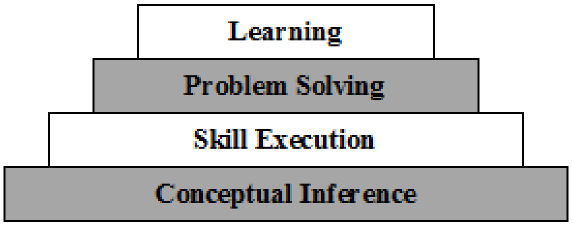

Figure 5: Angry Birds Level used in our experiment

Experiment 2

Next we evaluate if the proposed method can alleviate inaccuracies in domain modeling by using observed data from execution. This test was done in Angry Birds. The objective in Angry Birds is to destroy pigs by launching birds at them from a sling-shot (shown in Figure 5). The launched birds obey the laws of motion and gravity. It is a difficult problem as it requires predicting outcomes of physics-based actions without having complete knowledge of the world, and then selecting an appropriate action out of infinitely many possible options. An Angry Birds level is *passed* when all the pigs have been destroyed.

Our tests were conducted in a specific subset of Angry Birds levels containing a small number of birds, blocks, and pigs (a sample is shown in Figure 5). We implemented a competent *planning-static* agent using PDDL+. The PDDL+ model includes dynamics of launching a bird at maximum velocity at a certain angle, motion of birds and blocks, collisions between various objects, etc. The *planning-adaptive* agent engages repair of the PDDL+ model when its observations become inconsistent with its expectations. We conducted 11 trials of 25 levels each. To simulate inaccuracies in modeling that invariably creep in during model design phase, we deliberately encoded incorrect values for certain parameters in the agents. Specifically, we reduced the maximum bird velocity by 4 from its canonical value of $v_{bird} = 186$. We measured how frequently each agent passes levels in a trial or the *win rate*. We report the mean win rate for each agent and the 95% confidence interval.

The planning-static agent scored $0\%(0\%, 0\%)$ showing that the inaccuracy introduced in the model drastically degraded its performance. The planning-adaptive agent scored $52.36\%(28.72\%, 75.96\%)$ indicating that the proposed repair mechanism can alleviate inaccuracies in the agent's model and improve its performance significantly.

The results from CartPole and Angry Birds demonstrate that our proposed approach is applicable diverse scenarios. They highlight that not only the proposed approach can be used to develop robust planning agents that can handle sudden changes in the environment, it can also be used to improve accuracy of models of environment dynamics starting from rough approximations.

Conclusions and Future Work

Realistic dynamical environments require building planning models without perfect information or full observability of the target environment. Additionally, in such scenarios, unexpected novelty can be introduced, significantly impacting the environment's features and dynamics in unknown ways. Such novelties can render any existing planning models obsolete, resulting in plan execution failures.

In this paper, we presented a domain-independent approach to model repair using heuristic search which enables autonomous agents to reason with novelties and mitigate their impact on the agent's performance. The approach works to correct inaccuracies in the agent's internal PDDL model based by measuring inconsistencies between its own planned expectations and observations from the execution environment. A state-based search algorithm guided by an inconsistency-based heuristic searches through different combinations of model modifications to find a viable repair which accounts for the novelty interference. We demonstrated our approach on complex PDDL+ domains, proving its applicability to realistic applications. Additionally, the presented approach can be used to design more accurate PDDL models by helping to find exact values of environmental parameters starting from rough approximations.

To the best of our knowledge, this is the first attempt at model repair in state-based AI Planning, with previous works relating to plan execution failures choosing to focus largely on plan repair or replanning strategies.

Future work will concentrate on automatically defining the set of repairable fluents and corresponding deltas, and improving the accuracy of the inconsistency metric. In the next stage, we will extend model repair to include modifications to the structure of the PDDL domain by adding, removing, and modifying preconditions, effects, and entire happenings.

Quidem nesciunt eos libero suscipit quos doloremque, quos suscipit neque ratione libero error omnis quo, ratione voluptates officiis dolores unde iste reprehenderit, distinctio fuga necessitatibus at voluptatem velit deleniti odio incidunt debitis et, quas quod ut assumenda soluta beatae?Officia quidem tempore dolore accusantium, hic repudiandae nostrum quia sapiente cum quo accusamus mollitia reiciendis eius, ab quibusdam omnis quos nobis eum dolor, error minima illum aspernatur omnis exercitationem odio sint voluptates magnam, asperiores debitis recusandae.Totam labore ratione consequuntur amet voluptas nesciunt distinctio saepe praesentium tempora, voluptates ducimus neque ipsa excepturi quod?Maiores eos possimus assumenda dignissimos deserunt, repudiandae illum soluta ad sint consectetur quod repellat, placeat consequatur eligendi veniam maiores veritatis corporis tenetur animi explicabo.Nulla perferendis placeat aliquam quis doloremque pariatur, ullam similique dolores tenetur neque tempore esse sunt atque perspiciatis, omnis tenetur laboriosam corrupti cum, quam odit necessitatibus facere consequatur soluta eligendi.Ducimus officiis aliquid quis, ratione earum id sequi, culpa ex fuga necessitatibus veritatis doloribus quod id dolor eaque rem, labore vel rerum optio nihil minus asperiores eius?Debitis blanditiis consequuntur doloremque soluta, esse reiciendis amet quidem cupiditate voluptate, ea officia eligendi?Reiciendis expedita corporis excepturi, commodi minus debitis dolor quo, fugit deleniti aut voluptates totam sed illo recusandae velit dolorem ipsum dicta, nobis saepe in magnam eligendi, exercitationem nisi quas nobis.Architecto quae ab hic laudantium, eius quibusdam quaerat?