

Thad Sauter
CS 162
12/5/17

Final Project: Jail Escape

Story

Background:

You are a prisoner in a maximum-security prison. It is the middle of the night and your jail cell mysteriously opens. You peek out into the hallway and see that everyone else is asleep.

Objective:

To escape, you must use the tools you find scattered throughout the penitentiary.

How to Win:

You have to either find the spoon and dig your way to freedom, or find the warden's mask and escape out the entrance. If you get caught by the guard (end up in the same room as him) you automatically lose.

Initial Design

Idea Behind Project:

I wanted this to be a fun interactive game where you have a guard chasing you around the prison. There are tasks you have to complete to escape the prison, but you also have to worry about the guard catching you at the same time. To me, it makes the game more interesting, as well as more challenging.

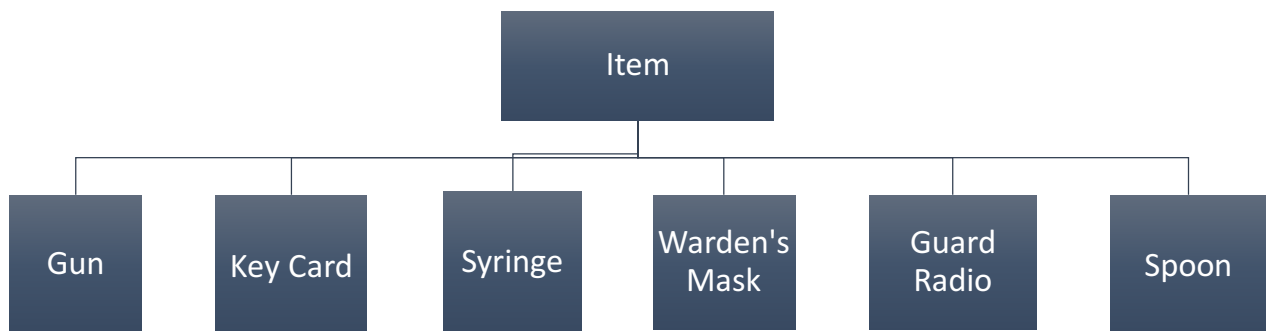
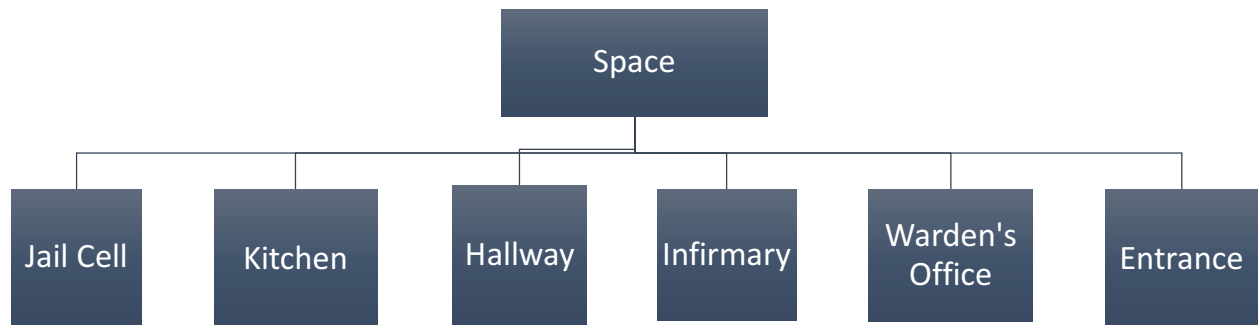
Figuring out the Game Layout:

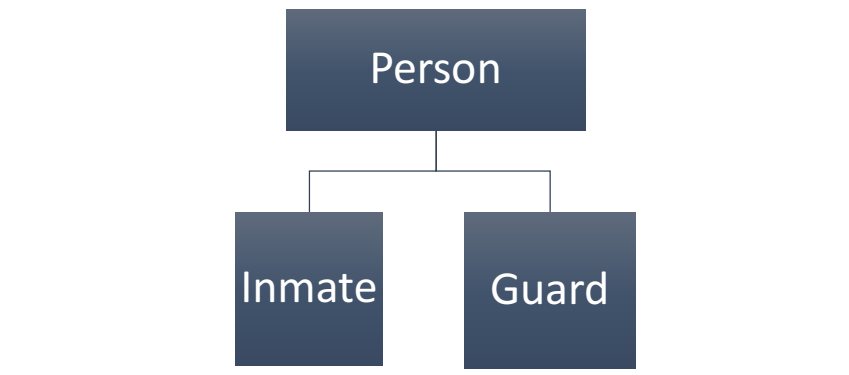
The first step was figuring out how I wanted the game to play. I knew I was going to have make changes anyways, but I had to get an idea down first. Because we had to implement the step limit, I decided to make it a "time" limit and have each action in the game take a certain amount of time. I thought about what each turn would look like and decided that on each action, the player could either move to a different room or inspect the current room. Inspecting the current room took time and gave the player interaction with the room. My thought was that this could be in the form of picking up an item, or using the item to do a task. On each turn that the player takes, I also knew that I wanted the guard to move in a random direction. If the player and the guard ended up in the same room, then the player would lose the game. I also liked the idea of having some spaces being inaccessible to the player if they did not have certain items. This was the basic structure that I laid out for the game and for the most part, I stuck to this plan.

Class Creation:

The first step to creating my classes was drawing out what I wanted my spaces to look like and how they were going to connect. I then brainstormed what types of rooms these could represent in a prison, and made my derived space classes based on that. The other classes that I knew I had to create were an item class (for player to collect and use), a backpack class (to hold items), a and person class. Most of my classes were derived from the Space class, Item class, and Person class.

Class Hierarchy:





Design Changes and Problems

I ran into quite a few design problems that I had to tackle before I could make any more progress on my project. The largest design problem I had was figuring out how to print different menu options for the user based on what room they were in, get user input which corresponded to the room they would like to travel to, and then change the player to that specific room. I had all of the design laid out correctly, as in I knew I wanted to return a string corresponding to the direction choice in the Space class, and then use the direction choice in my game class to change the location of the player. However, I couldn't figure out how to make sure I was getting the right direction choice each time, as the room connections depend on which room the player is in. A TA, Tsewei Peng, was extremely helpful in assisting me with this problem. He suggested that I store the valid choices in an array. This means that a direction would be equal to 0 if it were not an option and either 1, 2, 3, or 4 (based on the direction to travel) if it was an option. This design choice worked perfectly, and allowed me to continue and complete the rest of the project. This was the major design change that I had to make, and the major problem that I had to overcome.

I did have some other debugging problems that I had to solve to get the program to work correctly. I decided to make my own linked list like structure to store items in a backpack. I knew that we needed to limit the number of items that the player could carry, so I decided that I would limit it to 3. However, when I started to write my class, I forgot to decrement the number of items when an item was deleted. Because my `at_capacity` backpack function only checked if the amount of items was equal to 3, I could add more than 3 if I ever deleted an item. This caused a lot of segmentation faults, and took me a while to figure out. Another problem that I had was memory leaks. Originally, I was using the backpack class to delete previously allocated memory for items. However, a lot of the time, I would not pick up every item in the game and thus some items would not ever be deleted after they were constructed. This caused memory leaks in my program, and to fix it I just decided to have the derived Space classes, which dynamically allocated the objects in the first place, delete the objects in their destructors. This way, I would not have to worry about what happens to the objects once they are placed in the backpack. This worked well and got rid of all of my memory leaks

Future Work

Eventually, I would like to add graphics to this project. I was thinking about using a graphics library like SFML or something similar to do this, but am not too familiar with this yet. I would have also loved to add more spaces and make the game much larger. I think sometimes the game is a little bit hard to play because the user doesn't have that many different spaces to go to. This also makes it hard to avoid the guard sometimes. I also would have liked to add an AI feature to the guard so that he could "sense" where the player was or had been, but didn't have time to make this happen. Overall, I'm very happy with the way my project came out, and am looking forward to possibly continuing working on it in the future.