1. I picked the article How To Calculate Pi by Throwing Frozen Hotdogs, because it was a relatively interesting/humorous article that mapped particularly well to PDDL, since it described a concrete task, and each step mapped to actions and/or prerequisites, while many other wikihow articles mapped less directly to PDDL tasks (often, wikihow articles described many related tasks, or had missing steps).
2. The article only had one method which had 8 steps, which could be divided into 3 subproblems (get materials, throw hot dogs, and calculate pi/be amazed), so I used all 8 steps of the article. I did not use the tips, Q&A, or warnings, since they did not clearly map to directions, acting more like an appendix.
3. Other than the actions, types, and predicates already given, I added actions such as lay_masking_tape, throw, record_crosses, record_tosses, calculate_pi, and be_amazed (a goal action that requires you to first calculate pi, as described in the wikihow article), among many intermediate actions. I also added types such as items (food items), masking_tape, paper, and pen. Lastly, I added many predicates, some of which include is food_item, frozen, calculated_pi, is_clear, thrown_food_items, and has_throwing_distance, among others.
4. The larger problem for this Wikihow article was to calculate pi by throwing frozen hotdogs (or other suitable food items). I divided this into 3 subproblems: get_materials, throw_hotdogs, and be_amazed. The initial state of get_materials is that you are in the foyer with no items and the goal is to have all the necessary materials for throwing food items to calculate pi, which are unfrozen food items, a pen, and paper. This requires navigating multiple rooms in order to find the items, as they are in both the study and the kitchen. The initial state of throw_hotdogs is that you are in the kitchen and have successfully completed get_materials (i.e. you have the food items, a pen, and paper), and the goal is to throw the hotdogs/food items, which requires you to find a room with enough throwing distance (like the hallway), make sure that your food item has all the necessary properties, clear the room, lay masking tape, and throw the food items. The initial state of be_amazed is that you are in the hallway and have successfully thrown hotdogs (and thus completed all prerequisite steps), and the goal is to be amazed by calculated pi, which requires the tosses and crosses to be recorded, the crosses to be divided by 2, and the tosses to be divided by the crosses to calculate pi, thus amazing oneself.
5. One limitation of PDDL was the extremely large number of possible predicates, which made it extremely tedious to annotate despite not being close to comprehensive even with 24 predicates, many of which were redundant. This was made worse because predicates can only be boolean, which made it necessary to have predicates that were somewhat redundant. Despite this, the PDDL domain was still not close to comprehensive, encompassing only a tiny subset of the vast number of properties an object can have.
6. My PDDL could possibly be used for a text adventure game, although it would not be interesting without some adaptations. Specifically, it would be too easy without adding some clutter or red herrings, as it currently has the bare minimum to make a solution possible, which makes it somewhat linear, although that could be resolved by adding more rooms. The nature of the problem also makes it challenging to come up with a

compelling narrative reason why the player should care about calculating pi by throwing frozen hot dogs, even if it is semi-interesting as a puzzle/challenge.

7. GPT-3 could be fine tuned in order to automatically create a list of objects, predicates, and actions by clustering and summarizing nouns, adjectives and verbs in the article. One challenge could be getting good labeled data, which could possibly be addressed by training on wikihow articles with known PDDL adaptations. Articles could be clustered into subproblems, and the initial state could be outputted. One challenge for long articles is keeping a large volume of text in context subject to hardware constraints without losing coherence, although the domain could be restricted to short wikihow articles.