CPSC 2720 [Spring 2023]

Under Us

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

[Tali]

[Team Members:]

Logan Feit

Keagan Rieder

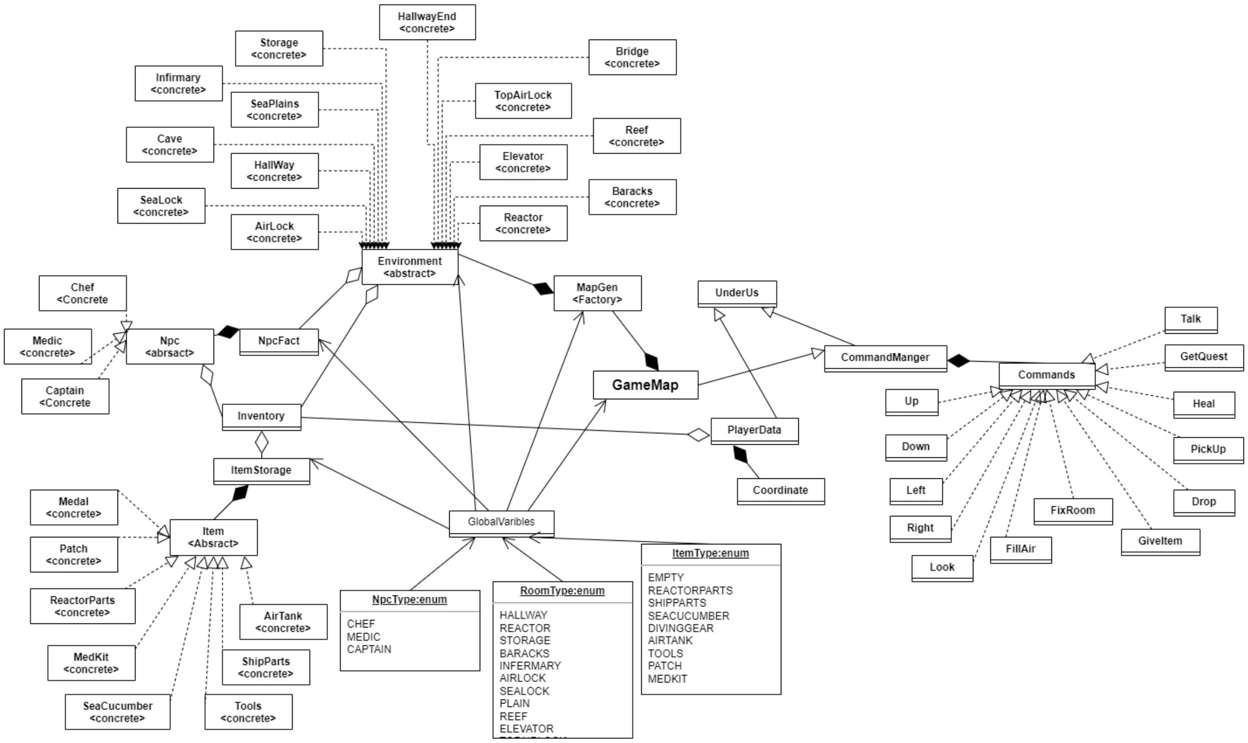
Maxenne Jubane

Jenil Jani

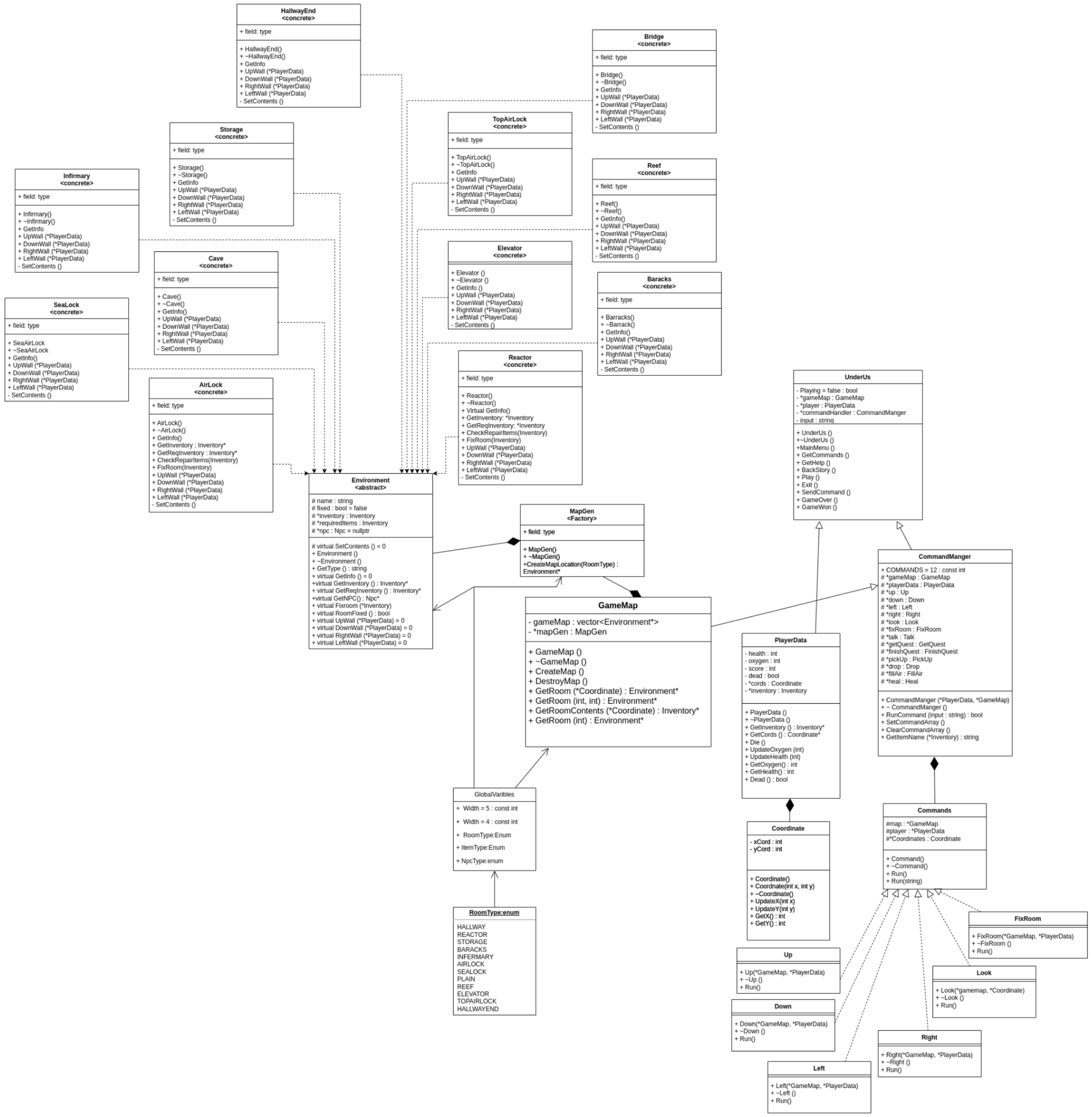
[April 6]

Software Design

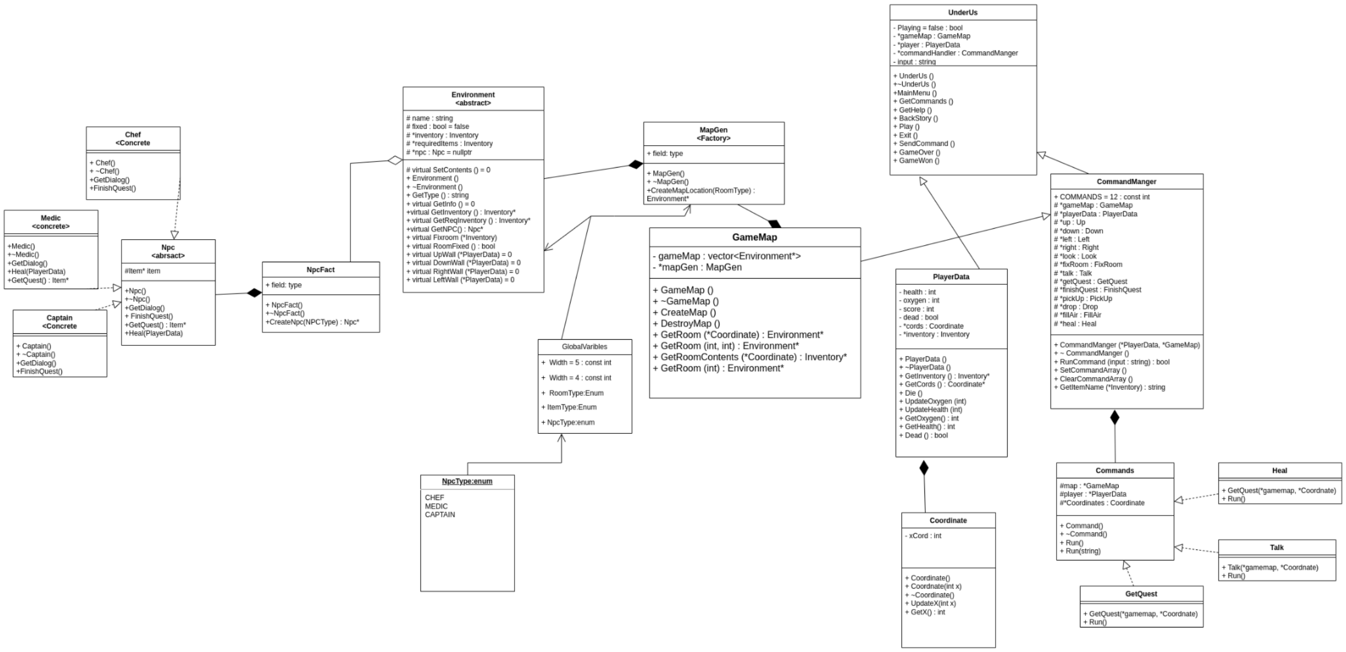
Design UML – Class diagrams



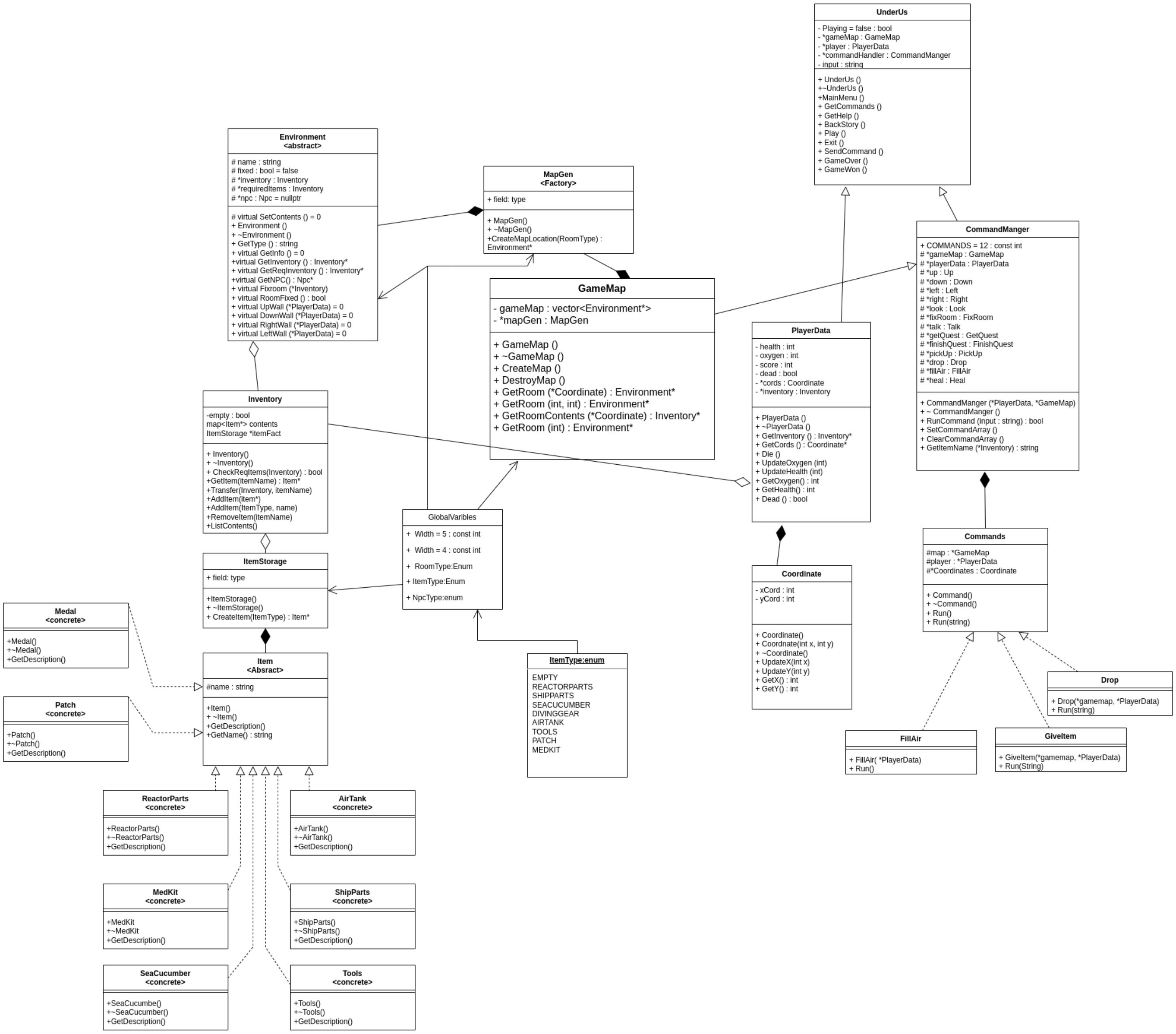
Map Interaction:



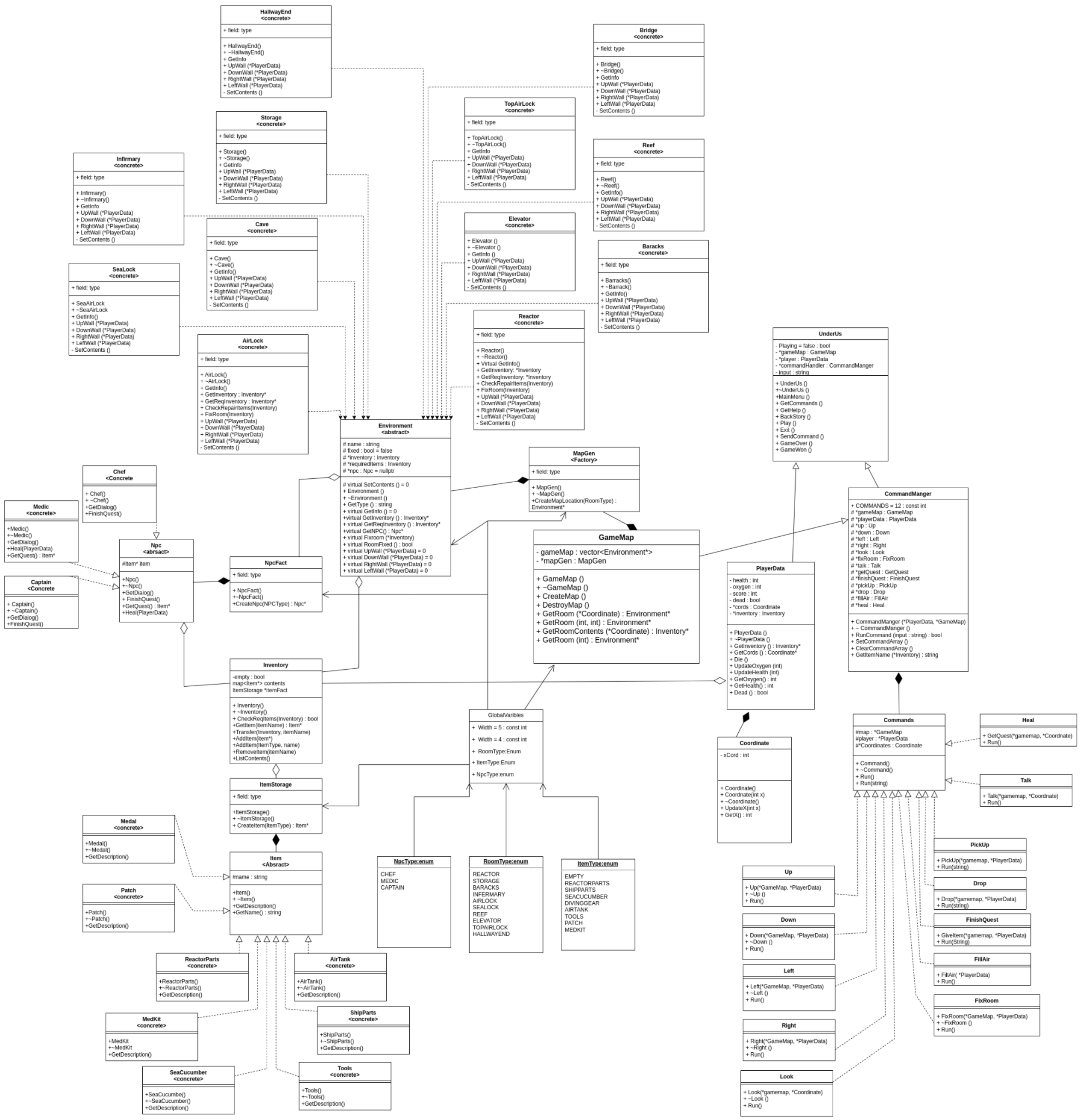
NPC Interaction:



Item Interactions:

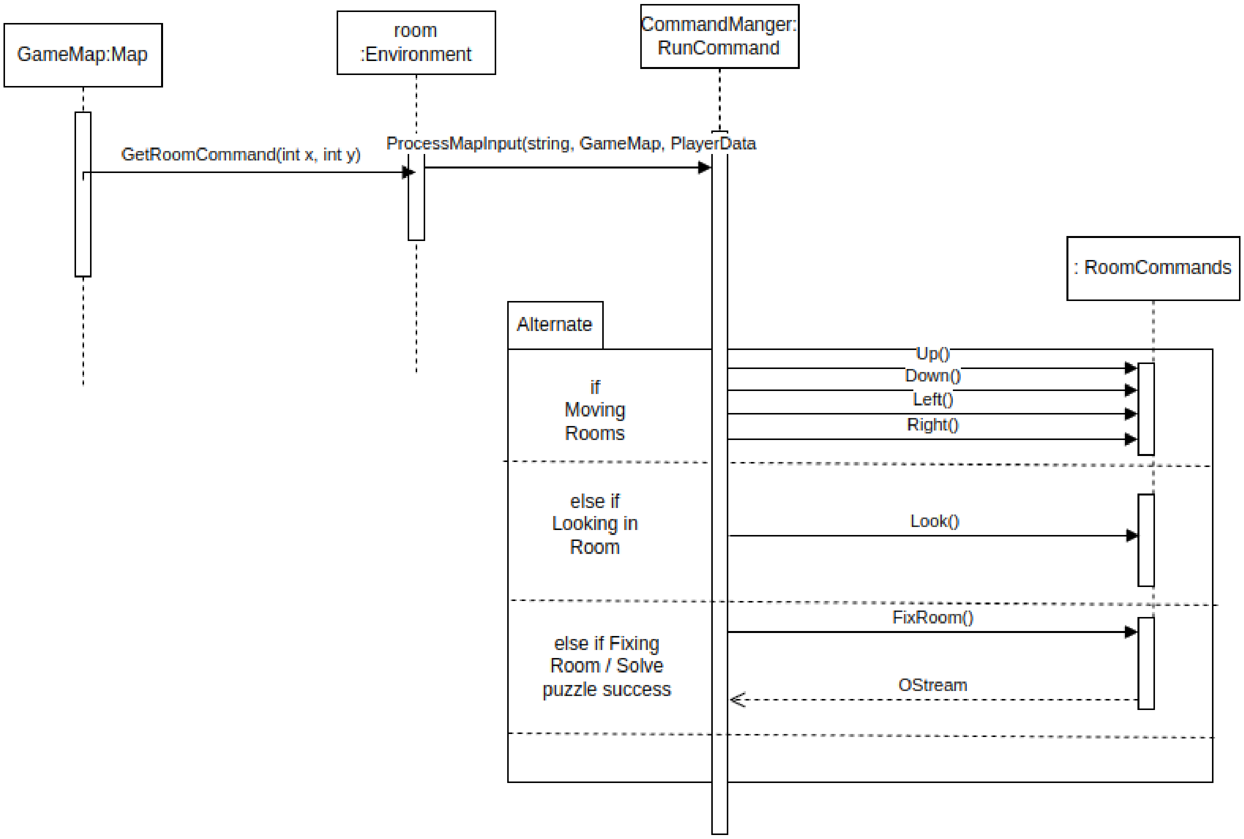


Full UML:

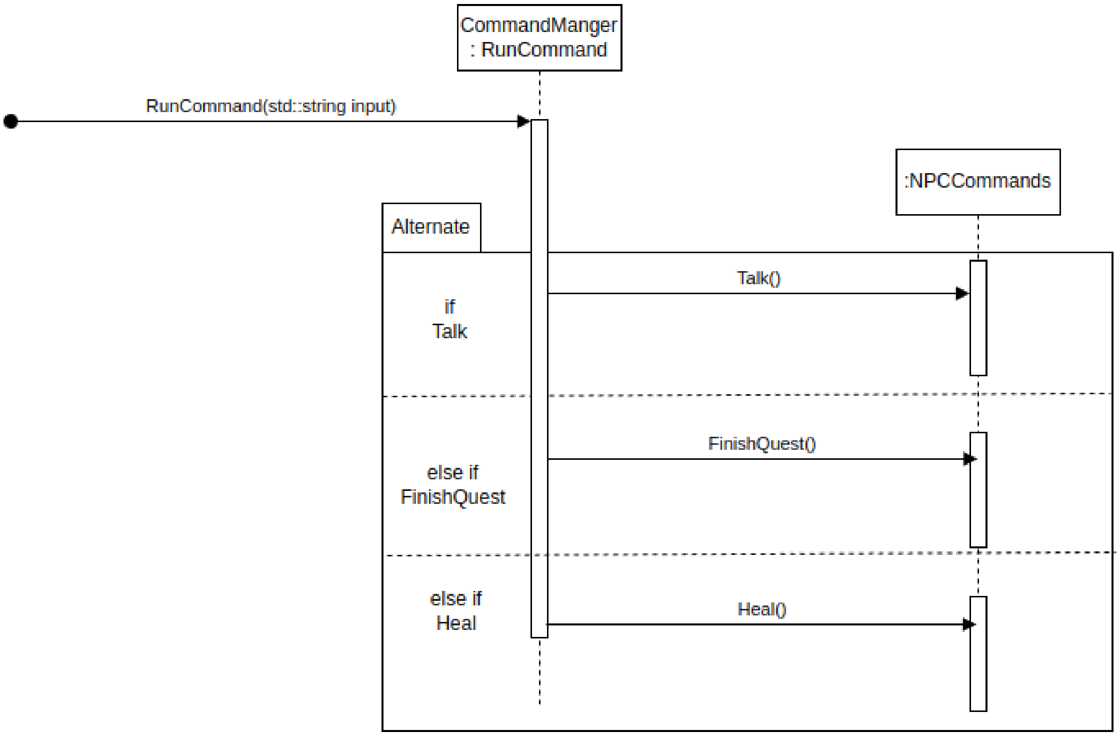

HallwayEnd
<concrete>
+ field: type
+ HallwayEnd()
+ ~HallwayEnd()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Storage
<concrete>
+ field: type
+ Storage()
+ ~Storage()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Infirmary
<concrete>
+ field: type
+ Infirmary()
+ ~Infirmary()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Cave
<concrete>
+ field: type
+ Cave()
+ ~Cave()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
SeaLock
<concrete>
+ field: type
+ SeaAirLock
+ ~SeaAirLock
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
AirLock
<concrete>
+ field: type
+ AirLock()
+ ~AirLock()
+ GetInfo()
+ GetInventory ; Inventory*
+ GetReqInventory : Inventory*
+ CheckRepairItems(Inventory)
+ FixRoom(Inventory)
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Bridge
<concrete>
+ field: type
+ Bridge()
+ ~Bridge()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
TopAirLock
<concrete>
+ field: type
+ TopAirLock()
+ ~TopAirLock()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Reef
<concrete>
+ field: type
+ Reef()
+ ~Reef()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Elevator
<concrete>
+ Elevator ()
+ ~Elevator ()
+ GetInfo ()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Baracks
<concrete>
+ field: type
+ Barracks()
+ ~Barrack()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Reactor
<concrete>
+ field: type
+ Reactor()
+ ~Reactor()
+ Virtual GetInfo()
+ GetInventory: *Inventory
+ GetReqInventory: *Inventory
+ CheckRepairItems(Inventory)
+ FixRoom(Inventory)
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
Environment
<abstract>
# name : string
# fixed : bool = false
# *inventory : Inventory
# *requiredItems : Inventory
# *npc : Npc = nullptr
# virtual SetContents () = 0
+ Environment ()
+ ~Environment ()
+ GetType () : string
+ virtual GetInfo () = 0
+virtual GetInventory () : Inventory*
+ virtual GetReqInventory () : Inventory*
+virtual GetNPC() : Npc*
+ virtual Fixroom (*Inventory)
+ virtual RoomFixed () : bool
+ virtual UpWall (*PlayerData) = 0
+ virtual DownWall (*PlayerData) = 0
+ virtual RightWall (*PlayerData) = 0
+ virtual LeftWall (*PlayerData) = 0
MapGen
<Factory>
+ field: type
+ MapGen()
+ ~MapGen()
+CreateMapLocation(RoomType) : Environment*
GameMap
- gameMap : vector<Environment*>
- *mapGen : MapGen
+ GameMap ()
+ ~GameMap ()
+ CreateMap ()
+ DestroyMap ()
+ GetRoom (*Coordinate) : Environment*
+ GetRoom (int, int) : Environment*
+ GetRoomContents (*Coordinate) : Inventory*
+ GetRoom (int) : Environment*
UnderUs
- Playing = false : bool
- *gameMap : GameMap
- *player : PlayerData
- *commandHandler : CommandManger
- input : string
+ UnderUs ()
+~UnderUs ()
+MainMenu ()
+ GetCommands ()
+ GetHelp ()
+ BackStory ()
+ Play ()
+ Exit ()
+ SendCommand ()
+ GameOver ()
+ GameWon ()
CommandManger
+ COMMANDS = 12 : const int
# *gameMap : GameMap
# *playerData : PlayerData
# *up : Up
# *down : Down
# *left : Left
# *right : Right
# *look : Look
# *fixRoom : FixRoom
# *talk : Talk
# *getQuest : GetQuest
# *finishQuest : FinishQuest
# *pickUp : PickUp
# *drop : Drop
# *fillAir : FillAir
# *heal : Heal
+ CommandManger (*PlayerData, *GameMap)
+ ~ CommandManger ()
+ RunCommand (input : string) : bool
+ SetCommandArray ()
+ ClearCommandArray ()
+ GetItemName (*Inventory) : string
PlayerData
- health : int
- oxygen : int
- score : int
- dead : bool
- *cords : Coordinate
- *inventory : Inventory
+ PlayerData ()
+ ~PlayerData ()
+ GetInventory () : Inventory*
+ GetCords () : Coordinate*
+ Die ()
+ UpdateOxygen (int)
+ UpdateHealth (int)
+ GetOxygen() : int
+ GetHealth() : int
+ Dead () : bool
Chef
<Concrete
+ Chef()
+ ~Chef()
+GetDialog()
+FinishQuest()
Medic
<concrete>
+Medic()
+~Medic()
+GetDialog()
+Heal(PlayerData)
+GetQuest() : Item*
Npc
<abrsact>
#Item* item
+Npc()
+~Npc()
+GetDialog()
+ FinishQuest()
+GetQuest() : Item*
+Heal(PlayerData)
NpcFact
+ field: type
+ NpcFact()
+~NpcFact()
+CreateNpc(NPCType) : Npc*
Captain
<Concrete
+ Captain()
+ ~Captain()
+GetDialog()
+FinishQuest()
Inventory
-empty : bool
map<Item*> contents
ItemStorage *itemFact
+ Inventory()
+ ~Inventory()
+ CheckReqItems(Inventory) : bool
+GetItem(itemName) : Item*
+Transfer(Inventory, itemName)
+AddItem(item*)
+AddItem(ItemType, name)
+RemoveItem(itemName)
+ListContents()
GlobalVaribles
+ Width = 5 : const int
+ Width = 4 : const int
+ RoomType:Enum
+ ItemType:Enum
+ NpcType:enum
ItemStorage
+ field: type
+ItemStorage()
+ ~ItemStorage()
+ CreateItem(ItemType) : Item*
Medal
<concrete>
+Medal()
+~Medal()
+GetDescription()
Item
<Absract>
#name : string
+Item()
+ ~Item()
+GetDescription()
+GetName() : string
Patch
<concrete>
+Patch()
+~Patch()
+GetDescription()
NpcType:enum
CHEF
MEDIC
CAPTAIN
RoomType:enum
REACTOR
STORAGE
BARACKS
INFERMARY
AIRLOCK
SEALOCK
REEF
ELEVATOR
TOPAIRLOCK
HALLWAYEND
ItemType:enum
EMPTY
REACTORPARTS
SHIPPARTS
SEACUCUMBER
DIVINGGEAR
AIRTANK
TOOLS
PATCH
MEDKIT
ReactorParts
<concrete>
+ReactorParts()
+~ReactorParts()
+GetDescription()
AirTank
<concrete>
+AirTank()
+~AirTank()
+GetDescription()
MedKit
<concrete>
+MedKit
+~MedKit
+GetDescription()
ShipParts
<concrete>
+ShipParts()
+~ShipParts()
+GetDescription()
SeaCucumber
<concrete>
+SeaCucumbe()
+~SeaCucumber()
+GetDescription()
Tools
<concrete>
+Tools()
+~Tools()
+GetDescription()
Coordinate
- xCord : int
+ Coordinate()
+ Coordnate(int x)
+ ~Coordinate()
+ UpdateX(int x)
+ GetX() : int
Commands
#map : *GameMap
#player : *PlayerData
#*Coordinates : Coordinate
+ Command()
+ ~Command()
+ Run()
+ Run(string)
Heal
+ GetQuest(*gamemap, *Coordnate)
+ Run()
Talk
+ Talk(*gamemap, *Coordnate)
+ Run()
PickUp
+ PickUp(*gamemap, *PlayerData)
+ Run(string)
Drop
+ Drop(*gamemap, *PlayerData)
+ Run(string)
FinishQuest
+ GiveItem(*gamemap, *PlayerData)
+ Run(String)
FillAir
+ FillAir(*PlayerData)
+ Run()
FixRoom
+ FixRoom(*GameMap, *PlayerData)
+ ~FixRoom ()
+ Run()
Up
+ Up(*GameMap, *PlayerData)
+ ~Up ()
+ Run()
Down
+ Down(*GameMap, *PlayerData)
+ ~Down ()
+ Run()
Left
+ Left(*GameMap, *PlayerData)
+ ~Left ()
+ Run()
Right
+ Right(*GameMap, *PlayerData)
+ ~Right ()
+ Run()
Look
+ Look(*gamemap, *Coordinate)
+ ~Look ()
+ Run()

Design – sequence diagrams

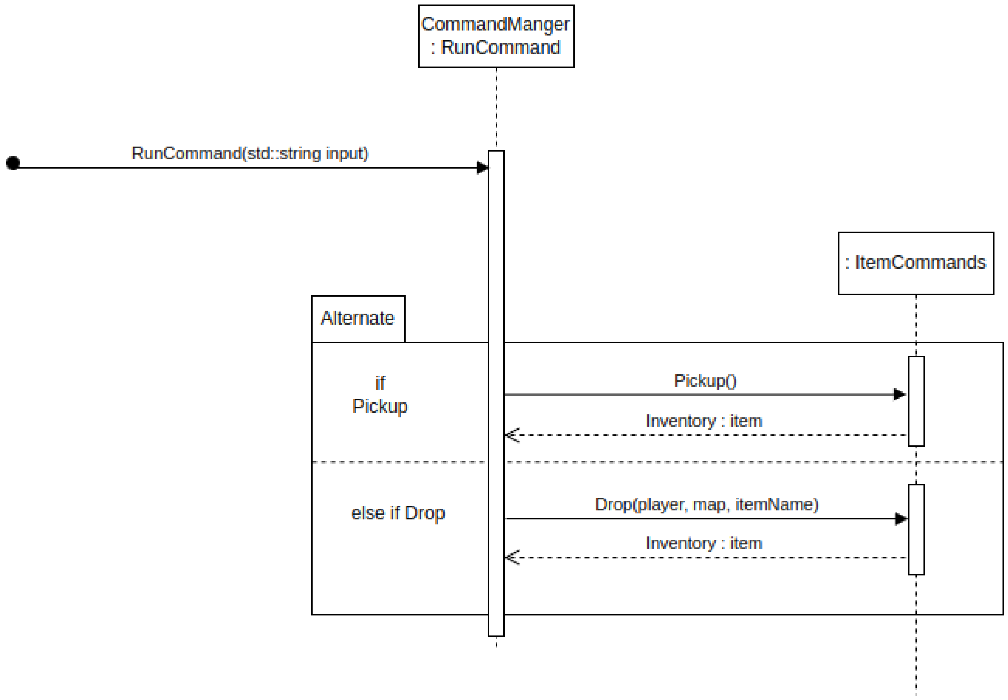
Environment


GameMap:Map
room :Environment
CommandManger: RunCommand
GetRoomCommand(int x, int y)
ProcessMapInput(string, GameMap, PlayerData
: RoomCommands
Alternate
if Moving Rooms
Up()
Down()
Left()
Right()
else if Looking in Room
Look()
else if Fixing Room / Solve puzzle success
FixRoom()
OStream

NPC


CommandManger
: RunCommand
RunCommand(std::string input)
:NPCCommands
Alternate
Talk()
if
Talk
FinishQuest()
else if
FinishQuest
else if
Heal
Heal()

Items + Inventory



Class Descriptions

Game Overview

Plot

- You are in a Submarine stuck in the middle of the ocean. The reactor of the Sub is broken and needs fixing.
- It is your goal to help and fix the problems in and around the submarine.
- Explore the ship, talk with your Crew Members and find a way to fix the Submarine and get it moving again.

- **Main Goal:** Find the [Shipparts] and [ReactorParts] under the Ocean and fix the [Reactor] of the Submarine.
- **Side Quest:**

Description

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

# name : string
# fixed : bool = false
# *inventory : Inventory
# *requiredItems : Inventory
# *npc : Npc = nullptr

Member functions:

# virtual SetContents () = 0
+ Environment ()
+ ~Environment ()
+ GetType () : string
+ virtual GetInfo () = 0
+virtual GetInventory () : Inventory*
+ virtual GetReqInventory () : Inventory*
+virtual GetNPC() : Npc*
+ virtual Fixroom (*Inventory)
+ virtual RoomFixed () : bool
+ virtual UpWall (*PlayerData) = 0
+ virtual DownWall (*PlayerData) = 0
+ virtual RightWall (*PlayerData) = 0
+ virtual LeftWall (*PlayerData) = 0

AirLock

-Concrete class

Member Functions:

+ AirLock()
+ ~AirLock()
+ GetInfo()
+ GetInventory ; Inventory*
+ GetReqInventory : Inventory*
+ CheckRepairItems(Inventory)
+ FixRoom(Inventory)
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

SeaLock

-Concrete class

Member Functions:

+ SeaAirLock
+ ~SeaAirLock
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Hallway

-Concrete class

Member Functions:

+ HallWay ()
+ ~HallWay ()
+ GetInfo ()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Cave

-Concrete class

Member Functions:

+ Cave()
+ ~Cave()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

SeaPlains

-Concrete class

Member Functions:

+ SeaPlains
+ ~SeaPlains
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Infirmary

-Concrete class

Member Functions:

+ Infirmary()
+ ~Infirmary()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)

- SetContents ()

Storage

-Concrete class

Member Functions:

+ Storage()
+ ~Storage()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

HallwayEnd

-Concrete class

Member Functions:

+ HallwayEnd()
+ ~HallwayEnd()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Bridge

-Concrete class

Member Functions:

+ Bridge()
+ ~Bridge()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

TopAirLock

-Concrete class

Member Functions:

+ TopAirLock()
+ ~TopAirLock()
+ GetInfo
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Reef

-Concrete class

Member Functions:

```
+ Reef()
+ ~Reef()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
```

Elevator

-Concrete class

Member Functions:

```
+ Elevator ()
+ ~Elevator ()
+ GetInfo ()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()
```

Barracks

-Concrete class

Member Functions:

+ Barracks()
+ ~Barrack()
+ GetInfo()
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

Reactor

-Concrete class

Member Functions:

+ Reactor()
+ ~Reactor()
+ Virtual GetInfo()
+ GetInventory: *Inventory
+ GetReqInventory: *Inventory
+ CheckRepairItems(Inventory)
+ FixRoom(Inventory)
+ UpWall (*PlayerData)
+ DownWall (*PlayerData)
+ RightWall (*PlayerData)
+ LeftWall (*PlayerData)
- SetContents ()

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
#reqItems: Items - the required items to fix
# description : string
# inventory : Inventory

Member functions:

+ Enviorment( )
+ ~Enviorment()
+ SetDescription(string)
+ GetInfo() : string
+ GetInventory(): * Inventory
+ FixRoom(*Inventory) : bool
+ RoomFixed() : bool
+ NorthWall(*Coordinate)
+ SouthWall(*Coordinate)
+ RightWall(*Coordinate)
+ LeftWall(*Coordinate)

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

- gameMap : vector<Environment*>
- *mapGen : MapGen

Member functions:

+ GameMap ()
+ ~GameMap ()
+ CreateMap ()
+ DestroyMap ()
+ GetRoom (*Coordinate) : Environment*
+ GetRoom (int, int) : Environment*

+ GetRoomContents (*Coordinate) : Inventory*
+ GetRoom (int) : Environment*

MapGen

- Creates rooms based on enumeration type given.

Member functions:
+ MapGen()
+ ~MapGen()
+CreateMapLocation(RoomType) : Environment*

Coordinate

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

Variables
- xCord : int

Member functions:
+ Coordinate()
+ Coordnate(int x)
+ ~Coordinate()
+ UpdateX(int x)
+ GetX() : int

PlayerData

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

Variables

- cords : Coordinate
- Inventory : Inventory
- score : int
- fixedReactor : bool = false

Member functions:

+ PlayerData()
+ ~PlayerData()
+ GetInventory() : *Inventory
+ GetCords() : *Coordinate
+ SetFixed(bool);

//+ GetScore(): int
//+ UpdateO2(int)
//+ GameOver(Int death):ostream

Inventory

- Manages inventory management for locations, characters and player

Variables

-empty : bool
map<Item*> contents
ItemStorage *itemFact

Member functions:

+ Inventory()
+ ~Inventory()
+ CheckReqItems(Inventory) : bool
+GetItem(itemName) : Item*
+Transfer(Inventory, itemName)
+AddItem(item*)
+AddItem(ItemType, name)
+RemoveItem(itemName)
+ListContents()

ItemStorage

Member functions:
+ItemStorage()
+ ~ItemStorage()
+ CreateItem(ItemType) : Item*

Item

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

Variables
#name : string
Member functions:
+Item()
+ ~Item()
+GetDescription()
+GetName() : string

ReactorParts

Member functions:
+ReactorParts()
+~ReactorParts()
+GetDescription()

MedKit

Member functions:
+MedKit
+~MedKit
+GetDescription()

SeaCucumber

Member functions:
+SeaCucumbe()
+~SeaCucumber()
+GetDescription()

AirTank

Member functions:
+AirTank()
+~AirTank()
+GetDescription()

ShipParts

Member functions:
+ShipParts()
+~ShipParts()
+GetDescription()

Tools

Member functions:
+Tools()
+~Tools()
+GetDescription()

Medal

Member functions:
+Medal()
+~Medal()
+GetDescription()

Patch

Member functions:
+Patch()
+~Patch()
+GetDescription()

NpcFact

Member functions:
+ NpcFact()
+~NpcFact()
+CreateNpc(NPCType) : Npc*

NPC

Variables
#Item* item
Member functions:
+Npc()
+~Npc()
+GetDialog()
+ FinishQuest()
+GetQuest() : Item*
+Heal(PlayerData)

Chef

Member functions:
+ Chef()
+ ~Chef()
+GetDialog()
+FinishQuest()

Medic

Member functions:
+Medic()
+~Medic()
+GetDialog()
+Heal(PlayerData)
+GetQuest() : Item*

Captain

Member functions:
+ Captain()
+ ~Captain()

+GetDialog()
+FinishQuest()

Interaction

- Handles the definitions for the games various interactions, this is the base class which has through children who inherit from it.

+ Interaction()
+ ~Interaction()
+ ProcessMapInput(const &string,
*GameMap, *PlayerData) : bool
+ ProcessInventoryInput(const &string,
*GameMap, *PlayerData) : bool
+ North(*GameMap, *PlayerData)
+ East(*GameMap, *PlayerData)
+ South(*GameMap, *PlayerData)
+ West(*GameMap, *PlayerData)
+ Talk(*Npc) : String
+ PickUp(*PlayerData, *GameMap, const string)
+ Drop(*PlayerData, *GameMap, const string)
+ Look(*GameMap, *PlayerData)
+ Fix(*GameMap,*PlayerData)
+ ListInventory(Inventory)
+ ListInventory(*Inventory)

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

- Playing = false : bool
- *gameMap : GameMap
- *player : PlayerData
- *commandHandler : CommandManger
- input : string

Member functions:

+ UnderUs ()
+~UnderUs ()
+MainMenu ()
+ GetCommands ()
+ GetHelp ()
+ BackStory ()
+ Play ()
+ Exit ()
+ SendCommand ()
+ GameOver ()
+ GameWon ()

Exceptions

- The game's error handling class.

Member Functions:

+Invalid_parameter_error(const char* errMessege)
+Calculation_error(const char* errMessage)

Constants

- Class for room enumeration.

+ Room : enum {
HallWay,

Reactor
Other
}