Kara Mansel
CS 162
Final Project Plan and Reflection

Brainstorming: Enter the OSU mansion. You are locked inside. You must search the OSU Mansion's many rooms for the hidden key-the USB drive with your final project. Each room has a piece of furniture. Check all of the rooms and spaces to find the one that holds the key to get out of there! You will collect furniture to keep along the way.

*project hidden in random room at start of game by using rand() w/a function in game class

Classes:

Room-abstract class with pure virtual functions

```
private:

public:

        virtual ~Room()

        virtual checkFurniture()

        virtual moveRoom()
```

Base Room

Private spaces:

Room1-Office-type private furniture-file cabinet-bad grad

Room2- Bathroom-type private furniture-medicine cabinet-magic pill

Room3-Master Bedroom-type bedroom furniture-bed

Room4-Guest Bedroom-type bedroom furniture-nightstand

Room5-Media Room- multipurpose furniture-recliner

Room6-Play Room-type multipurpose furniture-rocking chair

Game (*menu)*

Main

Pseudocode:

```
Game newGame;

newGame.menu()

menu(){

Introduce game, announce goal
}
void play(){
generate random number to place key in room
ask player which room he wants to enter
enter room and check closet and furniture for key
change player status as needed depending on room
player has 15(20?) turns at start of game; decrement for each
add and remove turns as player interacts with parts of room
check for key found. If not, repeat until no turns left
}
lostGame(){}
if (keyFound){
wonGame();
}
```

Initial layout of Rooms:

Hallway->Playroom or MB

MB-> Office->Bathroom

Bathroom->GuestRoom->MasterBedroom

GuestRoom-> upstairs-Playroom->downstairs-MediaRoom

MasterBedroom->upstairs-Office->downstairs-MediaRoom

Playroom->Office->-downstairs-GuestRoom

MediaRoom->-upstairs Bathroom->up and left->GuestRoom->Office

Edited Room Layout: Bathroom is starting place->upstairs is Office->downstairs is Media Room

Issues and Lessons Learned:

I planned this project better than any I'd done before. I spent about a night coding the majority of the project and got stuck on how to implement the 4 pointers. I finally decided to use at least 2 pointers per Room with some rooms having 4 pointers in use and some with pointers to NULL. This was the best decision for me as I was designing a house that had Rooms with paths to certain other rooms. This decision was one of the last I made. Early on, I decided to use a final project as my "key". The key is hidden in a room randomly chosen at the start of the game. The different rooms meet the requirements as they all have a different piece of furniture in them. The player adds the furniture onto their list for the moving van to pick up when they finally leave the OSU Mansion.

The concepts I really felt comfortable with after and even before this project are Polymorphism and Inheritance. I came across a new issue of circular dependency, but with a little research, discovered how to remedy that by using forward declarations, and how to avoid it if at all possible. I did not succeed in ridding my project of leaks. In the initial phases, I had 0 leaks as I was checking after each added function to my Game::play(). However, I began to get tunnel vision near the end and did not check to see if I had leaks throughout coding as I had initially done. This lead me to run out of time to find all of the leaks. Leaks aren't something I am afraid to work on, so it was frustrating that I could not find them in time. Planning the game was actually pretty fun. This class, and this project help me see games in general, and programs all around me very differently. I have a much better appreciation of memory and an even better love of the command line.

Testing:

Room Tests:

Virtual Functions:

moveRoom- testing using Bathroom class, then added add'l classes

test1-failed-was initializing Rooms to NULL in Room constructor-unnecessary

test2-failed-was initializing Rooms to Derived in Bathroom consructor-unecessary

Non- virtual functions:

getRoomName:

worked as expected-returned Room Name for derived class in string form

getFurnitureName:

worked as expected-returned Furniture Name for derived class in string form

Player Tests:

Count Chances:

accurately set and counted number of chances player had left in game

Get/Set Location:

Initial test failed as Room pointer was not properly passed in Game class

Final test passed after code was reworked

Game Tests:

Start Player in Bathroom always:

Move left:

success-player moves to Guest Room and chances increment

Move right:

success-player moves to Master Bedroom and chances increment

Move upstairs:

success-player moves to Office and chances decrement

Move downstairs:

success-player moves to Playroom and chances decrement