

Final Project Design Documentation and Reflection

Requirements:

1 space abstract class with 1 pure virtual special function.
spaces, that point to 4 other locations
3 derived classes for different types of spaces
the ability of the player to pick up and hold items with a limit
delete a space and add a space at runtime
theme
objective for the player
time limit for player
fun

Design:

Goal:
Retrieve the Golden Statue before Bellock gets to the temple.

Tasks:
clear snakes from the statue room with fire
 find torch in back room
 find ritual knife in sacrifice room
 find flint in jungle
Retrieve Statue
 Get leather pouch from dead explorer in front room
 fill pouch with dirt in the jungle
 deftly recover statue and replace dirt filled pouch
Escape the temple
 enter secret room (that was blocked by rubble)
 use bullwhip to swing across chasm
 escape to jungle

Classes:

Space:
string name
Space* pointer1
Space* pointer2
Space* pointer3
Space* pointer4

Space()
pointer1 = NULL
pointer2 = NULL

```
pointer3 = NULL
pointer4 = NULL
```

```
virtual menu(player* player)
```

```
move()
```

```
int selection
```

```
    cout << "1. Move to " << pointer1->displayName << endl;
    cout << "2. Move to " << pointer2->displayName << endl;
    cout << "3. Move to " << pointer3->displayName << endl;
    cout << "4. Move to " << pointer4->displayName << endl;
    cout << "5. "Don't move" << endl;
    cout << "Please enter the number corresponding to your choice:"
    cin >> selection
    while (!cin || selection < 1 || selection > 5)
    {
        cout << "Invalid selection" << endl
        cout << "Please enter the number corresponding to your choice:"
        cin.clear()
        cin.ignore()
        cin >> selection
    }
```

```
string displayName()
setPointer1
getPointer1
setPointer2
getPointer2
setPointer3
getPointer3
setPointer4
getPointer4
virtual void special = 0
```

Space 1: Jungle:

menu

```
    move
    inventory
    gather flint
        then gather dirt
```

Space 2: Temple Front Room:

menu

```
    move
    inventory
    loot body
```

Space 3: Temple Back Room:

```

menu
    move
    inventory
    remove unlit torch
Space 4: Ritual Sacrifice Room:
    move
    inventory
    take knife
Space 5a: Jade Statue Room:
menu:
    move
    inventory
    clear snakes
        then retrieve statue
Space 5b: Secret Tunnel:
menu:
    swing across

```

```

Player:
enum item [EMPTY, UNLIT TORCH, RITUAL KNIFE, FLINT, FLAMING TORCH, LEATHER
POUCH, DIRT, FILLED POUCH, JADE STATUE]

```

```

Space* location
items array[3] inventory
bool snakes
bool statue
bool win

```

```

Space* getLocation()
setLocation(Space*)
bool getSnakes()
setSnakes(bool)
bool getStatue()
setStatue(bool)
bool getWin()
setWin(bool)

```

```

int main
Player player
create spaces
link spaces
introduction
start/hint/walkthrough options

```

```

while (player->getStatue == FALSE)
    (player->getPosition)->menu

```

when statue:
delete space add space
move to new space
escape

Goal:
Retrieve the Golden Statue before Bellock gets to the temple.

Tasks:
clear snakes from the statue room with fire
find torch in back room
find flint ritual knife in sacrifice room

Retrieve Statue
Get leather pouch from dead explorer in front room
fill pouch with dirt in the jungle
deftly recover statue and replace dirt filled pouch

Escape the temple
enter secret room (that was blocked by rubble)
use bullwhip to swing across chasm
escape to jungle

Testing:

Test	Expected Results	Actual Results	Comments
After creation of all spaces and Player, move between them	Moving correctly with cout prompts	Moving correctly	While I had to change my location from player to main, because I could not get the Space class pointer to work properly in player. Moving worked in main
Test the menu function with simple cout	Menu's properly displayed	Some spacing errors, but the menu calls the cout correctly	Output needed cleaning up but menu was properly called and choices were properly made
Test displaying inventory	Inventory contains: 1. EMPTY 2. EMPTY 3. EMPTY	Inventory contains: 1. EMPTY 2. EMPTY 3. EMPTY	No problems
Test gathering	Simple cout that gather	Gather called	No problems

	was called	successfully	
Test inventory full in gathering	Simple inventory full	Inventory full	No problems
Test replace functionality for gathering	Item is replaced when the inventory is displayed.	Item replaced	No problems
Test combining items that should not combine	“Nothing happens”	“Nothing happens”	No problems
Test combining items	Items are combined and inventory changed accordingly	Failed to combine item	Adjusted code
Test combining items	Items are combined and inventory changed accordingly	Items combined, inventory updated	No problems.
Test jungle menu	can gather dirt can use inventory can move	Gather worked inventory worked move worked	No problems
Test front menu	can use inventory can move can examine explorer and gather leather pouch	inventory worked move worked examine explorer and gather leather pouch worked	No problems
Test back menu	can use inventory can move can gather torch	inventory worked move worked gather torch worked	No problems
Test sacrifice menu	can use inventory can move	inventory worked move worked	No problems
Test statue menu with just snakes	can use inventory can move can scare without torch	inventory worked move worked scare worked	Scare without torch was not formatted correctly, needed to update
Test statue menu with torch.	can use inventory can move can scare with torch	inventory worked move worked scare worked	Scare with torch was not formatted correctly, needed to update cout
Test statue menu after snakes cleared	can use inventory can move can examine statue	inventory worked move worked examine worked	No problems
Test examine statue with filled pouch	can use inventory can move can gather statue	inventory worked move worked gather worked worked	
Test the deletion of statue and the adding of secret	They are deleted and created and the player is placed in ritual sacrifice room	Properly placed in the ritual sacrific room	

Test time limit	Time limit is reached after 40 turns	Time limit never reached	Found that the while statement for repeat and the if statement for game over were off by one. Adjusted code so that time limit is > 39, and it worked properly
Test gathering statue setting off delete	Gathering statue sets off the delete and then player sent to ritual sacrifice room	The deletion happened immediately.	Found that I had not initialized the statue boolean variable to the correct false starting point. Updated and retested and it worked correctly
Test move in main to another room	Move will just reply with "Must move" or Belloq's men shoot	Worked properly	No problems
Test move in main to secret room	Player moves into secret room	Worked properly	Some formatting changes necessary
Test secret room menu	Move "Bad feeling" inventory swing across	Worked properly	Realized I needed to clear the inventory, since the player could see the pouch in inventory, then just eliminated the inventory in secret room menu
Test input validation			Lots of input validation testing, found no errors.

Reflection:

Design decisions:

The space class helped with inheritance of the move function as well as setters and getters for the pointers. I wanted most functionality to be contained in the individual Spaces and the Player class. With that in mind, I only have main call the polymorphic menu function with the Space pointer location variable, and then at the end it calls move. The menu function was different for each space, which then called move, inventory, or executed a special action.

Changes to design:

I had the location in the Player class. However, I was getting many compiler errors trying to call the location function from any of the Space classes. I spent a lot of time trying to debug, but could not figure out the problem, so I switched location to main, and had main keep track of location. Later, I realized I had a circular dependency in the header files for Space and Player, which caused all of the errors.

This changed a lot of my initial design, as no longer was it calling player->location, but instead main gave the location pointer and the pointer to the player, and then menu used those to call all other functions and then to return the location from move, or the original location back to main.

I had to add a few more setters and getters for player class values. Particularly the win condition. I also realized the Mayan use flint knives, so the knife wouldn't be steel, so I just used it as flint and didn't make the player find another item. This did simplify things with only having to combine 2 items.

Improvements:

I would have liked to keep the location in the player class, but after completely shifting gears and design, it was too late to go back and change everything.

Also, in hindsight, per the requirements of the assignment, I should have had main control more of the game, so that the menu would just be contained in main and the calls to special would just be the 3rd option. That would have been less work. It was difficult in terms of balancing encapsulation, with simpler code.

I would have liked to have a bit more interaction within each room, but again time was a constraint. Definitely could have improved some of the formatting of the text.

Thoughts:

Fun assignment, although the requirements tended to force some design decisions I would not have made. Maybe if main had more control over the flow of the game, polymorphism would have been more helpful, but main did a lot less work in my program, than each space and the player class.