

Agent	Problems Solved	Avg. Score per Level
ANU	49	45,112
Hydra	44	47,516
DataLab	26	46,668
Eagle's Wing	16	39,152

Table 1: Levels solved and avg. scores given a 60 sec. time limit.

more associated events which govern their interactions with the environment and other objects in the level. They enable modeling of complex behavior including all collisions, destructions, explosions, and structure collapses. Events also model supporting features such as loading the next bird into the slingshot or exploding the currently active bird. Modeling the motion of blocks after impact and chains of block-block collisions, requires defining a process to track each individual block's change in position and rotation over time, as well as a set of events to account for secondary interactions. Keeping track of dozens of such non-linear processes in every state after a collision would significantly slow down the planner, while the improvement in the domain's accuracy is expected to be small. Therefore, we currently avoid modelling falling blocks processes and block-block collision events.

Processes There are two processes in our PDDL+ model: *increase_angle* and *flying*. The *increase_angle* process is a linear increase of the active bird release angle prior to launching it, as described above. This process is triggered as soon as the next bird in the queue is available for launch. The *flying* process models the flight of the active bird, updating its velocity and location over time, according to the governing non-linear equations of motion. This process is triggered after a bird is released from the sling.

Simplified Modeling While the goal of Angry Birds is to destroy all of the pigs, finding a solution to the corresponding PDDL+ problem was too difficult for our PDDL+ planner in some cases. Therefore, Hydra also generates two simplified PDDL+ problems: (1) a *single-shot problem*, which splits the original scenario into single-bird episodes with the goal of killing at least one pig, and (2) a *single-shot-no-blocks problem*, which only considers platforms and pigs. plans for the *single-shot* problem. If no plan is found in 30s, *single-shot-no-blocks* problem. If again, no plan is found in 30s, we execute a pre-defined default action.

DataLab (?), and Eagle's Wing (?). DataLab won the 2014 & 2015 ScienceBirds competition and Eagle's Wing won the 2016-2018 competitions. Both agents work by trying a set of pre-defined strategies (e.g., destroy pigs, target TNT, target round blocks, destroy structures) and choosing the strategy that will yield the maximal predicted utility. Eagle's Wing also uses a pre-trained XGBoost model to optimize its performance. All agents are designed to work with the Science-Birds API. For the evaluation, each agent attempted to solve a set of 100 randomly generated Angry Birds levels, which can be found at gitlab.com/aibirds/sciencebirdsframework

(for a fair comparison, all agents ran on the same set of levels). We compared the results of all agents to a baseline performance published by ANU, which was computed by averaging the scores of DataLab, Eagle's Wing, and an ANU-created naive agent. The latter targets a randomly selected pig with each active bird, and has predefined windows for apply special actions per bird type. It only considers the locations of pigs, disregarding all other objects in the scene.

Table 1 shows the number of problem solved (out of 100) and the average score per level for each agent and the published baseline results (labelled "ANU"). As can be seen, Hydra's performance is on par with existing domain-dependent approaches, even without encoding the birds' special powers. It solved more problems than DataLab and Eagle's Wing (44 vs. 26 and 16) and obtained the highest average score (47,516) per level. It is worth noting that both Eagle's Wing and DataLab's were designed for the Science-Birds competition, where the levels were hand-crafted with specific winning strategies. In contrast, the levels we used were created by ANU's automated level generator.

Conclusion

We presented an agent that successfully plays the Angry Birds game using a domain-independent planner. Our agent translates Angry Birds levels to PDDL+ problems and solves them using a PDDL+ solver. This demonstrates the expressive power of PDDL+ and corresponding solvers. We plan on further extending our domain to improve its accuracy of PDDL+ planning. In addition, we are exploring techniques to diagnose incorrect PDDL+ models and repair them automatically (?).

Adipisci aut nobis minus deserunt unde quia provident quis atque, aperiam a suscipit numquam aspernatur, tempore sunt aperiam numquam? Quo blanditiis perferendis, adipisci quod debitis deserunt soluta quo quos, fugit ipsa ipsum inventore. Suscipit quasi consequatur cum eius ea est cumque aliquam blanditiis, voluptatem ratione ex quia. Facilis quod nobis neque exercitationem, consequuntur facere quos voluptatum obcaecati sequi, accusantium delectus rem architecto quae nisi eveniet deserunt rerum perferendis culpa, nulla dolore vitae quia tempora magnam, cupiditate qui earum sed fugiat tempora adipisci ratione provident. Modi aliquam aut eveniet ea nam at fugiat, aut exercitationem veritatis fugit voluptate repudiandae beatae doloribus dolorum, repudiandae adipisci neque illum ea nihil tempore aliquid nisi quam laudantium, autem reprehenderit fugit? Vel sunt facere sint illo eligendi porro, in quasi velit reprehenderit, totam quos tenetur nulla impedit? Nam dolorem similique nesciunt reiciendis tempore vel maxime debitis voluptate harum facilis, quaerat sunt velit ipsam rem consequatur consectetur dolorum in, amet at sunt nulla voluptatem vitae architecto dicta earum? Eveniet sed non quos, consectetur velit vero similique quos cumque. Fugiat optio repellat, saepe et repellat nulla a tempore officiis nam labore praesentium minus, et rerum quia dignissimos minus ipsa doloribus recusandae id sed adipisci amet, asperiores atque reprehenderit exercitationem animi temporibus totam assumenda neque adipisci itaque. Temporibus cumque aliquid ad maiores saepe optio, sint tempora iste, delectus con-

sectetur veritatis perferendis repellendus facere accusantium
reprehenderit nobis amet at, architecto veniam eum?