# Project Description

A ‘whodunnit’ game that features 10 computer-generated players, one villain powered by artificial intelligence, and the protagonist. In a birthday party scenario, the lights periodically go off and a murder is committed. The protagonist must collect clues from other players to determine who is the villain. The villain must commit the murder at the optimal time and place to the most critical player in the protagonist’s future.

# Competitive Analysis

This game features some similar elements to the game ‘Among Us’. However, this game is single-player and the ‘impostor’ and other players are computer-generated. ‘Among Us’, on the other hand, is necessarily multiplayer.

# Structural Plan

The game will be entirely in one file, likely ~1500 lines of code long. The functions are in top-down design. The first section in the file contains the classes necessary, the second section contains the main logic, and the third section contains the user interface.

# Algorithmic Plan

The most complex feature of this is the villain’s AI, accomplished by recursive backtracking. This, assuming perfect logic on the user’s side, hypothetically murders each remaining alive person, and for each, plots the possible paths the user can take (i.e., the number of ways to talk to different people and realize that the only possible murderer is the villain). The murder without that realization or the murder with the longest path to realization is then chosen and executed in the optimal room and with the optimal other people (preferably other suspects of the user). This room and the other people in it is also found through recursive backtracking and seeing which path would make it the most difficult for the user to win. The villain also takes into account the different people in the room after they decide who to kill, and see if them knowing that the villain was in the room when the murder was committed would shorten the path for the user.

# Timeline Plan

By Wednesday (2-Dec): Finish villain backtracking algorithm, complete movement of villain and lying algorithm, add feature to show innocence of person

By Friday (4-Dec): Complete beta version of speaking with other characters (without nltk), add map generation feature

By Monday (7-Dec): Refine user interface; make people stick figures rather than circles, add furniture in each room,

By Wednesday (9-Dec): Complete video, presentation, readme, design docs

# Version Control Plan

I have Google Drive on my laptop and have connected it to my term project folder such that it gets stored on the cloud every time I save the file.

# Module List

I’m using the module ‘names’ to generate random names for the players and ‘uuid’ to generate random unique ids for them.

# TP2 Updates

I refined and almost-fully implemented the villain’s algorithm. I also tried to add a feature to take into account the people NOT in the room and have the people able to give that information, but the recursive backtracking algorithm could simply not take that much of information without lagging the gameplay. I added the chatting feature and added a smiling feature to show the innocence of a person. I did not manage to complete the map generation feature, however.

# TP3 Updates

I worked on the user interface and improved functionality. I included some hand-drawn graphics. I did try to increase the complexity of the backtracking algorithm further, but found that it was taking too long. I added the feature of people in the game sharing information that they collect with their friends, so the murderer tends to kill people with more friends.