William Kim
CS162
Final Reflection
12/6/15

Final Game Design:

Escape
Hint – Find key to open front door OR use sledgehammer to open up boarded window.

House with 5 rooms
Bedroom 1 (START)
- -linked to hallway
  - -leave room
  - -go to window
    - -attempt to open
      - if (sledgehammer && clownIsDead)
        - escape
      - else if(sledgehammer && !clownIsDead)
        - clown wakes up and kills you
      - else if(!sledgehammer)
        - window is nailed down and boarded up

Bedroom 2
- -linked to hallway
- -sleeping clown
  - -attack
    - -if (sledgehammer)
      - -clownIsDead = true;
    - -else your Dead
  - -leave room
  - -go to window
    - -attempt to open
      - if (sledgehammer && clownIsDead)
        - escape
      - else if(sledgehammer && !clownIsDead)
        - clown wakes up and kills you
      - else if(!sledgehammer)
        - window is nailed down and boarded up

Living Room
- -linked to hallway and kitchen
  - -front door
    - -open(must have key)
    - -if you have sledgehammer, smash open door.
      - -if clown is still alive, DEAD

Kitchen

-linked to hallway and kitchen
-open refrigerator
-inspect severed head
-pick up key
-close
-close

Hallway
-linked to all rooms
-closet
-pick up sledge hammer
-close

Character:
Backpack
-hold 2 items max

First I mapped out the game on paper to figure out how I want the game to interact. I will create a Player, Space, BedroomOne, BedroomTwo, Hallway, Kitchen, and LivingRoom class. The player class has five Boolean variables to determine if the player is alive, the clown is alive, carrying the key, carrying the sledgehammer, and if the player has escaped. I will also have a string array that is used as a backpack container. It will hold the items. There will be an addItem class that takes a item string and then stores the item and also sets the proper Booleans. Also there will be setters and getter for all the Boolean variables.

The Space abstract class will have four space pointers and also a player pointer. There will be a default constructor that will initialize all the variables to null. Then there is a constructor that takes a Player as an argument. There will be a pure virtual option method. A getSpace method will take an integer as a parameter and return a pointer to one of the four spaces. Then there will be an addSpace function that will take four Space pointers and set them.

The subclasses for each individual room will be overloading the pure virtual option function in Space and they will perform as mapped out above. The main class will be the game driver that handles the game play.

Testing:
The player must be able to move through the house correctly.
- Completed the task correctly

If the player attempts to open the window and the clown is alive, the clown wakes up and kills the player.
- Completed the task correctly

If the player attempts to kill the clown without finding the sledgehammer, the clown should wake up and kill the player.
- Completed the task correctly

The player can find and pick up the key.
- Completed the task correctly

The player can find and pick up the sledgehammer.
- Completed the task correctly

The player kills the clown with sledgehammer.

- Completed the task correctly

The player can smash open the window with the sledgehammer while the clown is head.

- Completed the task correctly

The player can open the front door with the key.

- Completed the task correctly

Results from all of the testing were great. I did find some logic errors in my hallway closet and refrigerator code. They worked correctly but if I placed the item back and wanted to open or pick the severed head back up, they would not work until you closed the closet or refrigerator and tried again. Once I fixed the logic errors, everything worked great. I also had to add a time limit that updates and loops through the game until the time limit is reached. If the time has passed, the clown wakes up and kills the player.