Rob Navarro
CS162

# Final Project

**Design:**
My design for this game will consist of the user waking up inside of there house and discovering that a tornado is approaching. In order to survive the user must move around the house and get to the basement. The catch is that the basement is locked and without the key they can't enter. The key has been randomly placed in the house and the user must find it and get to the basement in order to win.

For this game I will need a few classes. The classes that I will use will include the required space class, which will use polymorphism and create several other classes. The other classes that I will use will include a Item class, Player class, and Bag class.

Space Class:

- The space class will consist of several sub classes for the house. These sub classes will include: Bedroom, Office, Entry, Garage, Kitchen, Dining, Living, Basement and Outside. Each class will have 4 pointers that point left, up, right, and down. These will be set so that they point to the room that is next door. Below is a quick illustration of what the game board will look like:

| | | |
|---|---|---|
| Basement | Master Bedroom | Office |
| Kitchen | Living | Entry |
| Dining Room | Small Bedroom | Garage |

- On the perimeter of the house I will have a place holder outside class. Due to the nature of the game the user will not actually be allowed to go outside but they will lose several turns for trying.
- Each time a player moves from room to room they will be presented with a addition or subtract problem. If they solve correctly, they will gain some sort of advantage (this only will occur on the first visit to the room). If they answer incorrectly there will be a penalty that will cause damage to their health or result in the loss of turns until the tornado arrives.

Player Class:

- This class will need variables to track the players health, location, and how many turns until the tornado arrives.
- The class will include several setter and getter methods to adjust the variables as needed.

Item Class:

- This will just be a basic class that includes a name. The item will be identified by it's name that is set when an object is created.

Bag Class:

- This class will be used to track the number of items that the user has found. The limit is 4 but for this version of the game there will only be 4 items placed on the map. If the user finds the key it will be placed in the bag and will allow the user to unlock the basement.

Rob Navarro
CS162

Game/main:

- The game as a whole will be controlled by a do/while loop. The game will continue as long as the player hasn't won, hasn't died, or the tornado hasn't hit.
- This part of the program will control the user's actions throughout the game. The user will be presented with menu options for how the would like to proceed through the map. Only basic input is needed (1-4) and if there is an incorrect choice the user will be asked to re-enter their choice. An additonal do/while loop will control this section
- The user will be presented with their current health and number of turns until the tornado hits after each round.

**Testing:**
The testing for this program will be done as the program is built. I will test the bag class to make sure that I can add items to the bag and search for them properly. The testing will be done with objects created from the item class. This will confirm that not only the bag class is working properly but that the item class is working properly.

After confirming those two objects are working I will build the menu that allows the user to move around the house. With this built I will be able to test if the player can move from room to room. At this point I will also need to create each individual space and set the pointers for each room. Once this is done I will spend time moving around the map to ensure that I can move around successfully.

Once the player can move around the house I will then need to set up the special functions and test them. The special function will be called for each room that the user enters. Each room will have a math problem for the user to solve so I also need to make sure that these problems are being calculated correctly.

I will then test to be sure that it is possible for the player to win, and lose if the tornado hits or the player loses health.

**Testing Results and corrections:**
The testing for the bag and item classes went very well. The classes themselves are fairly simple so there was not much to test, but I did confirm that they were working as expected. This was done by simply adding items to the bag and then getting the names to make sure that they were in the correct spot.

Testing the movement throughout the house was a bit trickier. I decided to store the spaces in a vector so that they remained relatively organized and would be easy to change in the future. Once the vector was filled with spaces I filled each space with the appropriate pointer. After setting the pointers for each room I found that a few pointers were pointing to the wrong spot. Luckily this wasn't a huge issue and was relatively easy to correct.

I also decided to create a vector of 4 item pointers. These items were then placed randomly throughout the house based on a random number generator. I confirmed that this was working while moving through the house and picking up item.

The special functions were actually the hardest part for me to solve and actually ended up in some large changes to the whole program. I found that it was easier to pass the player to each special function and update the player stats in the function itself. Because of this I had to remove the space parameter from the player. I hadn't run into this issue before, but I found that the player and space class could not use each other in the same program. I decided instead to track the current player location in the Game file.

I also found that I needed a way to stop the special function from running again if the player had already entered a room. To solve this, I decided to add a bool parameter for each room type to the player class. Once a player entered that room it would be set to true. The special function would check the bool and if it was true the special function would not run. Instead I would display a message that nothing else was happening in the room and that the player should keep moving.

I also decided to have the special function add and reduce the amount of health or number of turns until the tornado hit each room. If the player did not solve the math problem correctly the would take a varying amount of damage for each room.

The outside class was also changed to inflict both a loss of turns and a loss of health. The special function for the outside class handled these actions. If the user were to go outside too many times the tornado would hit or the player would lose all of their health and die.

Besides these issues testing showed that there were a few errors in the prompts but these were easy to fix. I played through the game several times and took different paths to confirm that it was working as expected.

The final part of the testing was just playing through the game. I asked my girlfriend to play and see if she was able to beat the game. She was able to follow along and figured out how to win after a few turns. She had a few syntax suggestions that were used in the program. I made sure to test that I could lose by losing all of my health or having the tornado hit the house. I also made sure that the player could only enter the basement if they had the key in their bag.