CPSC 2720 Spring 2023

Under Us

A Sea Exploration Adventure
(The Hunt for the Lost Sea Cucumber)

[Team Tali]

[Team Members:]

Logan Feit

Keagan Rieder

Maxenne Jubane

Jenil Jani

[April 6]

Team roles

Phase Leader - Keagan Rieder

Secretary - Logan Feit

Quality Assurance - Maxenne Jubane

Mentor - anyone at anytime

Developer - anyone at anytime

Game Overview:

Plot

- Submarine is anchored. You're a crew member who has to collect items to fix the problems and maybe make some sea cucumber friends along the way.
- Second the chef tells you about missing food and equipment, he sends you to look for the source. Gives you keys and a flashlight.
- Looking at the newly unlocked rooms you find the Stowaway who has been causing a ruckus in the submarine.
- During the interrogation on the Stowaway the comms for the submarine are cut. You must find spare parts underwater to fix these grave emergencies.
- The underwater section is unlocked and now explorable. You try to find comms pieces in the dangerous waters. What you do and find there from the randomly generated maps will lead to what ending you get.

Map Layout

- Note this is more of a design layout, and shows the instances of things, not the

infirmary 0 >>	topAirLock 1 < ↓ >	Barracks 2 <<
HallwayEnd 3 >	Elevator 4 < ^ > ↓	Control Room 5 <
Engine 6 -Broken on start >>	AirLock 7 < ^ >↓	Storage 8 <<
Sea 9 (Sea Cucumber Cave/ Atlantis/ Lost City) Random until subs progressed far enough	SeaLock 10 this isn't generated and is always a plains type < ^ >	Sea 11 (ShipWreck) Random until subs progressed far enough

Main Guide:

**Capitalization of Input *doesn't* matter **

Menu Commands:

Play : Goes away from the Menu and Starts playing the game. It will output a quite long amount of words. You'll have to **scroll up** to get the Backstory, then it'll show the recognizes Commands you can type in, then the descriptor of the room you spawn in.

Help : Outputs all the recognized commands, and what it does, when ***playing the game***. Can also be typed during play time.

Exit : Asks if you want to quit out the game, quits if typed 'yes', goes back if types 'no'. Can also be typed during play time.

In-game Commands:

Look : Outputs the description of the room, where you can move to, and if it may have items you can pickup

U, D, L, R : Directional options for moving between rooms,
U = Up, D = Down, L = Left, R = Right.

Talk : Outputs the dialogue the **NPC** in the specific room you are in. Often times they will also give you a **[Quest]** objective you have to do. Getting the item then giving it to them when inputting **FinishQuest** should let you complete it.

Pickup : Let's you Pickup an item in the room. It requests which item you want, and if there are none, or don't want to pickup anything, type *'cancel'* to stop the interaction.

Drop : Let's you drop an item from your inventory into the room. It requests which item you want to drop, and if there are none or don't want to drop anything, type *'cancel'* to stop the interaction.

FinishQuest : Finish the quest given by the person in the room. You must have finished the criteria, for example getting an item, before for it to be successful

FixRoom : Fix current broken room with an item in your inventory.

FillAir : Fill the air in your Airtank. Oxygen will deplete if you don't and you may die as a result of not doing this.

Heal : Heals the player's HP (Health Points). Can only be done if the **Player** has a ***Medkit*** in their inventory and is in the **Infirmary** where the ***Medic NPC*** is located in.

Help : Outputs all the recognized commands, and what it does, when ***playing the game***. Can also be typed during play time.

Exit : Asks if you want to quit out the game, quits if typed 'yes', goes back if types 'no'. Can also be typed during play time.

Troubleshooting Guide:

Environment

- Allows for the deceleration/creation of rooms and locations inside of the game. There are two classes which inherit from this class and allow the formation of more specialized rooms able to do things the environment can't. These are ocean and fixable rooms and map entrances

Variables

+ interactions : Interaction - a list of interactions for the location, t
# presentCharacters : vector<NPC*> - a vector whitch contains the roms/locations current Npcs which are interactable
# contents : Inventory – the contents of a locations ie the items it has
# locationInfo: string a brief summary of a location, will be outputted upon entering it
# fixed : bool = false

Member functions:

+ Enviorment(*inventroy, vector<*NPC>,*Interactions, vector<string>)
+ ~Enviorment()
+ Setinfo(string)
+ AddNPC(*NPC)
+ GetNPC(int) : *npc
+ ListNPC(): string
+ GetInventory(): * Inventory
+ OutPutInfo() : ostream
+ SetFixed(bool)
+ RoomFixed() : bool
+ EnterSea()
+ EnterShip()

FixableRoom

- Allows for a declaration of a room which is broken on game start and requires items to fix. This inherits and defines it's variables and functions form the environment class

Variables

- fixed: bool = false - initially set to false, allows for the room to be checked if fixed, thai will allow for certain interaction inside the game, such as if the engine room is fixed then the sub can move, and new ocean terrain can be generated
- requiredItems: inventory - the required items to fix

Member functions:

+ FixableRoom(*inventory,String[]) - constructor
+ ~FixableRooms()
+ SetFixed(bool) - updates the bool fixed to be true
+ RoomFixed() : bool - returns if the room is fixed

Ocean

- A specialized version of environment witch allows for the spawning of sea creatures like sharks

Variables
- SpawnChance : const double the spawn chance which a sea creature will spawn

Member functions:
+Ocean(double)
+ ~Ocean()
+ GetSpawnChance() : double

Entrance

- Allow for the creation of an airlock room, witch will ensure the player has the proper gear to enter the ocean.

Variables
None

Member functions:
+ ZoneEntrence(string[] Interactson ,*inventory)
+ ~ZoneEntrence()
+ EnterSea() – checks to make sure the player wants to enter the see from the ship well wearing the proper gear/equipment
+ EnterShip()

GameMap

- Holds information that relates to map and interaction with it. this also checks the players cords to see what room they are in/what they can do There are two maps ocean and submarine.

Variables
- subMap : **environment
- seaMap : **ocean - oceans map

Member functions:
+ map()
+ ~map()
+ InitializeSub()
+ GetRoom(int x, int y): *Environment - gets the info of the given room based on the cords passed in
+ GetRoomCommands(int x, int y) : ostream – lists al the rooms available interactions

Coordinate

- Cordnates of the player, allows them to be tracked throughout the game

Variables
- xCord : int
- yCord : int

Member functions:
+ coordinate(int x, int y)
+ SetX(int)

+ SetY(int)
+ GetY() : int
+ GetX() : int

PlayerData

- The games player data, allows tracking of information involving the player

Variables

- score : int = 0
+ cords : Cordnate
- inventory : Inventory

Member functions:

+ AddScore(int)
+ GetInventory() : *Inventory
+ GetScore(): int

Inventory

- Manages inventory management for locations, characters and player

Variables

- contents : vector<item>

Member functions:

+ Inventory(Vector<*item>)
+ ~Inventory()
+ AddItem(*item)
+ GetItem(cell): *item
+ RemoveItem(int cell)
+ ListContents():ostream - outputs the inventory and items position inside of it
+ TransferInventory(*Inventory) - outputs the inventory and items position inside of it

Item

- A base item definition for the game, there is a child call tool which expanded upon it

Variables

- name : string
- descriptions : string
- score : int
- inventorySlot : int

Member functions:

+item(string itemName, string desc, int value)
+ item(*Item)
+ ~Item()
+ SetName(string)
+ SetDescription(string)
+ SetScore(int)
+ SetSlot(): int

+ GetName() : string
+ GetDescription() : String
+ GetScore() : Int
+ GetSlot():int
+ GetInteractions():Ostream

Interaction

- Handles the definitions for the games various interactions, this is the base class which has through children who inherit from it. That being npc, map and item Interactions

MapInteraction

- The maps interaction

Variables
none
Member functions:
+ Move()
+ Search() : ostream
+ Pickup(int Itempos): *Inventory
+ FixRoom(*Inventory): ostream

NpcInteraction

- Npcs Interactions

Variables
none
Member functions:
+ Talk() : String
+ GiveItem(*item)

ItemInteraction

- Items Interactions

Variables
none
Member functions:
+ Use()
+ PickUp()
+ Drop()

UnderUsSetUp

- The games setup, call generates functions and constructors for various classes throughout the game that require being initially defined.

Variables

none
Member functions:
+UnderUsSetup()
+~UnderUsSetup()
+SetUpGame()

UnderUs

- The game's ui/interface. Takes in input and outputs various things to the user. It's really the manger for the game

Variables
- running: bool
- Manger : UnderUsSetUp
Member functions:
+ MainMenu()
+ SetUp() – calls the games various setup functions to initialize objects upon game start when the player inputs the command to start the game inside main menu
+DisplayBackStory()
+ Run() - recursive function that quits when the player types exit
+ Help() -lists helpful information about the game
+ ListCommands() - outputs the commands which are available for the current room, items or npcs, ect/
+ GetInput() : string – gets the input of the player
+ Exit()