Homework #4 Report

[redacted]

March 4, 2022

1 wikiHow article

We picked the article How to Survive a Nuclear Attack. This is a highly relevant survival tip that is essential given the recent Russo-Ukraine war and that our society is on the brink of collapse.

For the PDDL task, we translated some selected steps from the 2 parts of the article. For part 1 (Preparing in Advance), we merged the 3 steps, make a plan, stock up on non-perishable food, and stock up on medical supplies into a single problem "get essential items." Similarly, for part 2 (Surviving an Imminent Attack), we translated step 4 and step 5 into two separate problems, "reinforce basement" and "stay in shelter with food."

2 PDDL examples

We made use of types common in interactive fiction games including player, location, direction, and item. We also introduced a new type for our domain, plan, because the player needs to make a plan ahead of time to survive a nuclear attack.

The predicates defined in our domain are as follows:

```
(:predicates
    (is_supermarket ?loc - location) ; this location sells grocery
    (is_nonperishable ?obj - item) ; an item is a non-perishable food
    (is_medicine ?obj - item) ; an item is a medicine
    (is_construction_material ?obj - item) ; an item can be used to reinforce a location
    (is_home ?l1- location) ; this location is player's home
    (is_pharmacy ?l1 - location) ; this location is a pharmacy
    (is_car ?obj - item) ; an item is a vehicle to move through blockade
    (is_underground ?l1 -location) ;
    (sheltered ?p -player) ; the player is in a sheltered place
    (driving ?p - player) ; the player is driving
    (at ?obj - item ?loc - location) ; an item is at a location
    (inventory ?player ?item) ; an item is in the player's inventory
    (has_plan ?p - player); the player has made a plan to survive
    (connected ?loc1 - location ?dir - direction ?loc2 - location) ; location 1 is connected to location 2 in the direction
    (blocked ?loc1 - location ?dir - direction ?loc2 - location) ; the connection between location 1 and 2 in currently blocked
)
```

Some actions we think are unique to our domain are shown below.

```
(:action drive
    :parameters (?dir - direction ?p - player ?l1 - location ?l2 - location)
    :precondition (and (at ?p ?l1) (connected ?l1 ?dir ?l2) (driving ?p) )
    :effect (and (at ?p ?l2) (not (at ?p ?l1)))
)

(:action make_plan
    :parameters (?p - player ?l1 - location)
    :precondition (and (at ?p ?l1) (is_home ?l1))
    :effect (and (has_plan ?p))
)

(:action reinforce
    :parameters (?p - player ?l1 - location ?obj - item)
    :precondition (and (at ?p ?l1) (has_plan ?p) (inventory ?p ?obj) (is_construction_material ?obj))
    :effect (and (reinforced ?l1) )
)

(:action stay_in_shelter
    :parameters (?p - player ?l1 - location ?obj1 - item )
    :precondition (and (at ?p ?l1) (has_plan ?p) (inventory ?p ?obj1) (is_nonperishable ?obj1) (reinforced ?l1) (is_underground ?l1))
    :effect (and (sheltered ?p) )
)
```

We used some custom actions like "drive" and "walk" to specialize a simple "go" action. Because one can only "drive" on a highway, the player needs to satisfy the preconditions of "drive" to unblock the connection from garage to highway. As shown in the preconditions of the "reinforce" action, to "reinforce" a shelter, the player must have made a plan to collect construction materials.

2.1 PDDL Problem

The goal for my problem is to survive the nuclear attack, which involves both stocking up essential items as well as reinforcing the basement for sheltering from the attack. The initial state is the player started at his/her home.

```
plan:
make_plan npc home
walk north npc home garage
get car npc garage
get_in_car npc garage car
drive east npc garage highway
drive down npc highway mall
drive north npc mall wholefoods
get_food canned_food npc wholefoods
drive south npc wholefoods mall
drive east npc mall homedepot
```

```
get concrete npc homedepot
drive west npc homedepot mall
drive up npc mall highway
drive west npc highway garage
drive south npc garage home
drive in npc home basement
reinforce npc basement concrete
stay_in_shelter npc basement canned_food
```

3 Discussions

3.1 PDDL Limitations

Many of the actionable tips mentioned in the wikihow article are very abstract that is hard to be defined using predicates and arguments. There's also a lack of functionality in PDDL to refactor some of the logic. I had to make a new predicate for every nuanced action in the article, which is a bad programming design choice.

3.2 PDDL text adventure

Compared to the previous method of using python OOP to make a text-adventure-style game, PDDL is a significantly less attractive option. While it's possible to create such a game in PDDL, object-oriented programming language is much more suited for such tasks.

3.3 GPT-3 to PDDL

One way to make GPT-3 convert a wikihow article to PDDL is to fine-tune the model with (wikihow titles, PDDL code) pairs, which is the same format as the previous annotation task.

To generate the domain, we can use the wikihow step description, PDDL code pair from the annotations to train GPT-3 to generate problem titles, as well as a list of different types of predicates. We can also use GPT-3 to generate possible action names and their corresponding preconditions and effects described in natural language (see 'natutal language description' annotation). To get the PDDL code that specifies the action logic, we will try to match our generated list of predicates and items (arguments) with generated preconditions/effects descriptions. Finally, we can generate objects/initial states/goals for each problem using the wikihow step description, PDDL code pair.