orientation of a boid to match the average orientation of local boids. See figure 2.

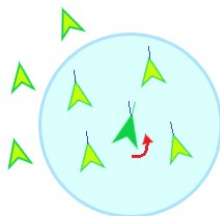

Figure 2. Alignment flock behaviour.

Separation is used to steer a boid away from other boids if they get to close too each other. This rule is the opposite of cohesion and prevents a flock of boids to converge to the same point in space. See figure 3.

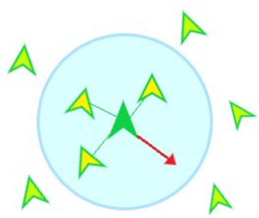

Figure 3. Separation flock behaviour.

The implementation of the flock behaviour was done by calculating the movement vectors according to all three rules for each boid. Then adding them up for a specific boid and multiplied with a corresponding weight, after dividing the vectors by the number of local boids. Lastly, the calculated movement vector will be added to the current boid's transformation.

3. Result

The Boids Swarm game mechanism was implemented with some base features that can be used in a game development. This was implemented in the game engine *Unity 3D*, which means that the code will not work directly in other game engines. However, the principle remains the same and the code can be tweaked to fit other engines.

3.1 The Flock

In this project the different behaviours for the flock was implemented as *scriptable objects*, which in Unity means that it is more efficient to save and store assets during runtime and development. For a set of behaviours one *Composite behaviour*, which is also a scriptable object, is needed for a flock in order to have multiple different behaviours, as well as adding weights for each said behaviour. The minimum behaviours for a classic Boids flock is to have a composite behaviour with cohesion, alignment and avoidance. An alternative cohesion was also implemented, called steered cohesion, in order create a smoother movement transition for agents.

A flock is created by attaching the script `Flock` to an empty game object. This script instantiate all boids in a flock with specific parameters that are modifiable. *Agent prefab* is a reference to the boid model, *Behaviour* refers to a composite behaviour that is customized specific to the flock, *Flock Spawn Point* is a vector position were the flock will instantiate with the number of boids given by the *Starting Count*. The *Drive Factor* and *Max Speed* is connected to boids' movement speed, while the last two *Neighbour Radius* and *Avoidance Radius Multiplier* is about how close boids can get to each other.

It is possible to create separate flocks, that do not join up, in the same space by applying the *Same Flock* filter. Boids that break out from its main flock can later join up again if they cross paths, but will not join other boids if they do not belong to the same flock.

3.2 Features

The features involving a Boids swarm that were implemented within the time frame of this project are shown in table 1.

Table 1. Implemented features.

Object avoidance
Patrol
Hover
Attack
Heal
Speed boost
Follow Player
Safe Zones
Swarm Areas

3.2.1 Object Avoidance

The first feature, object avoidance, is an extension of the separation rule. In this implementation there are three different object avoidances, the first one is one the separation rule which only applies to flock agents. The two other applies to non-boid objects, one avoidance behaviour avoids entire objects while the other avoids the center of objects, see figure 4. A flock will avoid non-boid object as long as they belong to a specific layer which is defined by the developer.

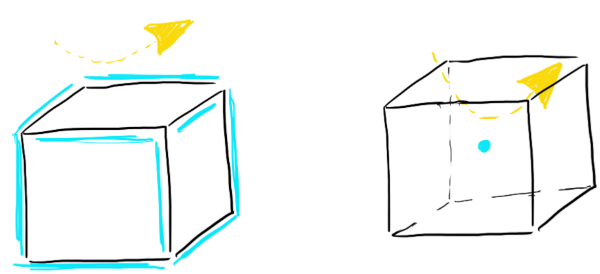

Figure 4. Illustration of the two different obstacle avoidance.

In figure 5 an example of how a flock would avoid an object, which is the green box. The avoidance between agents is also present, because of the separation between individual boids. The composite behaviour of the flock in figure 5 contained in total five behaviours: steered cohesion, alignment, avoidance, swarm area (which is the behaviour to keep the swarm in a designated area) and avoid obstacles.

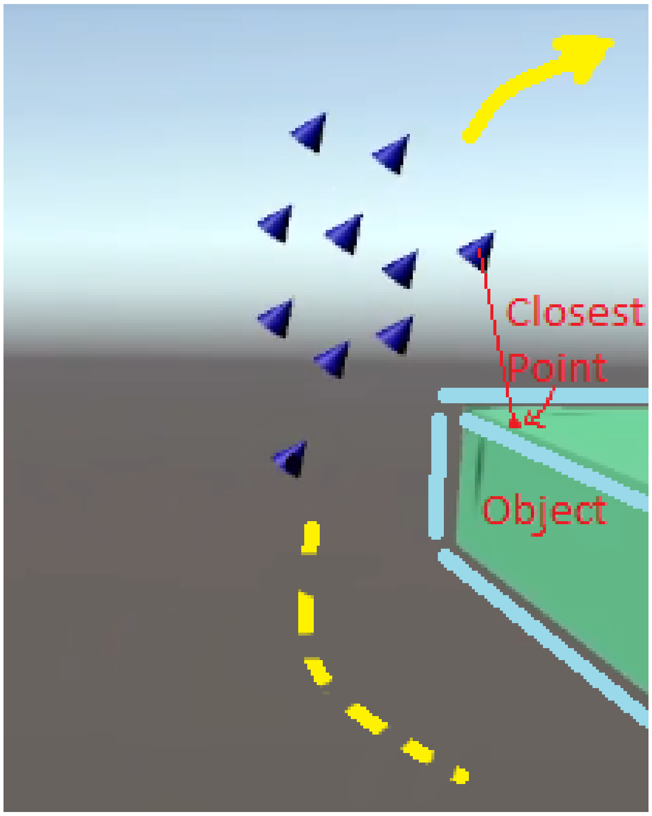

Figure 5. Example of how a boids flock would avoid an object (yellow line), as well as a ray cast by the object towards a boid.

The avoidance behaviour is computed in the boids movements and first checks if the move is necessary based on the number of close objects. It then initialize the movement vector that should be later added to the boid's current movement. After the initialization the behaviour loops over the list of objects that should be avoided, and calculates the closest point between an object and a boid. Depending on which type of avoidance behaviour it is, the closest point is derived differently. If the boid should avoid a point then it will use that specific position as closest point. If the boid should avoid an entire object, then the object casts a ray between itself and the boid position to determine which point on the object is closest to the boid. Figure 5 shows an example of how a ray can be cast towards a boid. If the calculated point is too close to the boid position, then the difference between the positions is added to the new movement vector. After the loop the new movement vector is returned.

3.2.2 Patrol and Hover

The patrol feature is when a flock travels between two different target destinations. The flock's target destination changes after a few seconds, which can be modified. Hover is similar in behaviour as patrol, the difference is that the hover behaviour do not change the target destination.

The behaviours have the same principle as the swarm area, meaning that the flock cannot travel too far from a specific point. It is an extension of the cohesion rule, but considers given points. A boid checks how far it is from a specific point, by calculating the difference between the boid's position and the point. If the offset is too large then it is added to the current boid's movement, in order to correct the distance.

3.2.3 Status Features

Other features that were implemented had a connection to a player character's status. The attack behaviour does what the name suggest, it can deplete the player's health when the player is too close to the swarm. An attack type of flock can be seen in figure 6, where figure 6(a) shows the flock chasing the player and figure 6(b) is attacking. Heal behaviour does the opposite from attack. Both of these behaviours have two parameters that can be modified, the strength of the action and how often the action can happen. The speed boost feature affect the player's speed for a duration of time when the player comes into contact with a flock with this behaviour. The duration of the boost is modifiable. These behaviours react to the player's hit-box in order to trigger the specific actions.

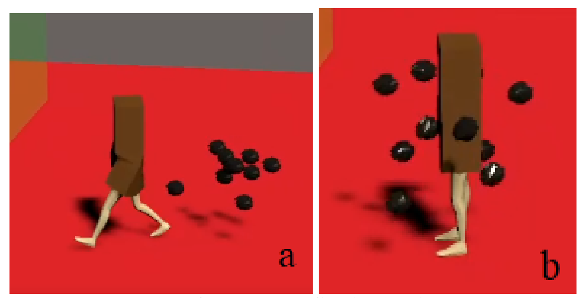

Figure 6. Example of an attack flock chasing (a) and attacking (b) a player.

3.2.4 Environment Type Features

Two environment based features are safe zones and swarm areas. Swarm areas are used to contain a flock within a specific boundary, the shape of these boundaries can either be a sphere or a cuboid. The radius of the sphere and height, width and depth of the cuboid are modifiable. In this implementation two different types of safe zones were made, the first type, type A, is when a player enters the safe zone the following flock looses the player. The other type, type B, does not loose the following flock but prevents the flock from harming the player, see figure 7. The follow player behaviour works when there is an game object in the scene that is tagged "Player" and has a connection to the swarm areas and safe zones, meaning that a flock can start following the player if they enter a swarm area. One thing to note is that if there are multiple game objects in the scene with the "Player"-tag the flock will only follow one of them.

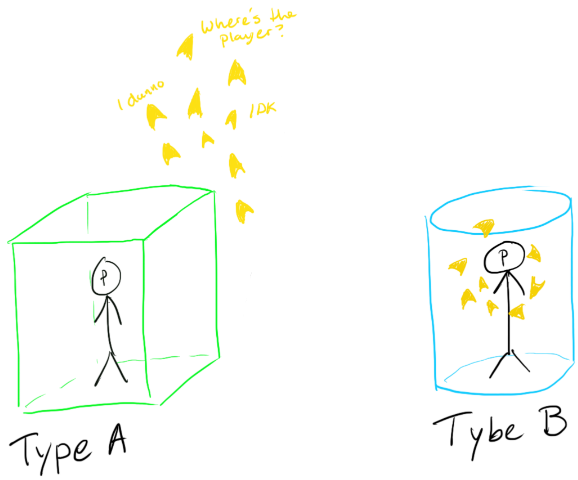

Figure 7. The two different safe zones and their corresponding flock interaction.

4. Discussion

There were a couple features that were implemented in this project which can act as game mechanics in various situations. It was important to make the mechanic modifiable in order to give the developer free hands on how the flock should behave, thus making it more versatile.

One simple way of using the flock is to have it as a background prop, boids were intended to simulate bird-like creatures, however it could also simulate different creatures by tweaking the different parameters. Other types of swarms that could be simulated are for example schools of fish, flocks of sheep, fireflies and so on. Along with the swarm area feature it will make it easier to contain swarms to specific areas within a level and more manageable.

The Hover feature can be used to indicate important items in a game by letting a flock fly around it, which is a way of drawing attention from the user. The follow player behaviour can be used in other situations than being chased by enemies, for example a player needs to collect small flocks in order to solve a puzzle or unlock a door. As for the patrol behaviour it can have different kinds of additional behaviours like attack or heal to create a flock-player connection. It could also act as a guidance for a player in order to indicate where they should go in a level to progress a story or towards some secret.

4.1 Future Work

A game mechanic would benefit from a variety of features and in this project there are a couple additional features that could be implemented in the future. For example, the only boost that was implemented was a speed boost, but this could extend to other types of boosts like a slow-down debuff or an invulnerability timer. Another example is to create a regenerate health boost, which would be a variant of the constant healing that was implemented.

Behaviours that can be improved are the patrol and follow player behaviours. The patrol behaviour could be improved by adding more target destinations that the flock could travel to, in order to create a longer path and/or a more complex path. In this implementation the follow player behaviour only works if there is one game object with the tag ”Player”, this tag reference in the behaviour could be changed to be an developer input in order to modify it easier. Another suggestion would be to create an additional flock follow behaviour with a developer input instead of a hardcoded tag name, thus making it more general.

A different type of safe zone could also be added were the flock hovers around the boundary of the safe zone, instead of being inside the zone. Which would be a mix of the avoid obstacle behaviour and the second type, type B, safe zone.

One unwanted behaviour that can happen to a boid is that it freezes mid-air when the agent wants to rotate towards vector zero in Unity. This problem could possible be solved by re-calculating the move for that specific boid or give it a different position to move towards. However, this will be left for future work.

5. Conclusion

The game mechanism that was implemented in this project obtained animated movements in a convenient way with the help of the Boids AI. It could easily be paired with other behaviours to create more complex flocks, which resulted in different features for the mechanism. The Boids swarm as a game mechanism showcased the versatility of the AI and the possibilities it could achieve within a game environment, which in the end help developers in the development of video games.

References

[1] Georgios N. Yannakakis and Julian Togelius. *Artificial intelligence and games.* Springer, 2018.

[2] Edmond Prakash, Geoff Brindle, Kevin Jones, Zhou Suiping, Narendra S. Chaudhari, and Wong Kok-Wai. Advances in games technology: Software, models, and intelligence. *Simulation Gaming*, 40(6):752, 2009.

[3] A.R. Tavares, H. Azpurua, and L. Chaimowicz. Evolving swarm intelligence for task allocation in a real time strategy game. *2014 Brazilian Symposium on Computer Games and Digital Entertainment, Computer Games and Digital Entertainment (SBGAMES), 2014 Brazilian Symposium on, Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*, pages 99 – 108, 2014.

[4] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.

[5] Wim Westera, Rui Prada, Samuel Mascarenhas, Pedro A. Santos, João Dias, Manuel Guimarães, Konstantinos Georgiadis, Enkhbold Nyamsuren, Kiavash Bahreini, Zerrin Yumak, Chris Christyowidiasmoro, Mihai Dascalu, Gabriel

Gutu-Robu, and Stefan Ruseti. Artificial intelligence moving serious gaming: Presenting reusable game ai components. *Education and Information Technologies*, 25(1):351 – 380, 2020.

[6] Ingemar Ragnemalm. *So How Can You Make Them Scream?: Course book in advanced game programming*. Createspace Independent Publishing Platform, 2017.