

# Wikihow Article and Sections

For the PDDL generation, we selected an article on “How to survive an EMP” because it was an interesting article, had very straightforward directions and potentially can have some amount of knowledge transfer for post apocalyptic situations in games.

Since the article doesn't have multiple parts, we converted the following steps from the article to PDDL problem statements

- Turn on the Radio
- Shelter in Place
- Collect Water
- Prepare for an attack

We further defined the following actions :

- Put\_in\_stopper
- Fill
- Purify
- Seal
- Wait
- Go
- Put\_batteries\_in
- Turn\_on\_radio
- Create\_cage
- Close\_cage
- put\_in\_cage

And the following predicates

- Has\_batteries
- On
- Full
- Can\_be\_filled
- Has\_stopped
- Cage\_closed
- In\_cage
- Purified
- Can\_be\_sealed
- Sealed
- At
- Inventory
- Connected
- blocked

We defined the following predicates

- Item
- Player

- Direction
- Location
- Radio
- Faraday cage

## Problems

We created multiple problems

### Turn On Radio

The article recommends turning on battery powered radios to wait for news which we converted to a PDDL problem. Thus the problem's initial state where an npc is in the kitchen, the radio is in the basement and the batteries are in the bedroom. The goal state is that the radio is turned on.

For this problem, the planner came up with the following plan

1. go up npc kitchen bedroom
2. get batteries bedroom npc
3. go down npc bedroom kitchen
4. go down npc kitchen basement
5. put\_batteries\_in npc basement radio
6. turn\_on\_radio npc basement radio

### Shelter In Place

The article recommends sheltering in place, so the initial state is the player/npc at work, the player's children (type item) at school, and the goal state is to have the kids back at home in the basement. For which the planner comes up with

1. wait npc
2. go east npc work backroads
3. go south npc backroads school
4. get kids school npc
5. go west npc school home
6. go down npc home basement
7. drop kids basement npc

### Collect Water

The article recommends collecting water in the event that the city's water supply will fail. The initial state is the player is in the basement, bottles are in the kitchen, there's a bathtub in the bathroom and purification tables in the bathroom. The goal state is to have a full bathtub with purified bottles and the bottles are sealed. The planner comes up with

1. go up npc basement kitchen

2. get bottles kitchen npc
3. go up npc kitchen bedroom
4. go in npc bedroom bathroom
5. get purification\_tablets bathroom npc
6. drop bottles bathroom npc
7. put\_in\_stopper bathtub bathroom npc
8. fill bathtub bathroom npc
9. fill bottles bathroom npc
10. purify bottles bathroom npc
11. seal bottles bathroom npc

## Prepare for Attack

To prepare for an attack, the article recommends creating a faraday cage for your electronics. The initial state has the player in the bedroom, the items needed to make the cage in the kitchen, some items in the basement and the electronics for the cage in a few rooms in the house. The goal state is for the cage to be closed and the electronics to be in the cage. The planner created,

1. get phone bedroom npc
2. get ipad bedroom npc
3. go down npc bedroom kitchen
4. get aluminum\_foil kitchen npc
5. get laptop kitchen npc
6. go down npc kitchen basement
7. get trashcan basement npc
8. get trashlid basement npc
9. get nintendo\_switch basement npc
10. create\_cage npc basement
11. put\_in\_cage npc basement phone
12. put\_in\_cage npc basement ipad
13. put\_in\_cage npc basement laptop
14. put\_in\_cage npc basement nintendo\_switch
15. close\_cage npc basement

## Limitations for PDDL and Scope

A limitation for PDDL that we ran into was a lack of disjunctions or “OR” statements for the predicates and conditionals. Having more complicated predicate conjunctions would allow for richer goal states and perhaps more interesting games. We chose not to implement any problem that would have required disjunctions.

We don't think our PDDL is complicated enough to form an interesting challenge - our current PDDL schemas could make for great fetch quests, but for a compelling challenges, there needs to be some element of opposition, some hurdle to overcome and our PDDL doesn't currently incorporate elements like that. One thinks of the troll in action castle as an example.

## GPT-3

GPT3 can probably be prompted to generate PDDL; however perhaps encoder/decoder models are more suited to the problem. The prompting and generation would have to be done piecemeal like we did in HW2 but without a large amount of PDDLs already created, it would be challenging to generate valid schemas. It should be possible to generate mostly correct schema or schema that requires a small amount of manual corrections. Taking some amount of text, like the content of the steps from wikihow, and using them to generate a list of actions and predicates is already very similar to the kind of prompting we did in HW2, except with larger text context. A challenge might be to have predicates that are consistent and make sense and we believe that would be a major challenge.

GPT3 would likely have problems generating PDDL rules from WikiHow articles because they assume a certain amount of information about the world. For example, the WikiHow article said to "put all electronics in a faraday cage." The model would need to know which objects are electronics and would also need to know that "put in" means to open and close the cage. GPT3 may also violate some basic integrity constraints because of this. For example, a bathtub can only be filled if it is empty. Perhaps augmenting with information from a knowledge base would be beneficial.