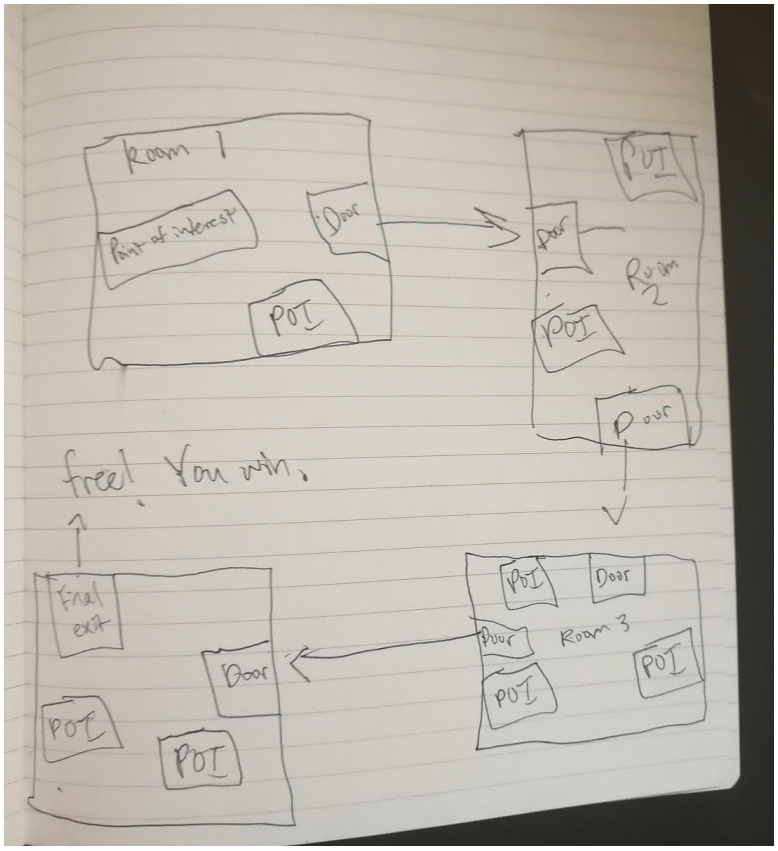
Project Proposal - Simon, Ela, Nick, and Dana

- What is your project?
- We are doing an escape room type adventure game.
- What are the rules of your game, simulation, etc.? Diagrams can be extremely helpful here. Either link to an existing explanation or explain the rules clearly in your own words. **Simply copying and pasting others' work (rules, images, etc.) without citation is plagiarism and unacceptable!**
- The player will start in a room, which will various points of interest that they will have to interact with. At each point of interest, the player will be presented with a decision to make and will have to provide input. Correct answers may result in acquiring items which will help the player escape. Incorrect answers may result in nothing, or could result in death and force the player to restart. The steps within each room eventually allow the player to move onto the next room, until eventually they reach the last room and can escape altogether.
- What will the user's experience be like when they use your program? What will they see on the screen, what buttons or keys will they press, and how will the program react?
- The user will probably be a dot on the screen, and each room will be a square with points of interest represented by smaller squares or triangles or something. The user can navigate through the room with ASDW keys, and when they come across a point of interest, text will pop up presenting a decision to make. Either there will be options to click, or they will have to type in commands. There will be pre-programmed responses for different choices the player makes.
- Include at least one detailed diagram of what the program will look like in action.



- What parts of the program do you think will be most difficult?
- The most difficult thing will be figuring out the visuals for our project. Another important thing is working with the points of interest, figuring out the code which would stop the user from continuing the game until they solve the quest we give for them.
- What do you expect to accomplish during each of the two subsequent versions of the project?
- We hope on visualizing what classes to write and finish the story of the game first. Then we hope to write the tests first, then figure the classes out and probably add the visuals at the end. With the first version of the project, we want the skeleton in place and for it to be usable. For the second version, we'd like to make it more interesting, and maybe do more impressive visuals.
- How will you break your program into classes? You will need at least three: the GUI, the logical model, and a test for the logical model; depending on the project you may need more.
- We would have the class that the program runs from—which would handle user input and drawing—a class for rooms, a class for items, and a class for dealing with what decisions the player has to make and determining if they have made the right choice (as you told us in class, this will involve an array of all decisions and a method which checks them). We would also have classes like vector, lineSegment, and car in the racing game (except car would be the player), to control the movement of the player in the space.

- List at least five early tests you plan to write as you start developing your game. These should be clear and specific enough that you'll be able to express them as JUnit tests. "Moving to a space beyond the edge of the board is illegal" is better than "only legal moves are accepted”.
- First test: The game responds with text when the player encounters (collides with) a point of interest
- Second test: The game responds with the right text when the player makes the correct choice
- Third test : The door to the next room does not open until the player has completed all the points of interest within the room
- Fourth test: The player cannot move beyond the boundaries of the room
- Fifth test: The game knows which items the player has (e.g. if they have acquired two items in a room, we can have a method checkBackpack and the method lists all of the correct items).