

under Us

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



[Team Tali]

[Team Members]

Logan Feit

Keagan Rieder

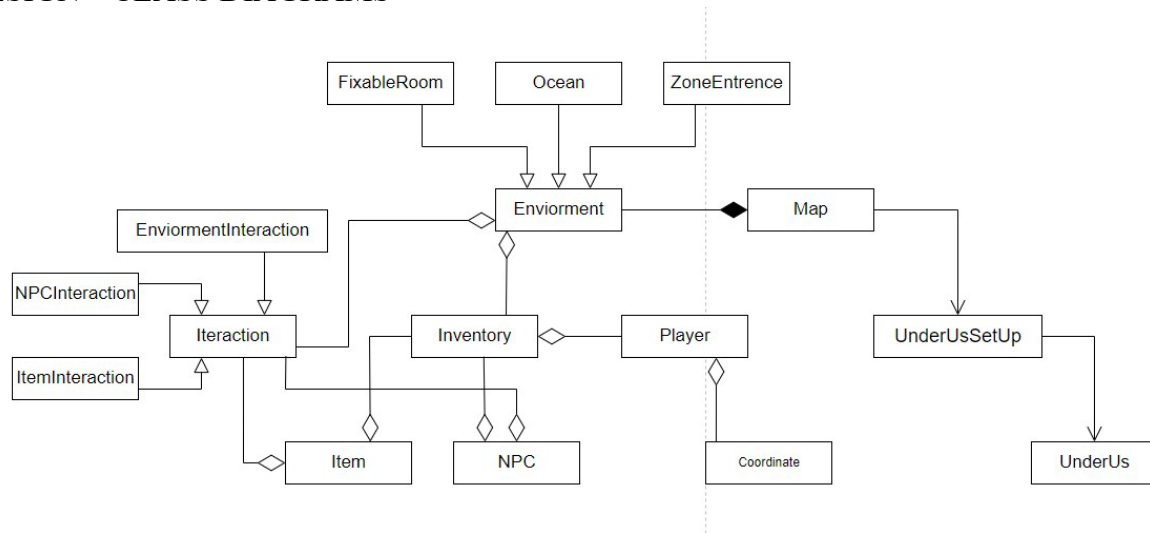
Maxenne Jubane

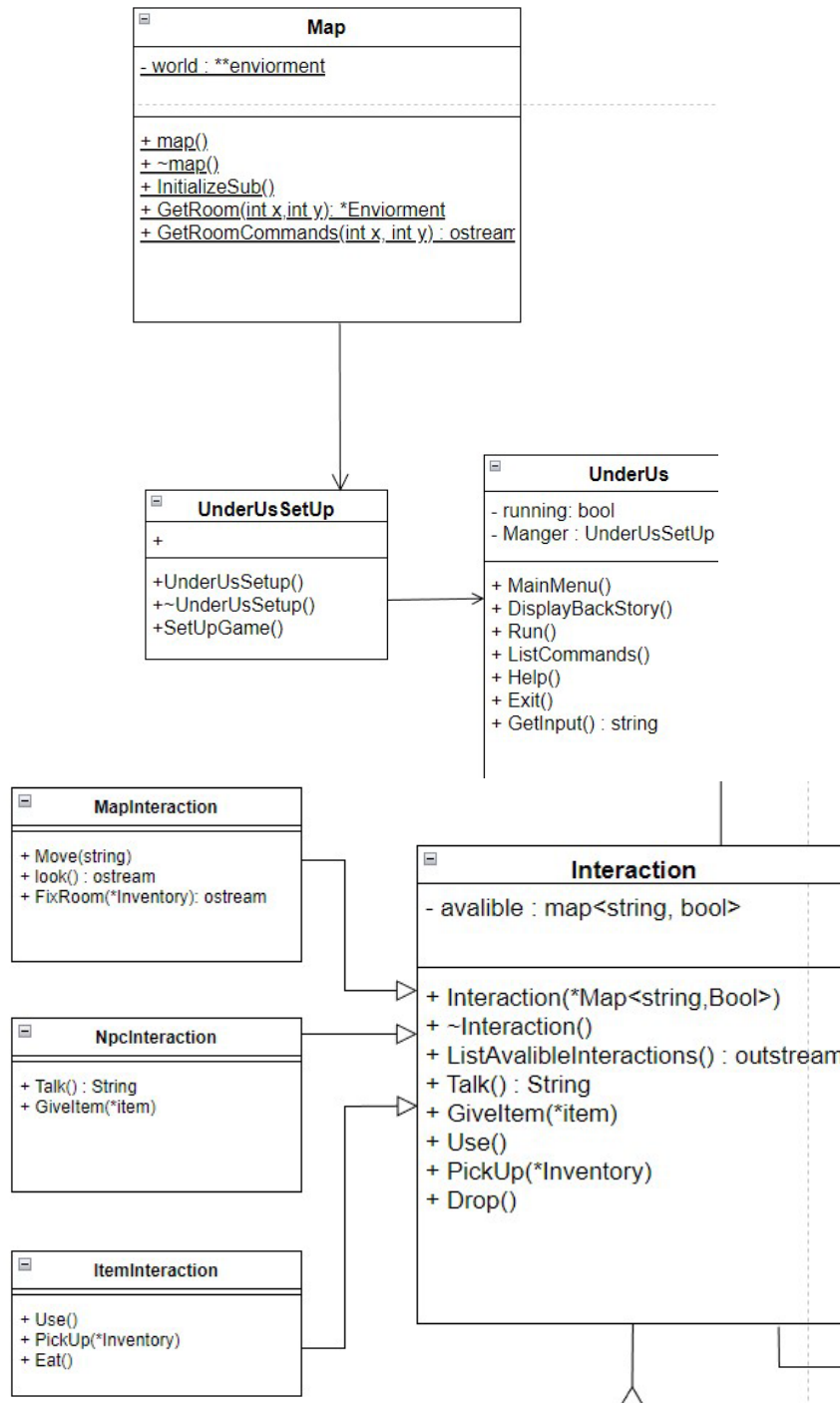
Jenil Jani

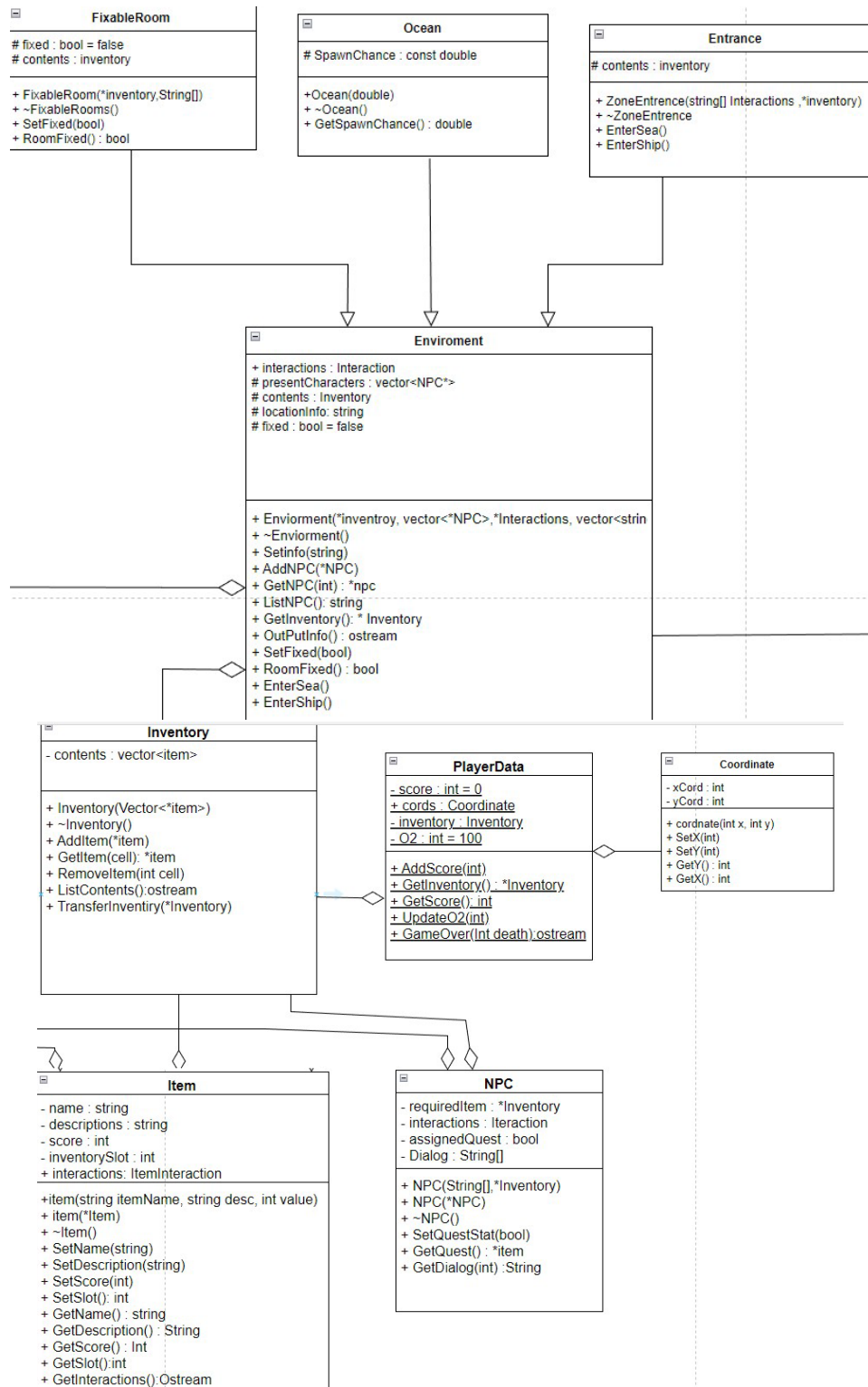
[march 6]

SOFTWARE DESIGN

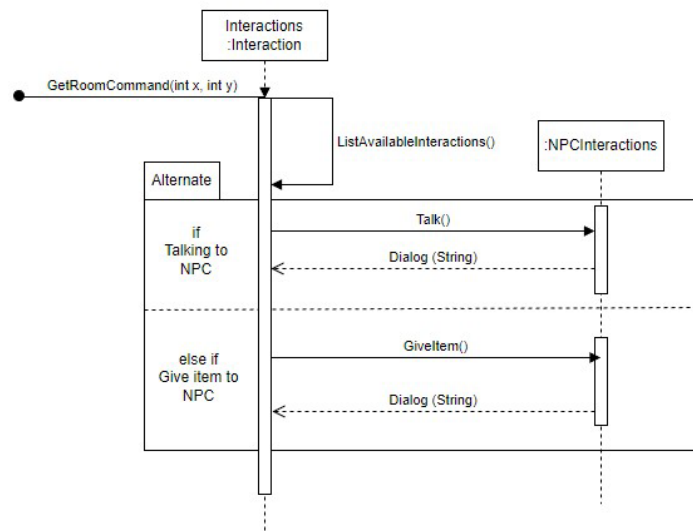
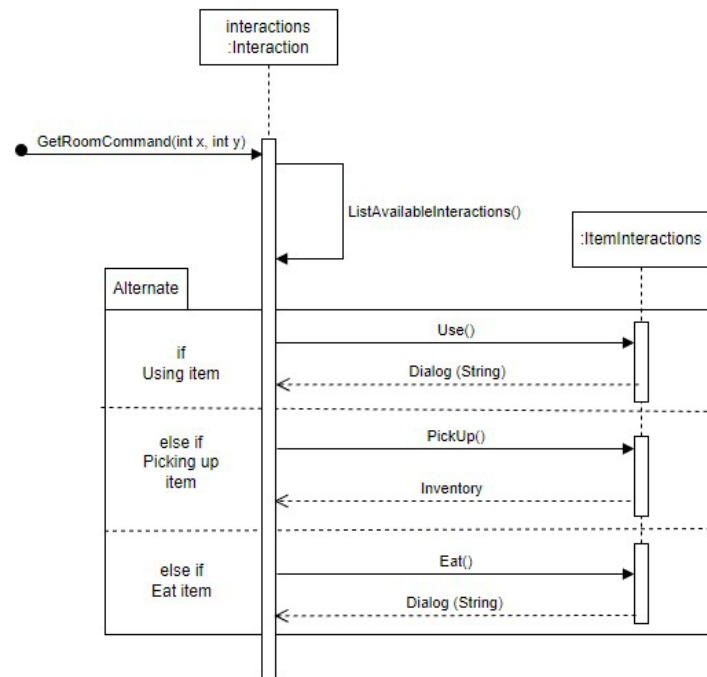
DESIGN – CLASS DIAGRAMS

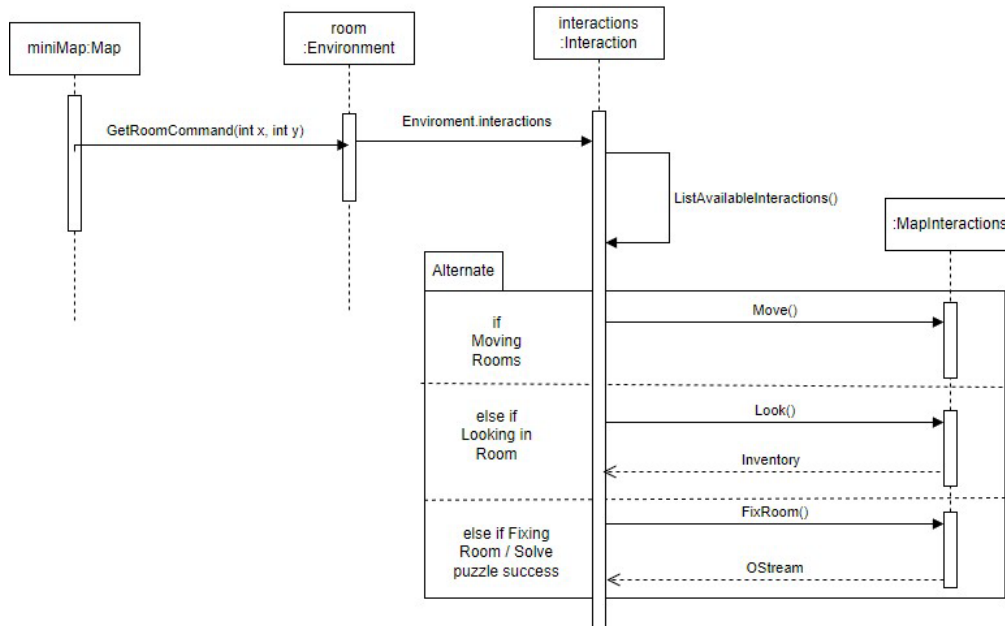






DESIGN – SEQUENCE DIAGRAMS





CLASS DESCRIPTIONS

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 which 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 from the environment class

Variables

- fixed: bool = false - initially set to false, allows for the room to be checked if fixed, that 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 which 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, which will ensure the player has the proper gear to enter the ocean.

Variables

None

Member functions:

- + ZoneEntrance(string[] Interactson ,*inventory)
- + ~ZoneEntrance()
- + EnterSea() – checks to make sure the player wants to enter the sea 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 player's coords 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()

OTHER GAME INFO

PLOT

- Captain's dead, communications gone, engine is not working. You're a crew member who has to collect items to fix the problems and maybe make some sea cucumber friends along the way.
- First the Soldier man sends you to look for engine parts, you get it, fix it, talk to the Soldier again.
- 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 is cut. You must find spare parts underwater to fix this 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 classes

NullPtr (0,0)	infirmary (0,1) >>	Kitchen (0,2) < ↓ >	Crew Quarters (0,3) <<	NullPtr (0,4)
bathroom (1,0) >>	Hallway (1,1) < >	Hallway (1,2) < ^ > ↓	Hallway (1,3) < >	Control Room (1,4) <<
nullptr (2,0)	Engine (2,1) -Broken on start >>	AirLock (2,2) < ^ > ↓	Storage (2,3) <<	NullPtr (2,4)
Sea (3,0) (Sea Cucumber Cave/ Atlantis/ Lost City) Random until subs progressed far enough	Sea (3,1) Randomly generated	Sea (3,2) this isn't generated and is always a plains type < ^ >	Sea (3,3) Randomly generated	Sea (3,4) (ShipWreck) Random until subs progressed far enough