

Design

struct Inventory

- have its own header for simplicity
- use an unsigned int and shift bits. Why? Because it's an interesting problem.

class Space

- abstract class, serves as basis for other spaces.
- has 4 ptrs to other Spaces
- virtual void run(struct&) = 0;
- run() in the derived classes will contain the logic for the space.
- bool beenHere so know if running in circles.

int main

- create a text file for the intro flavor text and just read that in instead of putting it in the code.

class River : public Space

- has a small raft at the edge of a river with a can. Player must place two pennies, found elsewhere, into the can for the boat to take them across.

class Fountain : public Space

- has an assortment of pairs of coins the player can place in their inventory.
- removing a pair makes the other pairs vanish, replacing the pair makes the other pairs reappear.
- will require a bool to track if coins are missing - an array for the different types.

class Wheel : public Space

- has two concentric rings that can rotate - the correct combination allows entry to the next room.

class Ruins : public Space

- filler, has flavor text for ancient ruins.

class Start : public Space

- starting point for the game.

class End : public Space

- ending point for the game.

class Cave : public Space

- more filler, flavor text for rocks.

- use one for the start point.

#### Text files

- names after the class, will have the flavor text that is used by the room.

#### Map:

room	p1	p2	p3	p4	
start	wheel	ruins	fount	cave	gets turned into a cave once wheel solved
wheel	start	ruins	ferry	cave	
ruins	start	fount	ferry	cave	
fount	start	ruins	ferry	cave	
cave	cave	wheel	ferry	fount	

#### Testing

A file, test.cpp, was created to test the functions of the various spaces. As the results were dependent upon user input they cannot feasibly be pasted here.

#### Reflections

Text documents for the strings with the flavor text for the game was a good idea - they allow me to change the flavor text without having to re-compile the program. Setting the delimiter to '~', which is not commonly used in the English language allowed me to better format the text as well.