Final Project

Design:

Space class

- Protected data members include Space pointers, integer value, and string value
  - Space pointers include north, south, east, and west
- Virtual destructor
- Getter functions that return Space pointers
- Void setter function that takes a char value and Space pointer through its parameter
- Pure virtual functions were for pick axe, drill, tnt, and menu.

Airport class

- Inherits from the Space class
- Constructor that assigns unique value to reserve and unique string to name
- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

DeSoto class

- Inherits from the Space class
- Private data members for gold and coal
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 20, subtract gold value from

reserve and return gold. If random number is less, return coal.

- Override drill function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 40, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override tnt function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 65, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

Mammoth class

- Inherits from the Space class
- Private data members for gold and coal
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 15, subtract gold value from reserve and return gold. If random number is less, return coal.
- Override drill function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 30, subtract gold value from

reserve and returns gold. If random number is less, return coal.

- Override tnt function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 60, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

Mars class

- Inherits from the Space class
- Private data members for rocks and invaders
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 25, subtract rock value from reserve and return rock. If random number is less, return invaders.
- Override drill function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 50, subtract rock value from reserve and returns rock. If random number is less, return invaders.
- Override tnt function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 75, subtract rock value from

reserve and returns rock. If random number is less, return invaders.

- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

Nigara class

- Inherits from the Space class
- Private data members for gold and coal
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 15, subtract gold value from reserve and return gold. If random number is less, return coal.
- Override drill function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 30, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override tnt function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 60, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

Lost class

- Inherits from the Space class
- Private data members for fish and monster
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function
    - If reserve value is less than 0, print statement. Else, if random number is greater than 25, subtract fish value from reserve and return fish. If random number is less, return monster.
- Override drill function
    - If reserve value is less than 0, print statement. Else, if random number is greater than 50, subtract fish value from reserve and returns fish. If random number is less, return monster.
- Override tnt function
    - If reserve value is less than 0, print statement. Else, if random number is greater than 75, subtract fish value from reserve and returns fish. If random number is less, return monster.
- Override menu function
    - Input validation
    - Returns character value corresponding to input integer

Wind class

- Inherits from the Space class
- Private data members for gold and coal
- Constructor that assigns unique value to reserve, unique string to name, and seeds a random number generator
- Override pick axe function

- If reserve value is less than 0, print statement. Else, if random number is greater than 15, subtract gold value from reserve and return gold. If random number is less, return coal.
- Override drill function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 30, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override tnt function
  - If reserve value is less than 0, print statement. Else, if random number is greater than 60, subtract gold value from reserve and returns gold. If random number is less, return coal.
- Override menu function
  - Input validation
  - Returns character value corresponding to input integer

Miner class

- Private data members include Space pointers, a vector, 2 integer values, and a bool value
- Constructor that assigns a new object to Space pointers, calls setter function, assigns Space pointer airport to location, and assigns values to integer and bool data members
- Destructor deletes new objects
- Play function contains a do-while loop that assigns the return value to a character value and calls member functions corresponding to the charter value. A counter decreases in value every time the function pick axe, drill, and tnt is used. Once the

counter value is 0, the bool value equals true. The loop ends with the charter value equals 'q' or the bool value is true. A for loop is used to accumulate the total value. If the total is greater than 5000, the player wins, else they lose.

Main function

- Creates Miner object
- Calls play function

Problems Occurred:

There weren't any major problems that I encountered when making this program. The one minor issue that I encountered was trying to figure out how I can implement a menu for each space that can connect me to other spaces. This was puzzling at first, but by looking at the 4 sample projects I was able to get a new perspective of how to use the menu functions. Before I was using the void data type for my menu functions and for this project I used the character data type for them. My menu functions now return a char value and I use a setter function that connects the corresponding char value and a specific Space object.

Test Case for tnt() called on Mars or Lost space:

Test Case: Number of times tnt(T) is called	Number of times pickAxe(P) and drill(D) is called	Expected Outcome:	Actual Outcome:
T = 0	P = 10 D = 10	Total > 5000 Player wins	Player wins
T = 1	P = 11 D = 8	Total > 5000 Player wins	Player loses
T = 2	P = 11 D = 7	Total > 5000 Player wins	Player loses
T = 3	P = 12	Total < 5000	Player wins

	D = 5	Player wins	
T = 4	P = 13	Total < 5000	Player wins
	D = 3	Player wins	