Project Description:

The name of my term project is “Five Nights with 15-112”, which is similar to the thriller video game “Five Nights at Freddy’s”. My term project will replicate this video game where the user will be provided with a set of controls such as a ventilation system controller, audio, and CCTV panel. The computer, which will be programmed using methods of backtracking, will try to jump scare the user by reading which control buttons the user presses the most and figure out the most efficient way to sneak its way to me.

Competitive Analysis:

This game is similar to Hello Neighbor, which is a game where the user has to sneak into a neighbor’s house while the neighbor reads the user’s movements and tries to catch the user in action. My project will be similar to this game in that the computer will have to be able to record and analyze my movements using backtracking in order to position itself in the best possible place to catch me (the user). My project will be different in that there will be no 3D graphics. All visual features and UI features will be in 2D, and, in addition, the computer’s exact calculations and movements will be displayed on the game screen somehow in order to clearly show how the algorithmically complex part of my code works.

The 15-112 term project named “Bathroom Floors” utilizes the maze-solving backtracking code, which will also be implemented in my project in order to figure out the best path the computer (monster) can take to catch me. A difference would be that my code would have to find the most efficient way of catching the user among all possible solutions rather than returning a whole list of solutions.

Structural Plan:

There will be a class for user controllable objects and subclasses such as the ventilation control system panel and the CCTV panel. There will be a class for the characters in which the subclasses will be anything related to the user and the computer (monster). There will also be a main menu and a game over menu which should be classes themselves or subclasses of a main Screen class. When user-interactable buttons are pressed, the user should be able to view the directed screens on their display. The main part of the game should incorporate switching between days and displaying “win” when the user successfully manages to survive through 5 nights. As of now, I am considering setting one night equal to 3-5 minutes.

Algorithmic Plan:

The trickiest part of my project is creating the backtracking code that will make the computer (monster) capable of finding the most efficient route to catching me and jump scaring me. A lot of factors go into this such as the computer being able to read the pattern in which I open the control panels and making its best moves during the time I am not keeping the CCTV open. The only way I will be able to efficiently prevent the computer from catching me is if I am able to keep an eye on him on the CCTV cameras. Thus, the computer should be able to efficiently calculate both the shortest distance to catching me and when to enter the ventilation systems to get to me or when to hide from the CCTVs and shut down the power systems. This backtracking code will have to be able to read all of these patterns and redo it to find the most efficient solution, not just any random one because every night is timed, and the computer is trying to stop me from getting through the 5 nights. Therefore, I would probably start by creating a backtracking code that will first be able to display the shortest distance for grading purposes. After this, the next step would be to tackle creating a code that would be able to allow the computer to hide itself from the CCTVs in the most efficient way possible. This would require more backtracking because it will have to track the patterns in when I check the CCTV panel. The final step would be to combine steps 1 and 2 and also add on the feature of the computer being able to shut down the ventilation systems just like it does in the real Five Nights at Freddy’s Game.

Timeline Plan:

Nov 26: Main menu, control panel, monster

Nov 27: Backtracking solution

Nov 30: Perfected graphics for the main menu, control panel, and monster

Dec 2: Switching between nights and adding the win/game over feature

Dec 4/5: Making the game easy to grade by displaying the path the computer takes on the screen.

Version Control Plan:

As you can see in the images below, I am backing up my code using Google Drive. Every time I finish a big part of my code, I will drag the folder on my hard drive into Google Drive and save everything there.

A screenshot of a computer

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Module List:

* Pygame

TP2 Update:

I have completely changed the idea of my game. I felt like the original idea of FNAF would be too hard to use backtracking with. Furthermore, as a grader it would be quite difficult to see what’s going on with the monster if the whole point of the game is to keep the monster hidden as much as possible. Therefore, I decided to switch to a similar game, Hello Neighbor. Some of the changes that I have made are creating code that allows the neighbor to chase me when I am within a certain distance close to him. I have also used sprites to create walking animations of my player and the neighbor. I have also implemented code where the neighbor has 2 states: resting and angry. When the neighbor is resting, he is moving randomly in a certain direction and this randomization in direction alters in a random interval as well. Hence, this gives the effect of the neighbor just wandering around his “house” (I haven’t implemented all the house stuff yet). When the neighbor is angry, which is when I come close to him, he starts to chase me at a speed faster than his normal resting walking speed. I now need to work on creating the interior of the house (maze) and create collectibles where the new objective of my game is to collect the collectibles successfully without getting caught by the neighbor. Also, to make the game harder, I plan to create limited vision for the player so that it’s hard to know where the neighbor is. I also plan to use a pathfinding algorithm in order to make sure the neighbor visits places more frequently within the house, such as, places near the collectibles. **So far, I have completed the code for collision, the neighbor chasing my character, the random motion of the neighbor and the sprite animation for the running motion of both characters. I have also partially completed the code for my collectibles which is named throwables in my code.**

TP3 Update:

In my final game, I have successfully implemented the fog of war which limits the field of view for the user. I have also polished most of the collision detection between the player and the walls. I have also added collectibles which are coins and a final key which can be picked up and used to unlock the door and escape the dungeon. I have also added some sprite animations to the coins and the torches to make it look livelier. I have also added a feature where if the player is in the proximity of the AI monster, an uncomfortable screeching noise would be heard in order to alert the user that the monster is near, sort of like the slender man game.