# Project Summary

*Given the fuel of a rocket and the radius of the orbital assist planet, will the mission from the take off-body, to an orbital assist around the orbital assist body, to landing on the landing body be successful?*

*This will be using a grid to represent space where the rocket takes up one cell and can move one cell at a time. There will be conditions that the rocket needs to meet to move to certain cells, and fuel will be used up when doing so. These conditions are split into three parts: takeoff, assist, and landing.*

*A diagram of a red and green square with arrows

Description automatically generated*

# Propositions

*F – Currently has fuel remaining*

*L – Will move left*

*R – Will move right*

*U – Will move up*

*D – Will move down*

*PTL – Planet Top Left*

*PTR – Planet Top Right*

*PBL – Planet Bottom Left*

*B1, B2, B3, B4, B5, B6, B7, B8 – Binary positions for simple fuel calculation*

# Constraints

*,*

# Model Exploration

*We had a problem that we could not use numbers for calculations, meaning our initial hopes of including velocities and orbital assist calculations as well as using a regular number variable for fuel were lost. However, calculating if there was enough fuel based on a number for the fuel is still extremely necessary, as it is the entire point of the project. So, to solve this problem, we used propositional logic to create a binary number composed of eight propositions, one for each digit. We also designed a function that would decrement the array of propositions representing fuel by one every time it was called.*

# Jape Proof Ideas

*Haven’t properly started this yet, but generally speaking:*

*Given more fuel than cells the rocket must travel this means the rocket will have enough fuel for the mission.*

# Requested Feedback

*Provide 2-3 questions you’d like the TA’s and other students to comment on.*

*How do we loop the code such that it checks all of the constraints for each position along the rocket’s path in the grid, running fuel\_calc for each cell movement, stopping the loop and returning False for finding a solution?*

*What are we allowed to do and use code-wise? Is our code formatted properly, or how can it be converted into the propositional library as seen being used in class?*

# First-Order Extension

Haven’t started this yet.