Using Goal-Oriented Action Planning within the Context of Turn-Based Combat

Philip White
Falmouth University

FALMOUTH UNIVERSITY

Goal-Oriented Action Planning

Goal-Oriented Action Planning (GOAP) (Orkin 2006) is an artificial intelligence decision-making algorithm for artificial intelligence agents that can make decisions in changing circumstances. A high-level goal is broken down into a series of actions to be executed by the agent. Each action in Goal-Oriented Action Planning has a set of preconditions and effects. The agent assesses the status of the environment, selects a goal to pursue, and then applies a planning algorithm that determines the sequence of actions that would most effectively accomplish that goal while fulfilling the preconditions of each action. Goal-Oriented Action Planning is widely used in the video game industry to develop artificial intelligence that can operate in complex and uncertain environments.

The Artefact

The computing artefact developed to test is a combat artificial intelligence attached to an enemy non-player character for a small turn-based strategy game with a focus on resource management and combo building. In this game, you have a limited set of randomised resources that can be used to build combos to attack an opponent. Combos vary in resource costs and which resource is required for it to be cast you may also use more or fewer resources at random when casting a combo creating an uncertain and dynamic environment that the artificial intelligence must overcome. Additionally, the player is able to steal 1 unit of the most abundant resource the opponent has but is only able to do this once. To win you must defeat the opponent by reducing their health to 0. The combat artificial intelligence must be able to find the optimal set of combos with the current resources that it has on hand and perform those actions to defeat its opponent whilst keeping itself alive.

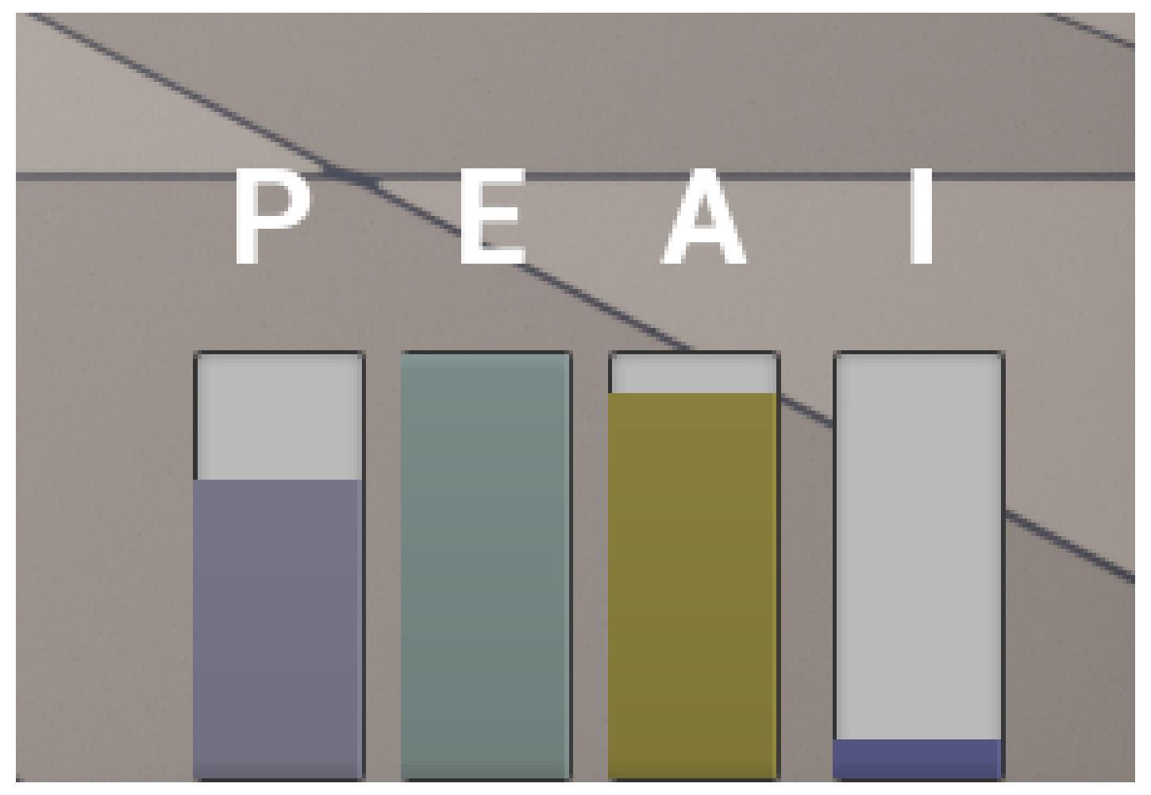


Fig. 1: Screenshot of the randomised resources.

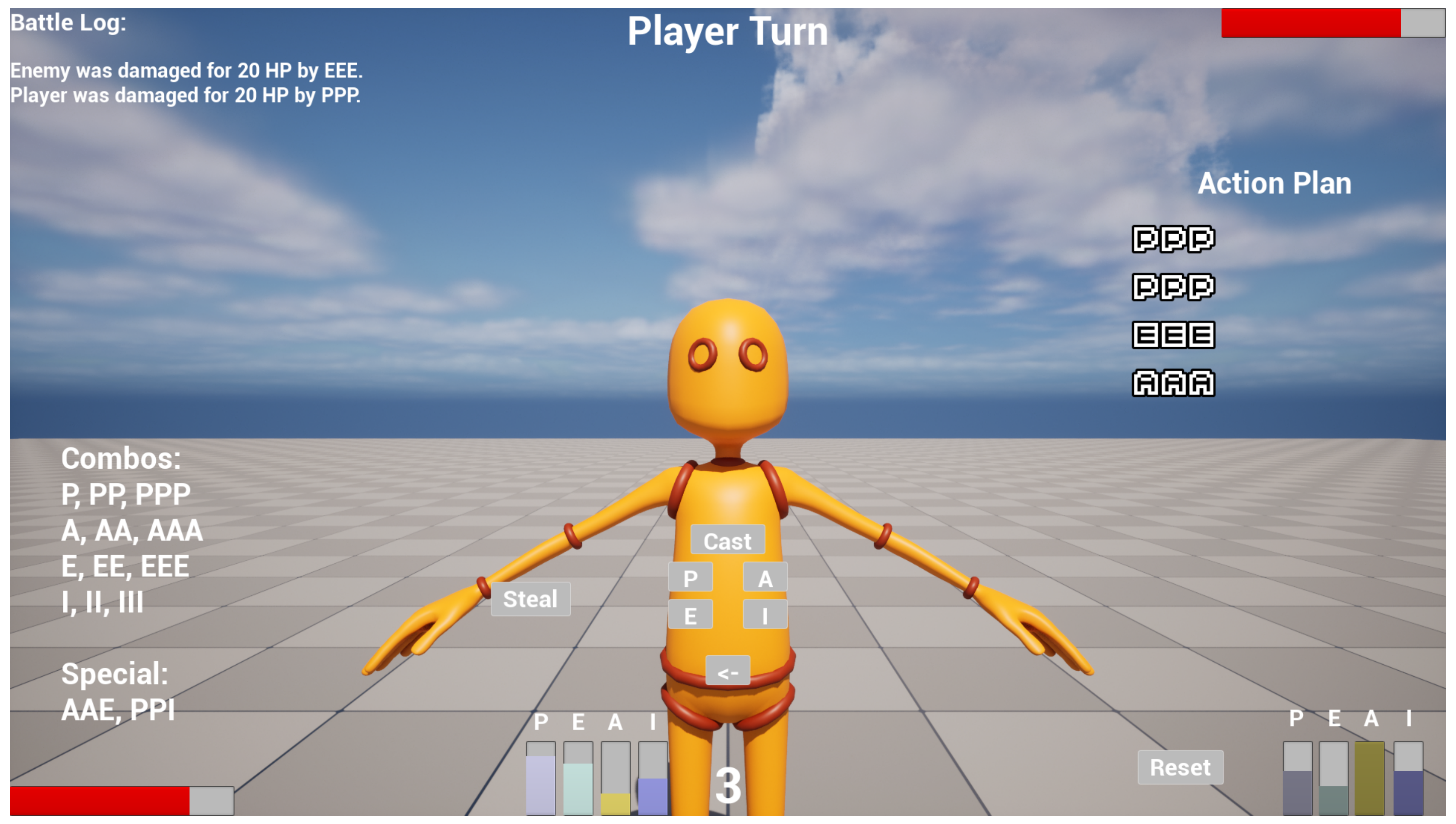
The Problem

To create an entertaining gaming experience using artificial intelligence one must be able to adapt and respond to the ever-changing environment around it. Goal-Oriented Action Planning is one of many solutions to this issue however in this context of turn-based combat Goal-Oriented Action Planning can be highly suitable for this specific use case. This is useful in a turn-based combat system as actions can be weighted based on the damage they apply, the resources they use and in certain actions, the health healed depending on the current state of the environment. Goal-Oriented Action Planning can then use these weights to determine the most optimal set of actions to take in the current situation creating a more entertaining and dynamic experience for the player.

Github Repository

Implementation

The implementation of the Goal-Oriented Action Planning algorithm is based on the implementation by *Whyte* (Whyte 2019) in the game engine Unity with some modifications, additions and optimisations to suit the needs of the game engine Unreal, in which the artefact was developed in, and constraints when building the tree of possible actions to reach the agent's goal. The Goal-Oriented Action Planning algorithm and agent are implemented as an actor component that can be attached to any actor, pawn or character game object that requires the ability to make decisions when in turn-based combat, providing developers with the flexibility to apply the artificial intelligence using Goal-Oriented Action Planning anywhere as needed. The Goal-Oriented Action Planning algorithm also implements self-recovery re-planning if the first action plan will fail to reach the goal if the environment state does not match the predicted environment state the action plan has accounted for.


Battle Log:

Enemy was damaged for 20 HP by EEE.
Player was damaged for 20 HP by PPP.

Combos:
P, PP, PPP
A, AA, AAA
E, EE, EEE
I, II, III

Special:
AAE, PPI

Action Plan
PPP
PPP
EEE
AAA

Fig. 2: Screenshot of the artefact.

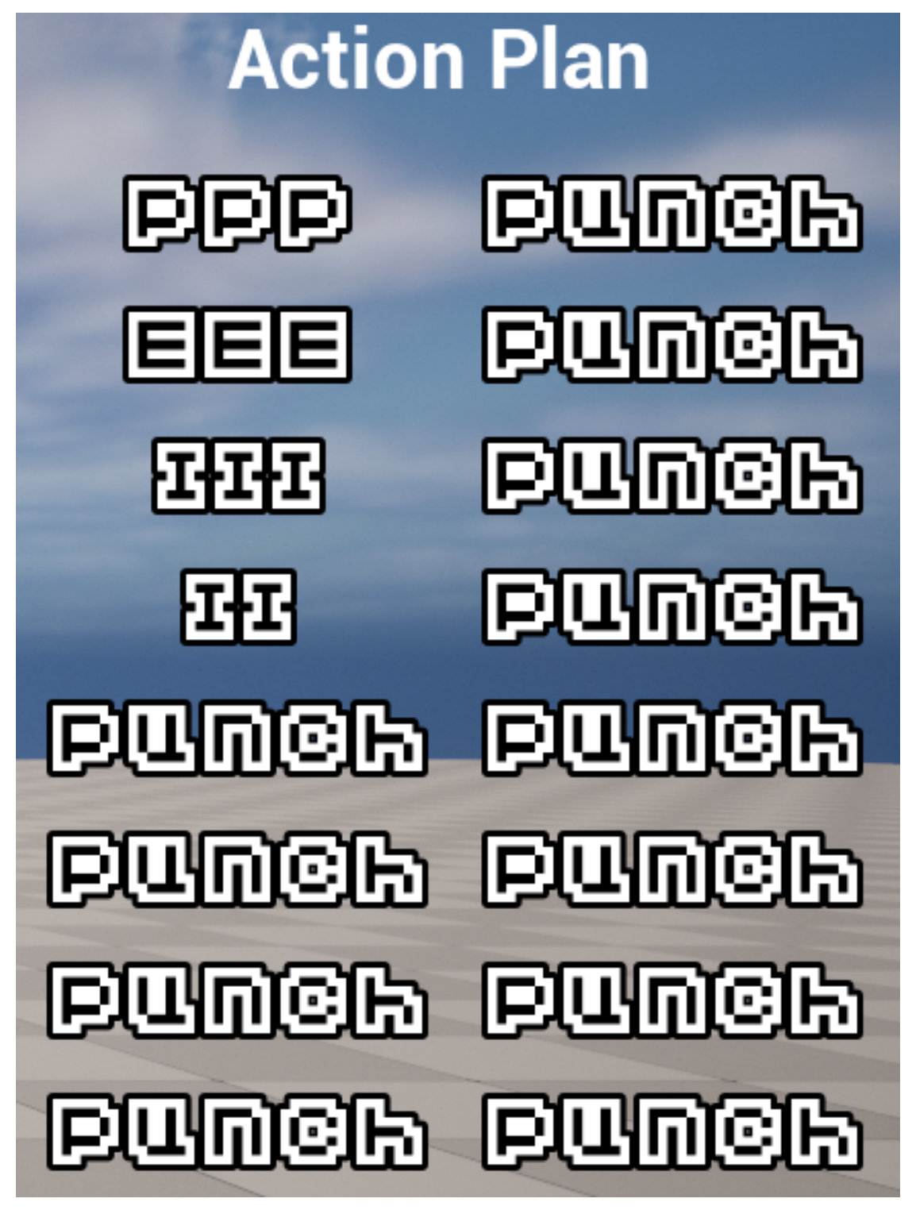
How the action plan is built in this implementation of Goal-Oriented Action Planning is that the algorithm uses a recursive function to build a tree of action nodes that contain an action, the current state of the environment, the parent action node and the total running cost at that point in the tree to generate possible actions to reach the agent's goal. The algorithm creates a list of leaves that contain the final action to an environment state in which the agent has successfully reached its goal. The algorithm then selects the leaf with the lowest cost and then works its way back up the tree to the root node to generate the final action plan. This is then executed by the agent to reach its goal.

One of the critical optimisations of the Goal-Oriented Action Planning that was developed when the algorithm builds the tree is that it sorts the actions based on their weighted costs. This allows the artificial intelligence agents to be able to make complex decisions based on the hierarchy of actions to fulfil the agent's goals. The algorithm considers the current state of the game and the actions available to the agent before deciding on the most efficient course of action. As a result, the agent will always select the action that will get it closer to accomplishing its goals.

Another critical optimisation of the Goal-Oriented Action Planning algorithm is that it uses a hash set to store the states of the environment that have already been visited. This allows the algorithm to avoid visiting the same state multiple times and instead focus on the states that have not been visited yet. This is useful as the algorithm will be able to generate a more efficient action plan as it will not have to consider the same states multiple times.

Results

In this context of turn-based combat, the Goal-Oriented Action Planning algorithm is highly suitable for this specific use case. This is because it allows artificial agents to generate a plan of action and consider the potential outcomes of those actions before performing the actions on their turn. This points to the great potential of using Goal-Oriented Action Planning in turn-based combat games to solve the issue of creating more dynamic and entertaining experiences for the player using artificial intelligence. Additionally, gives developers a great option when deciding what artificial intelligence to use in their games as it is a highly flexible algorithm that can be applied to any game that requires the ability to make decisions in a turn-based combat context.


Action Plan

PPP	punch
EEE	punch
III	punch
II	punch
punch	punch
punch	punch
punch	punch
punch	punch

Fig. 3: Screenshot of the artefacts action plan.

Comparison

Goal-Oriented Action Planning compared to traditional artificial intelligence agents such as finite state machine is that Goal-Oriented Action Planning is a more flexible agent that can be used in a wider range of contexts as it can actively adapt to the uncertainty of the world environmental state in contrast to a finite state machine is a more rigid agent that is only suitable for highly defined use cases with pre-defined actions. In conclusion, Goal-Oriented Action Planning is a highly suitable artificial intelligence agent for turn-based combat games as it can actively adapt to the uncertainty of the world's environmental state and can generate a plan of action to reach the agent's goal.

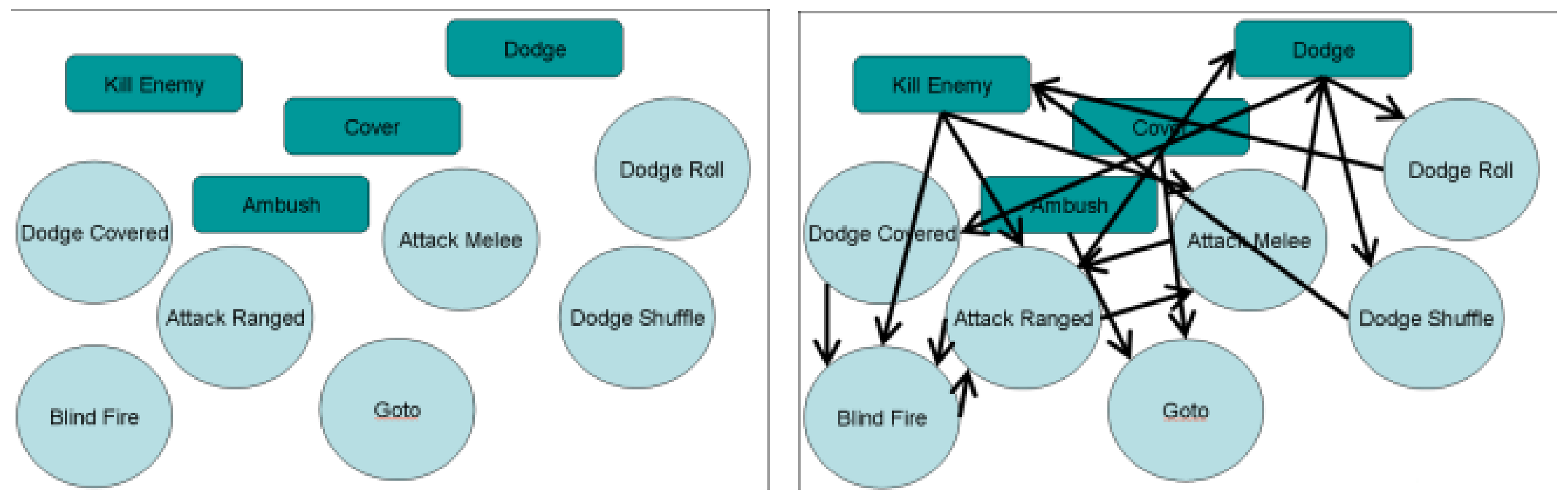


Fig. 4: GOAP vs Finite State Machine. (Orkin 2006)

References

Orkin, Jeff (2006). URL: http://alumni.media.mit.edu/~jorkin//gdc2006_orkin_jeff_fear.pdf.
Whyte, Shaun (2019). URL: https://github.com/shaunwhyte.