A. Describe the disaster recovery environment you chose and the two obstacles you have added to the environment.

The disaster recovery environment is built to simulate an area or home that has caught fire. The barriers(columns) in the environment on the outside are meant to simulate areas inaccessible due to flames. The barriers on the inside are intended to simulate other dangerous areas that should be avoided by the robot. The cube inside the simulator is intended to represent an individual trapped in the area; with the robots goal being to find that person and send a console alert. The robot was modified to include 2 additional sensors that can detect columns of flame and stop movement before touching them, keeping the bot safe. There is also an array of sensors that aim to detect people and alert the console if a person is found. There is a second obstacle in the middle of the map that is intended to simulate a building. Because the building is also on fire the bot must navigate the obstacle to find the person.

B. Explain how the robot will improve disaster recovery in the environment from part A after you have added the two obstacles from part A.

The BubbleRob disaster relief robot will enhance recovery efforts in the area affected by the fire in several ways. As described in the previous sections, initial access of emergency personnel to certain disaster recovery locations may be limited by distribution of debris, or other obstacles. The BubbleRob disaster relief robot can traverse these obstacles that would otherwise be too dangerous for humans to traverse. Once deployed, the robot can immediately begin search-and-rescue efforts. Emergency personnel can use BubbleRob's on-board camera to map out the disaster recovery zone and determine the safest strategy for extraction of surviving victims. In addition, while BubbleRob searches for victims of the fire, emergency personnel can work to create a clear and safe path for extraction of victims. Finally, BubbleRob allows emergency personnel to focus their recovery efforts on areas where victims are located, allowing faster access to medical attention for victims. (Anne Devineaux, 2017)

C. Justify the modifications you made to CoppeliaSim's robot architecture, including two sensors you chose to add, and explain how these sensors will aid the disaster recovery effort.

In order best equip BubbleRob for a disaster recovery mode, we need to bolster its existing toolset. I did this by adding two front facing sensors that are similar to the original, but include vision capabilities as well. Next, we need to change the code to lower the penalty to turning that results from activating each sensor. This change results in a faster and more graceful experience overall. From here we have a robot that can safely avoid flames and obstacles, and successfully navigate the environment without getting stuck. In order to allow the robot to easily identify disaster victims in the background, we added a sensor with a greater range and wider angle than the other proximity sensors.

D. Describe how the robot maintains an internal representation of the environment.

The robot can process through the majority of potential environments by just going in a straight line. Knowledge of the environment is created by discovering obstacles and changing their color to mark them. If a collision is detected it can use its gained knowledge of the surrounding area and adjust its movement accordingly. This demonstrates the ability to collect information about the environment and store it internally. This internally stored information also allows the robot to operate with a degree of uncertainty, and continue working even when there isn't a complete picture of the environment.

E. Explain how the robot implements the following four concepts to achieve its goal:

- **reasoning**
- **knowledge representation**
- **uncertainty**
- **intelligence**

Several methods of optimization were employed when designing the BubbleRob disaster relief robot. The most obvious example of optimization would be the modification of the movement logic to decrease the length of time that the robot spends in reverse when one of its object detection sensors is triggered. This allows the robot to both respond to the encountered object more quickly as well as prevents the robot from entering an infinite loop of turns (which would lead to the robot traveling in a circle).

1. REASONING

The BubbleRob disaster relief robot exhibits the capability to perform basic reasoning through its ability to navigate around and through obstacles. The Lua code that governs the robot's movement capabilities allows the robot to reason about what is considered an impassable obstacle versus navigable terrain. The code also provides the robot with instructions for how to reason through encounters with various obstacles (i.e. whether the robot should shift into reverse or turn to the right or left).

2. KNOWLEDGE REPRESENTATION

The BubbleRob disaster relief robot has several methods for representing knowledge. First, its object detection sensors allow it to organize and categorize terrain into two categories – navigable or non-navigable. In addition, the robot can justify ceasing all movement activities when it locates a disaster survivor. Finally, it can represent the fact that it has found a victim by engaging its victim indicator spotlight and can transfer this information to the emergency personnel responding to the disaster.

3. UNCERTAINTY

The BubbleRob robot operates with a constant level of uncertainty about its environment. It does not have any prior knowledge of the environment that it is being deployed into, or whether disaster victims may be present in said environment. Instead, it is equipped with the sensors to handle this level of uncertainty and respond to events in an appropriate manner.

4. INTELLIGENCE

The BubbleRob robot can be classified as a simple reflex agent – it selects its current action based on the current percept of its environment. Because of this, the intelligence of the BubbleRob robot is primarily driven by the condition-action rules embedded within its custom Lua code. The triggering of a condition will result in the BubbleRob robot taking a predetermined action. For example, triggering the left object detection sensor without triggering the center object detection sensor will result in the robot turning slightly right for 1 unit of simulation time.

F. Explain how the prototype could be further improved, including how reinforced learning and advanced search algorithms can improve the prototype's performance and learning.

Due to the fact that the robot lacks rear object detection sensors, it is possible for the robot to shift into reverse and back itself into debris, causing it to become lodged in the debris, rendering it unusable. This means that the disaster recovery team now must handle search-and-rescue efforts for disaster victims and also recover the prototype as part of their search-and-rescue effort.

A potential improvement that could benefit the movement capabilities of the BubbleRob robot would be a rear object detection sensor. This would remove the need for the robot to shift into reverse for a set amount of time. Instead, when the frontal object detection sensors cause the robot to shift into reverse, we could engineer the robot to either back up for a fixed duration of time or until it detects an object within a certain proximity behind it (whichever happens first). In addition, flattening of the BubbleRob chassis and replacing the wheels with treads would allow the robot to be more flexible in terms of the terrain that it can traverse.

In a real-world scenario, the BubbleRob robot will not have prior knowledge of its environment, and so it will not be able to determine a priori what effect the actions that it takes will have. To avoid disastrous consequences such as the robot missing detection of a disaster victim in need of aid, we could embed a reinforcement learning algorithm into the movement system logic of the robot. First, we will need to enhance the robot with a system that allows it to keep track of the previous coordinates that it has visited within the disaster recovery zone. Then, to allow the robot to "learn" what locations within the disaster recovery zone are more likely to contain a disaster victim, we can incorporate a reinforcement learning algorithm. Our reinforcement learning algorithm will be unsupervised, given that we have no knowledge of the environment ahead of time. We can also consider BubbleRob to be a Q-learning agent – the robot will learn an action-utility function which will provide the robot with the expected utility of taken a given action based on a given state (**Russel, 2021**). In this case, the utility would be finding a disaster victim and the state would be the input from the robot's sensors as well as the history of traversed locations. The robot would be "rewarded" for finding a disaster recovery victim at a given coordinate and penalized for not finding a survivor at a given coordinate. Based on this information, the Q-function would naturally cause the robot to favor locations within a disaster recovery environment that it has not visited before that would be more likely to contain a disaster victim.

G. Submit the robot code that you created.

ATTACHED

H. Provide a Panopto video.

ATTACHED

I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Russel, S. J., & Norvig, P. Artificial intelligence: a modern approach, Chapter 21: Reinforcement Learning, Introduction. Retrieved February 24, 2021, from https://wgu.ucertify.com/?func=ebook&chapter_no=23#top.

Anne Devineaux. Robots to the rescue in post-disaster recovery. (2017, April 10). Retrieved February 24, 2021, from https://www.euronews.com/2017/04/10/robots-to-the-rescue-in-post-disaster-recovery