Case study : The AI of Horizon zero dawn

- Sadhana (Reshi Krish Thanga Jawahar) EC22185

Introduction

Horizon zero dawn(HZD) is an action adventure game, developed by Guerilla games and launched in 2017 as a playstation exclusive. The game is set in a post-apocalyptic world where humans have been dethroned as the most dominant species, animal-like machines have taken over and humans now live in primitive gatherings/tribes. We are then thrown into this extensive open world and as the story telling progresses learn about the world and how it came to be this way.

Guerilla games as a studio prior to working on HZD had mostly worked on FPS games namely the killzone series and their tools had been tailor fit for that specific genre. All their AI systems from navigation to decision making had been built on the basis of a single human agent, In comparison HZD has 28 different types of enemies with each requiring unique navigation, Animation and Behaviour systems. As such the studio had to build a whole arsenal of new tools and in this case study, I will break down the different AI techniques and systems that Guerilla games built for HZD.

The Machines Ecology

According to the plot of HZD, GAIA is an AI system built to nurture the flora and fauna of the world which later on grew sentience and built Robotic wildlife(Agent), Each designed to fulfil a specific task for the ultimate goal of restoring the world to its natural Glore. Based on the task that each Agent is designed for, They are broadly categorised as:

- Acquisition and scavenger robots
- Recon robots
- Transport robots
- Combat robots

In the game, We can see these agents exhibiting unique behaviours based on whether they are an individual unit or part of a herd. I will first be looking at the AI systems which are used in bringing about this interesting behaviour patterns.

The Collective

The spawning, grouping and management of agents in the game world is managed by an AI system called the collective. The agents are spawned either individually or in groups based on the type of agent, the collective also maintains a list of all the agents present in the world and manages the group's hierarchy. Along with the 28 unique agents, the agents hierarchy plays a key role in making the world of horizon interesting to the player. The agents within a herd are able to share information with each other based on a common blackboard and form unique

co-ordinated behaviours increasing the replayability of the game as each interaction with a herd feels very unique.

Hierarchical task network planner(HTNP)

HTNP is a technique in AI which is used for breaking down a complex goal into primitive actions which can be executed by the agents. This is one of the AI methods which the studio was able to port from their previous titles and is a system which has matured through the course of development of the previous titles from the studio. The agents in HZD can exist independently or as a child to a herd, herds also act as an agent and any of these agents can request an action or goal from HTNP. HTNP checks on the group's dynamic and assigns tasks to each group and also breaks down the overall task of the herd into tasks assigned for each agent.

Movement systems

Each unique agent in HZD has its own size and movement capabilities, with some being able to swim and some able to fly. Even the agents which move exclusively on land also have a lot of variety in their sizes thereby exhibiting different ranges of motion. A key factor that went into the designing of each system as mentioned by Julian Berteling in 2018 is the balance between an optimal solution to a more natural looking solution.

Land based movement

The studio previously having worked on the Killzone series had all its navigation systems based on a waypoint grid system which had to be baked prior to runtime and remained static during runtime, but this was completely unusable when moving to a large open world with various agents having different requirements and having multiple dynamic obstacles in the scene.

Navmeshes

Navmeshes which are generated during runtime were developed for the navigation of agents on land. To cater to all the agents of different sizes and capabilities multiple navmeshes were generated, namely 4 navmeshes based on the size of the agent : small, medium, large and extra large and 2 navmeshes for agents with unique abilities : swimming and mountable. All the information about the environment, such as the obstacles and paths which contain undesirable objects are painted on the navmesh and the pathfinding AI can decide which route to take based on its current state and priorities.

Collision avoidance

The first issue the developers faced when developing agents with collision avoiding capabilities was defining a new type of collision box, as there were many elongated agents in HZD and using a circular collision box left a lot of empty space around the agent. Which then made herd behaviours look very scattered and thus unnatural.
After defining the collision boxes, the agents then had to plan their movements taking into account possible collisions. This was done by using velocity obstacles, where agents which are

in motion have their velocity recorded on the navmesh and other agents can use this information while planning their path. The avoidance is then done by calculating safe velocities and then choosing the velocity based on a scoring function. To reduce the complexity of the calculation for safe velocities only the 5 obstacles with least time to collision are considered. One drawback the team noticed with this avoidance system was that thought it looks good on robots as it finds the optimal path, when used on humans the generation of an optimal path looks more unnatural as in general humans don't necessarily look for optimal paths, are able embrace failure and end up colliding into each other and a system in which all humans move without collision looks unnatural.

Path following

After planning the path to be taken by an agent, a smoothening of the planned path by taking into account the avoidance velocity is done using bezier path smoothening. The generated smoothened path and the original path are then put together to form a hull in front of the agent and the feasibility of navigation on that hull is verified. This resulted in an easy to implement system which was also quite optimised and suited all the agents within the game.
Though this was a working system, it was not a perfect system and had a few issues.

- The maximum turning radius of the agent is not considered and this leads to foot sliding when dealing with sharp turns
- Though not common, sometimes the smoothened path is not possible to navigate and this sometimes causes the agent to rotate in place before following the desired path.

A system to deal with these issues gaining inspiration from how railroads are created was considered but did not make it into the game.

Aerial movement

Aerial movement of agents in HZD uses a different system when compared to land based movement, Instead of navmeshes it takes a form a steering behaviour derived from the A* pathfinding algorithm but also maintains similar performance when compared with its navmesh counterparts.

Heightmaps

A key factor in forming aerial movement is the identification of regions of space in which the agent is able to traverse and for this specific problem the studio had considered multiple methods as mentioned by Wouter Josemans in 2017, Such as Waypoints graphs, Voxel grids, voxel octrees which have been used in other games and also other methods such as multiple navMeshes with cross connectivity and finally ended up using heightmaps and steering.
The generation of heightmaps happens on runtime and for making this generation optimised, The studio had developed a system which generates height maps of different levels of details as required by the agent at each point, 4 levels of resolution were defined for the generation of height maps from level 0 to level 3. This was done similar to how textures of objects present far from the viewer are reduced in resolution, This technique is called mipmaps. One drawback that they had to work around was that there was only one height value for each coordinate, so traversing under arches was not possible under this system.

A* algorithm and Hierarchical path planning

The planning of a path for the agent is done recursively using A* algorithm over a coarse heightmap and then expanded on lower mipmap levels till the budget for amount of iterations is reached. In the A* algorithm the movement of the agent vertically upwards or downwards costs more than movement on the same level, this is done to give paths with the agent gliding around hills have more priority when compared to flying over the hills.
Path smoothing also occurs post path planning as angular movement of agents over the air feels very unnatural. Path smoothing is done via raycasting the starting location of the path to the ending location and when an obstacle is identified in the path, a midpoint is taken along the path and the same process is repeated recursively till a path without any obstacle is found.

Flying behaviours

The studio had to build solutions for agents transitioning from aerial to land based movement and vice versa via different behaviours such as landing, taking off, crashing or attacking the player via dive attacks. This was done by having a separate system which coordinates between the aerial and land systems and generating candidate positions where the transition can occur and switching systems on the decided position.

Animation systems

In most games, Animation is a simple overlay over the algorithms which define behaviours and they get away with the issues that arise as foot sliding and other consistency issues while blending from different animations are not very noticeable. But when coming into HZD where most of the agents are much larger than the player any level of foot sliding other issues are very clearly visible and breaks the immersiveness of the game. To tackle this issue the studio developed AI systems which refer to the animation systems and synchronise the end of animation cycles and action cycles.

Movement systems

Making the agents stop at the exact location derived from the planned path requires a lot of dynamic modification or blending of animation cycles or it can lead to foot sliding. In order to tackle this issue the studio developed a solution in which the movement system references the animation system and decides on suitable points to stop based on when the animations can be optimally blended. This set of points in which animation blending can be accommodated is stored in a motion table, the movement system can refer to this table for performing stop predictions and then stop the movement.

Perception systems

The animations for perception systems, or more specifically the “look at” perception system was built such that it can be dynamically generated. The studio had scrapped the Inverse kinematics system which was originally used as it felt very unnatural and switched to form a bendable rig along the neck of the multiple agent model. The difference between the source orientation and reference rotation was dynamically calculated and is then spread out onto the multiple joints.

Attack systems

Moving from Killzone which had only one form of melee attack in the whole game series to HZD which has huge animations and a variety of attacks for different agents needed the studio to build a completely new system which would handle agent combat interactions. The studio had accomplished this by repurposing the velocity obstacle system created for collision avoidance, agents check for the range in which they will be able to perform their attack from the motion table and when the predicted time for targeted agent to reach the attack range matches with the time taken for the animation the attack is initiated. Some level of animation and location warping is used during the course of the animation to increase the probability of the agent hitting the target.

Conclusion

Guerilla games moving from FPS games to a massive open world set long term goals for building a complete arsenal for AI systems sparing no effort in fine tuning them to give the most immersive and natural experience that could be made possible.
Each AI system has defined goals which it aims to achieve and the studio has made sure that those goals are made:
The behaviour planning(HTNP) and The collective system make the world come to life and make sure that the interactions between each agent is meaningful and well coordinated.
The navigation systems though not perfect with some issues like foot sliding in sharp turns for land based movement and not having support for under arch flying of aerial movement, still makes sure that each agent in the world has navigation mechanics suiting its specific needs.
And with the animation systems, We can see how the developers tied together the multiple systems to their animation counterparts to make sure that the immersiveness of the game doesn't get broken at any point.
Overall, there seems to be only 1 issue concerning the AI systems which has not been dealt with in detail and that would be the AI for human agents but considering the studio had to build almost all their AI systems from scratch for HZD I feel that the studio fulfilled their goal and more by building a strong foundation that they can build upon for further titles.

References

[1]https://www.playstation.com/en-gb/games/horizon-zero-dawn/
[2]https://www.gamedeveloper.com/design/behind-the-ai-of-horizon-zero-dawn-part-1-
[3]http://eej.dk/gain/2017/slides/PUTTING_THE_AI_BACK_INTO_AIR-NAVIGATING_THE_AIR_SPACE_OF_HORIZON_ZERO_DAWN.pdf
[4]https://www.gdcvault.com/play/1024912/Beyond-Killzone-Creating-New-AI
[5]https://www.gamedeveloper.com/design/behind-the-ai-of-horizon-zero-dawn-part-2-