3D Brawler Game Using a Hybrid Planning Approach

Mael Ahmad Addoum*, Maxime Rouffet*, Éric Jacopin†
* International School of Video Game and Animation 3D-FX, *Isart Digital*, Paris, France
†CREC, *Ecoles de Saint-Cyr Coëtquidan*, Guer, France

***Abstract*—We present a hybrid planning architecture that combines both Goal-Oriented Action Planning (GOAP) and Hierarchical Task Network (HTN) planning to avoid the enemies' predictable behaviors and dynamically adapt their actions according to the game-play context. The proposed approach is assessed and illustrated through a 3D custom brawler game where the player character fights invaders by controlling fluid material. Although the planning is computationally expensive, the first results show that our planner is efficient/fast and the game session has enjoyable playability.**

***Index Terms*—3D brawler game, AI Planning, GOAP, HTN.**

I. Introduction

Brawler games represent challenging simulation environments for many decision-making problems that model the behavior of Non-Player Characters (NPC). The most used methods are based on scripting, finite-state machines or behavior trees, and have been successfully integrated into game engines such as Unreal Engine or Unity. However, these methods can suffer from redundant and predictable actions thus reducing player immersion. Furthermore, it is very hard to maintain or evolve these techniques when the state space becomes large. The planning approach overcomes these limitations by providing a better ergonomic game AI architecture, the code is more structured, re-usable, and maintainable. However, the plan construction is time-consuming and requires memory resources [1]. This is a challenge for run-time brawler games since the enemies need to create a plan within a short amount of time and consider the uncertain environment.

We here propose a hybrid planning architecture coupling the GOAP [2,3] and the HTN [4-5] architectures where GOAP attempts to avoid repetitive, boring and predictable behaviors allowing the NPC to dynamically adapt his actions to the current situation of the gameplay [6], while HTN Planning is used to satisfy some requirements of the game design and to speed-up the planning.

In the next section, we describe our game "Kahiora", its planner implementation and the results through different state cases. The last part contains conclusions and future work ideas.

II. Our Approach

A. Brief description of our game

Our custom game is named *Kahiora* and is a dynamic 3D arena battle game offering a rich and storytelling universe. The player controls a character who manipulates a fluid matter

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

Fig. 1. Kahiora: The player character, enemy and the environment scene.

to attack enemies (Spiders or Crabs), see Fig. 1. The killed enemies release some matter, which then lies on the ground and will group up to form matter heaps. The character can absorb it to attack again. The more matter on the ground, the bigger the attack will be.

Kahiora has been developed using Unreal Engine 4 for both PC and PS4 consoles; visit *https://isart-digital.itch.io/kahiora* for a description and to download the game.

B. Planner Architecture

The purpose of our hybrid architecture is to combine the advantages of both GOAP and HTN Planning. Since we are looking to exhibit various, complex and adaptive behaviors for the enemies, GOAP fits our needs in providing non-repetitive and unpredictable actions due to its heuristic property. HTN planner has then been coupled with GOAP as the game designers require performing a specific well-defined action into the generated plan. This coupling first results in patterns of behaviors that the player can learn to better handle combat sequences; and it also allows to respect design norms such as warning the player before an attack. For instance, the crab-NPC should raise its arms or emit red lighting before executing a rush action or launching a Laser attack (Fig. 3), respectively. Moreover, with the HTN, it has been possible to create a task chain of several predefined sub-tasks which are treated as a single task. An example is given in figure 2.

HTNs easily provide this feature thanks to its expressive and intuitive character [7]. We expect also to speed up the planning since the search becomes guided with human-encoded knowledge [8].

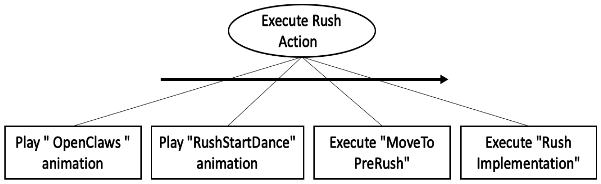

Fig. 2. An example of a task chain in the HTN architecture.

Fig. 3. Frame from the first configuration, the player has a high health level and fights one enemy.

Fig. 4. Frame from the second configuration, the player has a low health level and fights 4 crabs. The red background indicates the bad player life status.

It's worth noting that the development of this game with the PS4 console was a real challenge and constraint. Therefore, using planning with a realistic fluid simulation system can represent a major impact on the game quality. We have thus decided to generate multiple plans and distribute their computation over multiple frames.

AI Planning is implemented as an open-source plugin called *SPlanner* where the designers can easily add, remove or modify behaviors by changing weights and parameters. It can be found on *GitHub* at https://github.com/mrouffet/SPlanner.

C. Results and discussions

To test the performance of our planner, we analyze two situations. First, the player has a high health status and fights one spider (see Fig. 3) while he's damaged and should battle 3-4 spiders in the second situation (see Fig. 4).

The run time to generate the successful plan for the first configuration is neglected when it is only $0.01ms$ for the second situation despite its unfavorable scenario (several NPCs). Therefore, the proposed hybrid approach is efficient and does not lower the game fluidity even though it runs on the PS4.

The frame depicted in Fig. 3 shows that the enemy is enraged by executing a rush action that emits a Laser shoot. However, the enemies in Fig. 4 are less aggressive and in a standby position waiting for the player's action. The first three spiders will attempt to perform a simple attack with less damage contrary to the one in Fig. 3 while the fourth spider is absorbing the liquid matter without threatening the player.

The enemies' behaviors are thus adaptive to the player skills (health status) and the game-play context (number of enemies). That is provided by adjusting the corresponding actions' weights in the GOAP architecture regarding the players' abilities. This adaptive behavior leads then to obtaining both high entertainment and playability in *Kahiora*.

III. Conclusions

For this demo, we developed *Kahiora*, a 3D brawler game that uses a new hybrid planning approach combining both GOAP and HTN Planning. Our first tests showed that our planning system is suitable and neither affect performance nor fluidity. Thus, the player has a good immersion and suspension without having a repetitive feeling thanks to the various and unpredictable enemies' behaviors that adapt to the player's health status.

Other ideas for future work will include applying reinforcement learning techniques within the present planner [9] to dynamically adjust the game difficulty [6] while taking into account further gameplay parameters.

References

[1] John-Paul Kelly, Adi Botea, and Sven Koenig, " Offline planning with hierarchical task networks in video games," *AIIDE*, AAAI Press, pp. 60–65, 2008

[2] X. Huang, C. Du, Y. Peng, X. Wang and J. Liu, "Goal-oriented action planning in partially observable stochastic domains," *IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, pp. 1381-1385, 2021.

[3] C. Stéphane and J. Éric, "Binary GPU-Planning for Thousands of NPCs," IEEE Conference on Games (CoG), pp. 678-681, 2020.

[4] S. Lin, Z. Anshi, L. Bo and F. Xiaoshi, "HTN Guided Adversarial Planning for RTS Games," *IEEE International Conference on Mechatronics and Automation* (ICMA), pp. 1326-1331, 2020.

[5] X. Neufeld, S. Mostaghim and D. Perez-Liebana, "HTN fighter: Planning in a highly-dynamic game," *9th Computer Science and Electronic Engineering* (CEEC), pp. 189-194, 2017.

[6] M. Zohaib, "Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review," *Advances in Human-Computer Interaction*, vol. 2018, 12 pages, 2018.

[7] C. Nguyen, N. Reifsnyder, S. Gopalakrishnan, H. Munoz-Avila, "Automated learning of HTN for controlling minecraft agents," *Conference on Computational Intelligence and Games*, pp. 226-231, 2017.

[8] JP. Kelly, A. Botea, S. Koenig "Planning with Hierarchical Task Networks in Video Games," *American Association for Artificial Intelligence*, 2007.

[9] FF. Duarte, N. Lau, A. Pereira, LP. Reis "A Survey of Planning and Learning in Games". *Applied Sciences*. 10(13):4529, 2020.