Final Project Reflection

Design Description:

Main.cpp:

```
      Void intro()
      Void description()
      Int main()
            Intro()
            Description
            runGame()
            playAgain()
```

Functions.cpp:

```
      Int startLocation()
      Int getDirections()
      Void setDirections()
      Void printBoard()
      Void cubeRun()
      Void runGame()
      Bool playAgain()
```

User.cpp – User Class

```
      Private:
            Int health;
            Int steps;
            Bool game;

      Public:
            User()
            Int getHealth()
            Void setSteps()
            Int getSteps()
            Void reduceHealth()
            Void setWinGame()
            Bool getWinGame()
```

Inventory.cpp – Inventory Class

```
      Private:
            Int size;
            Bool key;
            Bool healthPotion;
      Public:
            Vector<int> backpack;
            Int keyPos;
            Int healthPotPos;

            Inventory()
            Void setSize()
            Int getSize()
            Void addItem()
            Int getItem()
            Void removeItem()
            Bool isFull()
            Void displayBackpack()
            Bool hasKey()
            Bool hasHealthPotion()
            Int getHealthPos()
            Int getKeyPos()


Space.cpp – Space Class (Base Class)

      Protected:
            Int rows;
            Int columns;
            Int number;
            Int space;
            Int top;
            Int bottom;
            Int right;
            Int left;
            Int cube[8][8];
            Bool trap;
            Bool item;
            Bool exit;
            Int itemNumber;

      Public:
            Space()
            ~Space()
            Void setSpace()
            Int getSpace()
            Void setTop()
            Int getTop()
            Void setBottom()
            Int getBottom()
            Void setRight()
            Int getRight()
            Void setLeft()
            Int getRight()
            Void setLeft()
            Int getLeft()
            Virtual bool getItem()
            Virtual bool getExit()
            Virtual bool getTrap()


ExitSpace.cpp – ExitSpace Class – Derived Space Class

      Public:
            ExitSpace()
            Bool getItem()
            Bool getExit()
            Bool getTrap()
```

```
NormalSpace.cpp – NormalSpace Class – Derived Space Class

      Public:
            NormalSpace()
            Bool getItem()
            Bool getExit()
            Bool getTrap()


TrapSpace.cpp – TrapSpace Class – Derived Space Class

      Public:
            TrapSpace()
            Bool getItem()
            Bool getExit()
            Bool getTrap()
```

Reflection:

This program stressed me out a lot but I also had a lot of fun with it. At first I was trying to implement too much into the program and I was getting very frustrated. So I decided to tone it back a bit and just focus on getting the core program working and then to add to it. This ended up helping me out a lot but I still did not get to add all of the stuff to the program that I wanted to as I ran out of time unfortunately. I plan to still add stuff and change stuff in the game to learn more and eventually upload it to GitHub so I can showcase some of my work. I plan on adding more user responses and NPC's to make it more similar to the Cube movie I tried to base the game off of.

I was confused as to why we were supposed to use 4 pointers for each space. I found this to be much harder. I was using a two-dimensional array and it really did not require pointers.

I got stuck in a few parts of the game. First it was hard to implement all the unique scenarios. My runGame function ended up being really long and convoluted and required lots of if/elseif/else statements. I plan on tidying this up and adding more functions so it is not so hard to read. I really did not struggle with pointers or objects at all in this program which shows that I really know how to use them. I still feel like I am doing some things in my program that are unorthodox and I hope to learn better ways to implement certain things in C++ in the future.

I plan on reading some more C++ reference books and working on some of my own programs after this class is over. I had a lot of fun programming in C++. It really stimulated my brain and I hope I can get a job in C++ or a language similar to it, in the future.

Test Case	**Input Values**	**Driver Functions**	**Expected Outcomes**	**Observed Outcomes**
If l->getSpace() == prime ower number on edge of cube	currentLocation	runGame()	Space *l2 = new ExitSpace()	Space *l2 = new ExitSpace()
If l->getSpace() == prime ower number not on edge of cube	currentLocation	runGame()	Space *l2 = new NormalSpace()	Space *l2 = new NormalSpace()
If l->getSpace() == composite number	currentLocation	runGame()	Space *l2 = new TrapSpace()	Space *l2 = new TrapSpace()
IF (*currentLocation < 8 && *currentLocation < 65)	*currentLocation	setDirections()	*up = *currentLocation – 8 Else {*up = 0}	*up = *currentLocation – 8 Else {*up = 0}
IF (*currentLocation > 0 && *currentLocation < 57)	*currentLocation	setDirections()	*down = *currentLocation + 8 Else {*down = 0}	*down = *currentLocation + 8 Else {*down = 0}
If (*currentLocation == not on edge of cube)	*currentLocation	setDirections()	*left = *currentLocation – 1 Else {*left = 0}	*left = *currentLocation – 1 Else {*left = 0}
If (*curentLocation == not on edge of cube)	*currentLocation	setDirections()	*right = *currentLocation + 1 Else {*right = 0}	*right = *currentLocation + 1 Else {*right = 0}