PDSim: Simulating Classical Planning Domains with the Unity Game Engine

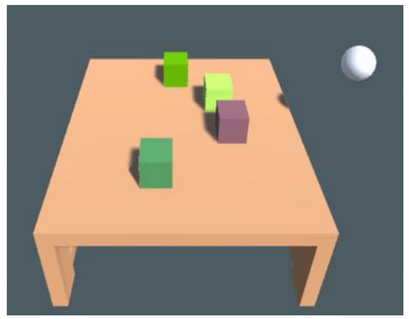
Emanuele De Pellegrin, Ronald P. A. Petrick
{ed50,R.Petrick}@hw.ac.uk
Heriot Watt University - Edinburgh, Scotland, United Kingdom

ABOUT

PDSim is used to simulate a classical plan.
The user is able to create real-world scenes that reflect the execution environment of
the planning problem, exploiting the functionalities of the Unity game engine to be used as an automated planning tool

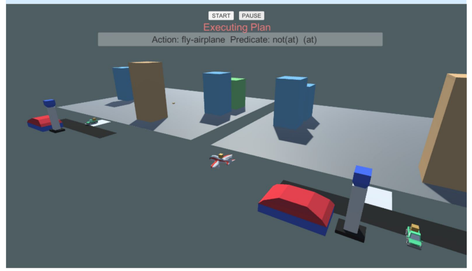
A text plan with the sequence of actions is translated in 3D graphics and animations.

```
(pick-up B)
(stack B A)
(pick-up C)
(stack C B)
(pick-up D)
(stack D C)
```



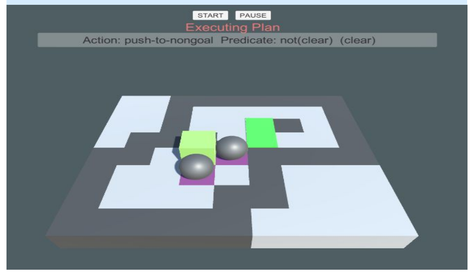
USE CASES

LOGISTIC DOMAIN



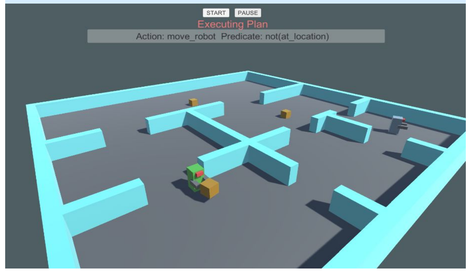
The simulation shows all the components of the logistic domain, such as the airplane the truck and package being moved around.

SOKOBAN DOMAIN



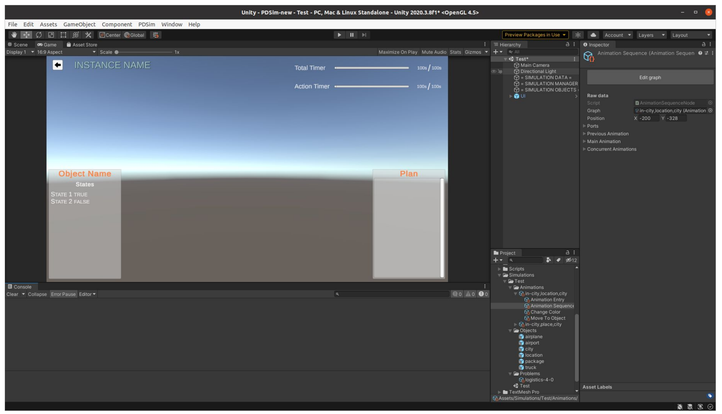
The simulation shows the grid map assembly from the PDDL init block and the player actively pushing the stone to the goal positions highlighted in green.

CUSTOM DOMAINS



This custom domain serves as example on how to set up a simulation that uses the Unity's path planning system, that the robot agents use to navigate in the map.

UNITY GAME ENGINE

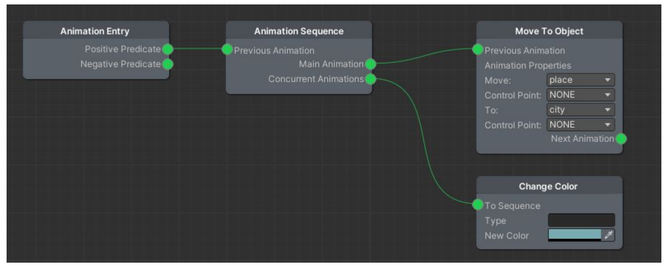


The Unity game Engine editor has been extended to help the user define animations and objects behaviours and appearance used during the plan simulation.
The components such as the physic engine, the path planning system, the audio engine are all available to the user with no need to implement all the components from scratch.

FUTURE WORK

NEW GRAPH-BASED ANIMATION SYSTEM

Animation definition will follow a graph based workflow, where the user can connect nodes that represent executable animations to create animation routines.



TEMPORAL PLANNING SUPPORT

Thanks to a new parser system and a new python backend for PDsim, temporal planning will be supported and the user will be able to see in real time the timed actions being executed in the User interface.

USE YOUR OWN PLANNER

The new python backend also unlocks the ability to use offline planners such as fast downward/forward or POPF.