David Ramirez

6/13/17

Final Project Class Design

Base class (for ColChamber, ComRoom, LoadBay, BioLab, MedBay, NavRoom):

Room class (abstract class)

Represents a room of a spaceship

- Protected: (data members)
  - String nameR
    - Holds the derived class name
  - Room* door1
    - Pointer to the Room class that represents a room that this room is connected to
  - Room* door2
    - Pointer to the Room class that represents a room that this room is connected to
  - Room* door3
    - Pointer to the Room class that represents a room that this room is connected to
  - Room* door4
    - Pointer to the Room class that represents a room that this room is connected to
- Public: (member functions)
  - Virtual void setDoor1(Room &room1)
    - Sets Room pointer door1 to address of room passed in parameter
  - Virtual void setDoor2(Room &room2)
    - Sets Room pointer door2 to address of room passed in parameter
  - Virtual void setDoor3(Room &room3)
    - Sets Room pointer door3 to address of room passed in parameter
  - Virtual void setDoor4(Room &room4)
    - Sets Room pointer door4 to address of room passed in parameter
  - Virtual Room* getDoor1()
    - Returns pointer to Room pointed to by door1 data member
  - Virtual Room* getDoor2()
    - Returns pointer to Room pointed to by door2 data member
  - Virtual Room* getDoor3()
    - Returns pointer to Room pointed to by door3 data member
  - Virtual Room* getDoor4()
    - Returns pointer to Room pointed to by door4 data member
  - Virtual string getDoor1Name();

        - Returns string holding the name of the Room pointed to by door1 data member
    - Virtual string getDoor2Name();
        - Returns string holding the name of the Room pointed to by door2 data member
    - Virtual string getDoor3Name();
        - Returns string holding the name of the Room pointed to by door3 data member
    - Virtual string getDoor4Name();
        - Returns string holding the name of the Room pointed to by door4 data member
    - Virtual string getNameR()
        - Returns string holding name of this room

Derived classes of Room

BioLab class

Represents a biology laboratory on a space ship

- Public: (data members)
    - None, other than those in Room base class
- Public: (member functions)
    - All Room class member functions
    - BioLab() constructor
        - Sets nameR data member to reflect name of room (Biology Laboratory)
        - Sets door1 pointer to Room to Null
        - Sets door2 pointer to Room to Null
        - Sets door3 pointer to Room to Null
        - Sets door4 pointer to Room to Null
    - Void showDescR()
        - Displays the description of this room
        - Overridden from pure virtual function
    - XenDet* special2()
        - Dynamically allocates a new XenDet (xenomorph deterrant) item and returns a pointer to the new item

ColChamber class

Represents a large room full of crew members in stasis pods on a space ship

- Private: (data members)
    - All data members of Room base class
    - Int descChange
        - Int that records whether the in game event of unlocking pilot Tennessee has occurred or not (0 = not occurred, 1 = occurred)
- Public: (member functions)

- All member functions of Room base class
- ColChamber() constructor
  - Sets nameR data member to reflect the name of this room (Colony Chamber)
  - Sets door1 pointer to Room to Null
  - Sets door2 pointer to Room to Null
  - Sets door3 pointer to Room to Null
  - Sets door4 pointer to Room to Null
- Void showDescR()
  - Display the description of this room
  - Overridden from pure virtual function
- Int special1()
  - Displays a message that states that Tennessee's stasis pod is locked
  - Only called by Game class when player has not yet accessed the access code to unlock the pod
  - Overridden from pure virtual function
- Void unlock2()
  - Displays a message that states that Tennessee's stasis pod has been opened and Tennessee will help the player by giving them a medkit every 4 turns
  - Only called when player has accessed the access code (from option 2 of communications room)
  - Increments the descChange integer that keeps track of the event of unlocking Tennessee
- Void talk()
  - Displays a sentence representing dialogue from Tennessee
  - 1 of 4 dialogues randomly selected

ComRoom class

Represents a communications room of a space ship

- Private: (data members)
  - All data members of Room base class
  - Int compAccess
    - Tracks whether player has accessed the access code from the terminal (option 2 of the communications room)(in game event)
  - int speaker
    - Tracks whether player has accessed the comms speaker (option 1 of the communications room) (in game event)
  - Int code
    - Keeps track of the code that the player accesses from option 1 of the navigation room (in game event)
  - Int noTurn
    - Keeps track of whether the Game class should take a turn or not

        - Incremented to 1 by the coord() member function when a turn should not be taken
- Public: (member functions)
    - All member function of Room base class
    - Comroom() constructor
        - Sets nameR data member to reflect name of room (Communications Room)
        - Sets data members that keep track of in game events to 0
            - These include int compAccess, int code, int speaker, and int noTurn
        - Sets door1 pointer to Room to Null
        - Sets door2 pointer to Room to Null
        - Sets door3 pointer to Room to Null
        - Sets door4 pointer to Room to Null
    - Void showDescR()
        - Displays the description of the room
        - Overridden from pure virtual function
    - Int coord(int,int)
        - Parameters are the code that the player must match (accessed in the navigation room option 1) and the number of turns
        - If the code has been retrieved from the navigation room, a prompt to enter the code is displayed
        - If the code has not been retrieved from the navigation room, a prompt is displayed to retrieve it
        - If the objective of telling HQ the code is complete, a message stating there is nothing here is displayed
        - Returns int to inform Game class whether to take a turn or not (return 0 to take a turn, return -1 to not take a turn)
    - Int special1()
        - Displays message stating that player has found access code (option 2 in comms room)
        - Sets compAccess to 1 to track this in game event
        - If this option was already accessed, output no new info message and return -1 so Game class does not take a turn
        - Otherwise, returns 0 so that Game class takes a turn
        - Overridden from the pure virtual function
    - Int getNoTurn()
        - Returns the data member noTurn, which keeps track of it the Game class should take a turn for the coord member function
    - Int getSpeaker()
        - Returns data member speaker, which keeps track of whether the player has accessed the comms speaker (option 1 in comms room)

LoadBay class

Represents the loading bay of a space ship

- Private: (data members)
  - None, other than those of base class Room
- Public: (member functions)
  - All member functions of base class Room
  - LoadBay() constructor
    - Sets nameR data member to reflect the name of this room (Loading Bay)
    - Sets door1 pointer to Room to Null
    - Sets door2 pointer to Room to Null
    - Sets door3 pointer to Room to Null
    - Sets door4 pointer to Room to Null
  - Void showDescR()
    - Displays the description of the room
    - Overridden from pure virtual function
  - MineTool* special2()
    - Dynamically allocates new mining tool item and returns pointer to new item
    - Overridden from pure virtual function

MedBay class

Represents a medical bay of a space ship

- Private: (data members)
  - None, other than those in base class Room
- Public: (member functions)
  - All member functions of base class Room
  - MedBay() constructor
    - Sets nameR data member to reflect the name of the room (Medical Bay)
    - Sets door1 pointer to Room to Null
    - Sets door2 pointer to Room to Null
    - Sets door3 pointer to Room to Null
    - Sets door4 pointer to Room to Null
  - Void ShowDescR()
    - Displays description of this room
  - Int heal (Daniels &)
    - Restores up to 4 hp if Daniel's hp is lower than 10
    - Outputs message if player health at max
    - Returns 0 when hp recovered so that Game class takes turn
    - Returns -1 when player at full health so that Game class does not take turn
    - Function knows health of player because player object (Daniels) is passed to it by reference
  - Medkit* special2()

        - Dynamically allocates a new Medkit item and returns a pointer to the new item

NavRoom class

Represents the navigation room of a space ship

- Private: (data members)
    - All data members of base class Room
    - Int coordGot
        - Keeps track of if player has accessed coordinate code (option 1 in nav room)
        - 1 if accessed, 0 if not
- Public: (member functions)
    - NavRoom() constructor
        - Set nameR data member to reflect name of room (Navigation room)
        - Set coordGot data member to 0, reflecting that the player has not accessed the coordinates yet
        - Sets door1 pointer to Room to Null
        - Sets door2 pointer to Room to Null
        - Sets door3 pointer to Room to Null
        - Sets door4 pointer to Room to Null
    - Void showDescR()
        - Displays description of room
        - Overridden from pure virtual function
    - Int special1()
        - Generates a random code
        - Displays and saves this code to be re-entered at the comms room
        - Sets coordGot data member to 1 to show that the player has accessed the coordinates
        - If the coordGot ==1 when this function is called, a message stating there is no new info is displayed, and -1 is returned so that the Game class does not take a turn
        - Otherwise, returns 0 so that the Game class takes a turn
        - Overridden from pure virtual function
    - Void window (int turn)
        - Displays a graphic of the outside of the ship
        - Graphic changes based on player progression (hence, passing the turn limit in the parameter)

<u>Base class (for Medkit, MineTool, and XenDet)</u>

Item class (abstract class)

Represents an item that the player uses to help them survive the game

- Protected: (data members)

    - String nameI
        - Holds the name of the item
- Public: (member functions)
    - Virtual string getNameI()
        - Returns the nameI data member that holds the item name
    - Virtual void showDescI()
        - Display description of item
        - Not implemented due to time constraints

Derived classes of Item

Medkit class

Represents a medkit, carried in inventory, that the player can use to heal 3 hp

- Private: (data members)
    - None, other than those in base class item
- Public: (member functions)
    - All Item base class member functions
    - Medkit() constructor
        - Sets data member nameI to reflect name of item (Medkit)
    - Void showDescI()
        - Not implemented due to time constraints

MineTool class

Represents a mining tool item, carried in inventory, that the player can use for defense

- Private: (data members)
    - None, other than those in base class Item
- Public: (member functions)
    - All Item base class member functions
    - MineTool() constructor
        - Sets nameI data member to reflect the name of this item (mining tool)
    - ShowDescI()
        - Not implemented due to time constraints

XenDet class

Represents a xenomorph deterrent item, carried in inventory, the player can use to protect themselves from the enemy for 2 turns

- Private: (data members)
    - None, other than those in base class Item
- Public: (member functions)
    - All item base class member functions
    - XenDet() constructor

        - Sets nameI data member to reflect name of item (xenomorph deterrent)
    - showDescI()
        - not implemented due to time constraints

Base class (for Xenomorph)

Enemy class

Represents an enemy aboard the space ship that attacks the player

- protected: (data members)
    - string name
        - holds the name of the enemy
    - Room* location
        - Holds a pointer to the room that is the enemies current location
- Public: (member functions)
    - Virtual string getNameE()
        - Returns the name data member which holds the name of the enemy
    - Virtual void setLocation (Room &)
        - Sets the location data member to point to the address of the Room class object passed by reference in the parameter
    - Virtual void move()
        - Moves the enemy randomly to the room pointed to by door1, door2, door3, or door4 of the current room location
    - Virtual Room* getLocation()
        - Returns a pointer to the Room class object that is the current location of the enemy

Derived class of Enemy

Xenomorph

Represents a xenomorph alien enemy that is on the space ship and attacks the player

- Private: (data members)
    - None, other than data members of base class Enemy
- Public: (member functions)
    - Xenomorph() constructor
        - Sets nameE data member to reflect the name of the enemy (xenomorph)
        - Sets location data member to point to Null
    - Int attack()
        - Generates a random number between 5 and 2 and returns it
        - The number represents attack power done by the enemy

Other classes

Daniels class

Represents the player's character throughout the game

- Private: (data members)
  - List <Item*> inventory
    - List of pointers to Item class objects to represent an inventory that the player takes with them
  - List <Item*> iterator iter
    - Iterator to access items in inventory
  - Int health
    - Integer representing the players health
    - Starts at 10 and max is 10
  - Int medCheck
    - Integer representing whether or not part of a medkit was wasted (ie. that the medkit was not used to the full potential)
  - Int det
    - Integer that keeps track of the number of xenomorph deterrant (XenDet) items are in player inventory
  - Int mine
    - Integer that keeps track of the number of mining tool (MineTool) items are in the player inventory
  - Int meds
    - Integer that keeps track of the number of medkit (Medkit) items are in the player inventory
  - Int movetrack
    - Integer that keeps track of whether or not the player moved the last turn
    - Set to 1 if the player moved last turn
    - Set to 0 if the player did not move last turn
    - Used by play() member function of Game class to assess whether the player will be able to hide when they encounter the enemy (xenomorph)
  - Room* location
    - Pointer to the Room class that represents the location of the player
- Public: (member functions)
  - Daniels() constructor
    - Sets health data member to 10
    - Sets location data member to point to Null
    - Sets medCheck data member to 0
    - Sets det data member to 0 to show no inventory
    - Set mine data member to 0 to show no inventory
    - Set meds data member to 0 to show no inventory
    - Set moveTrack data member to 0 to show no movement
  - Void modHealth(int inc)
    - Increases the players health by the number represented by the integer parameter (max 4)

        - Makes certain that player health is never above 10
    - Void addItem(Item*)
        - Adds a pointer to an Item class object to the list that represents the player inventory
        - Displays message stating what was added to inventory
        - Increments either det, mine, or meds data members corresponding to what item was added to the inventory
    - Void removeMine()
        - Removes a MineTool (mining tool) item pointer from the list that represents the player inventory and deletes it
        - Decrements the mine data member that tracks the number of mining tool items are in player inventory
    - Void showInvent()
        - Displays the current inventory of the player
        - Ie. displays the names of the items pointed to by the pointers held in the inventory list
    - Void setLocation(Room &)
        - Sets location data member to point to the room passed by reference in the parameter
    - Int getHealth()
        - Returns the current health points of the player indicated by the health data member
    - Void damage(int dmg)
        - Deducts the amount passed in parameter from players health
        - Represents the player taking damage from an enemy attack
    - String getLocationName()
        - Returns the name of the current location of the player
        - Ie. returns the name of the room pointed to by the location data member
    - Room* getLocation()
        - Returns a pointer to the room that is the players current locations
        - Ie. returns a pointer to the room pointed to by the location pointer
    - Int useMedKit()
        - Calculates amount of health missing and regains up to 3 health for the player if the player has a medkit in inventory
        - Deletes the medkit after use
        - Outputs a message if there are no medkits in inventory, and returns -1 so that Game class does not take a turn
        - Returns 0 if medkit successfully used
    - Int useXenDet()
        - Uses a xenomorph deterrent the player has in inventory to protect player from enemy for 2 turns
        - If there is no xenomorph deterrent in inventory, a message is displayed, and -1 is returned so that the Game class does not take a turn

        - Deletes xenomorph deterrant after use
        - Return 0 if xenomorph deterrent successfully used
    - Int getDet()
        - Returns the number of xenomorph deterrents in player inventory
        - Ie. returns the value of data member det
    - Int getMine()
        - Returns the number of mining tool items in player inventory
        - Ie. returns the value of data member mine
    - Int getMeds()
        - Returns the number of medkit items in player inventory
        - Ie. returns the value of data member meds
    - Void move()
        - Sets location data member to point to room pointed to by door1 data member of current location room
        - Used after the enemy attacks, to simulate the player running away
    - Int getMoveTrack()
        - Returns the integer that represents whether the player has moved in the last turn
        - Used by the play() member function of the Game class to assess whether to give the player the chance to hide or not
    - Void setMoveTrack(int set)
        - Sets the moveTrack data member to the value passed in parameter
        - 1 = player has moved last turn
        - 0 = player has not moved last turn
    - Void removeAll()
        - Deletes all dynamically allocated memory held in the user inventory
        - Called at the end of the program

Game class

Represents an instance of the game and keeps track of all game events

- Private: (data members)
    - ColChamber colR
        - ColChamber class object
    - ComRoom comR
        - ComRoom class object
    - LoadBay loadR
        - LoadBay class object
    - MedBay medR
        - MedBay class object
    - NavRoom navR
        - NavRoom class object
    - BioLab bioR
        - BioRoom class object
    - Daniels danP

        - Daniels class object (player)
    - Xenomorph xenE
        - Xenormorph class object
    - Xenormorph xenE2
        - Xenormorph class object
    - Xenomorph xenE3
        - Xenomorph class object
    - Int turnLimit
        - keeps track of turns to count down from until rescue arrives (after player completes final objective)
    - int turnNumber
        - keeps track of number of turns that have elapsed since the game began
    - int lost
        - keeps track of whether or not the player has lost the game or not
        - 1 if lost, 0 if not
        - Used to keep the program from outputting win game messages if the player lost
    - Int code
        - Keeps track of the randomly generated code that the player must access in the nav room and relay in the comms room
    - Int contact
        - Keeps track of whether or not the player has made contact with HQ
        - 1 if contact, 0 if not
    - Int trashCode
        - Used to store irrelevant code if coord function of nav room called multiple times
    - Int codeCheck
        - Keeps track of whether or not the player has accessed the code in the nav room
    - Int activate
        - Keeps track of whether or not the player has completed the main objective of obtaining and relaying the coordinate code to HQ for rescue
        - Starts counting down turnLimit each turn
        - 1 if objective is completed, 0 if not
    - Int gotAccess
        - Keeps track of whether or not the player has accessed the access code in the terminal in the comms room (option 2)
        - 1 if has accessed, 0 if not
    - Int tenCheck
        - Keeps track of whether or not the player has unlocked Tennessee from the colony chamber using the access code retrieved from the comms room
        - 1 if unlocked, 0 if not

        - If Tennessee is unlocked, a medkit will be given to the player every 4 turns
    - Int xenodet
        - Keeps track of how many turns the player is safe from the enemy after using the xenomorph deterrent item
        - Is decremented each turn
    - Int quit
        - Keeps track of whether the player decided to quit the game
        - Is used to make sure that winning or losing messages do not appear if the player quits the game
    - Int madeXenDet
        - Keeps track of how many turns the player must wait before making another xenomorph deterrent
        - Is decremented each turn
    - Int madeMineTool
        - Keeps track of how many turns the player must wait before making another mining tool
        - Is decremented each turn
    - Int thirdSet
        - Keeps track of whether or not the third enemy has come into play
        - Third enemy comes into play 2 turns after the final objective is complete
        - Third enemy makes game a little harder
    - Int oneMore
        - Keeps track of whether or not the message displaying that another enemy has arrived has been displayed or not
        - 1 if it has been displayed, 0 if not
- Public: (member functions)
    - Game() constructor
        - Sets turnLimit to 18
        - Sets turnNumber to 1
        - Sets all variables that keep track of in game events to 0 to represent beginning of game
        - Use set functions for all 6 Room derived class members to link the rooms together with pointers (4 pointers per room)
        - Sets the player location to the colony chamber
        - Sets one enemy location to start in the loading bay, and the second to enemy location to start in the navigation room
            - The player is not supposed to know there are 2 aliens at this point, but having only 1 alien did not produce enough encounters
    - Void preface()
        - Displays the preface of the game
    - Void play()
        - Facilitates the turns of the game and the encounters with the enemy

        - Gives player medkit every 4th turn if Tennessee is unlocked (in game event)
        - If player and enemy are ever in the same room, the enemy will attack
            - Depending on whether the player has moved the last turn or not, the player may be given a chance to hide
        - Displays a message and ends the game if the player wins (ie. it turnLimit reaches 0 and the player is still alive
        - Displays a message if the player loses the game
        - Is responsible for decrementing these variables every turn if they are above 0
            - madeXenDet
            - madeHeal
            - madeMineTool
            - xenoDet
        - calls the turn() function so the player can choose what to do next
        - deletes all dynamically allocated inventory items at the end of the game by calling the removeAll() functions of the Daniels class
    - void Turn()
        - shows the player HUD (heads up display) each turn
            - map
            - objective
            - health
        - uses menu1 function to retrieve player input on what they would like to do
            - move
            - look around room
            - access inventory
            - quit
        - whatever the user integer choice corresponds to is what function is called
        - if choice is to look around the room, options and descriptions will vary depending on the location of the player
        - if choice is to move player, choices of room to move to will vary depending on player location
    - int move()
        - called by the turn() function if player decides to move
        - uses menu3 to determine where player wants to move to
            - options will vary depending on location
        - uses the setLocation() function of the Daniels class to move the player corresponding to player choice
        - if player successfully moved, moveTrack is set to 1 to show that the player moved
        - returns 0 if move successful to let Game class function turn() know to take a turn

        - returns -1 if player decides to select back, letting Game class function turn() know to not take a turn
    - int med
        - called by the turn() function if the user decides to look around the room in the MedBay
        - Uses menuMed() function to retrieve user input as to what they would like to do in the medbay
        - If user decides to heal, player is healed for 4hp and madeHeal is set to 3, making the player wait 3 turns to heal again
        - If the player attempts to heal and is already at full health, a message is displayed and -1 is returned, letting the turn() function know to not take a turn
        - Calls MedBay class functions
        - If the player decides to make a medkit, a medkit is added to the inventory, madeMedKit is incremented, and 0 is returned to let the turn() function know to take a turn
            - The player cannot make more than 4 medkits per game
        - If the player attempts to make a medkit and the inventory is full, a message is displayed and -1 is returned, letting the turn() function know to not take a turn
        - returns -1 if player decides to select back, letting Game class function turn() know to not take a turn
    - int Com
        - called by the turn function if the player decided to look around the room in the communications room
        - uses the menuCom() function to retrieve user input as to what they would like to do in the com room
        - calls ComRoom class functions
        - if player decides to access the comm speaker, the appropriate functions are called depending on objectives complete and the contact data member is set to 1
        - if player accesses comm speaker or terminal multiple times, successive times will display a no new info message and -1 will be returned so the turn() function does not take a turn
        - if player decides to access the terminal, a message about a access code is displayed and gotAccess is set to 1, showing that the player can open Tennessee's stasis pod now
        - returns -1 if player decides to select back, letting Game class function turn() know to not take a turn
    - int bio
        - called by the turn() function if the player decided to look around the room in the biology laboratory
        - uses the menuBio() function to retrieve user input as to what they would like to do in the biology lab

        - if user chooses to make a xenomorph deterrent and the inventory isn't full and they haven't made on in a recent turn, the appropriate BioLab class function will be called to add a new xenomorph deterrant to the inventory
        - if the inventory is full or the player chooses the back option, -1 is returned to let the turn() function know to not take a turn
    - int load
        - called by the turn() function if the player decided to look around the room in the loading bay
        - uses the menuLoad() function to retrieve user input as to what they would like to do in the loading bay
        - if the user would like to make a mining tool and the inventory isn't full and they haven't made one in recent turn, the appropriate LoadBay class functions will be called and a new mining tool item will be added to the inventory
        - after the player creates a new mining tool, the madeMineTool data member is set to 4, making the player wait 4 turns to make another
        - if the inventory is full or the player chooses the back option, -1 is returned to let the turn() function know to not take a turn
    - int nav()
        - called by the turn() function if the player decided to look around the room in the navigation room
        - uses the menuNav() function to retrieve user input as to what they would like to do in the navigation room
        - it the user chooses the access the nav terminal, the approtpriate NavRoom class functions will be called to generate, display, and save a code (representing the coordinates of the ship)
            - code must be reentered by player in com room
            - codeCheck data member will be set to 1
        - if the player chooses to look out the window, the window() function of the NavRoom class will be called and a graphic will be displayed
        - if the player chooses the back option, -1 is returned to let the turn() function know to not take a turn
    - int col()
        - called by turn() function if the player decided to look around the room in the colony chamber
        - uses the menuCol() function to retrieve user input as to what they would like to do in the colony chamber
        - If user has access code, they can unlock Tennessee, otherwise a message will display that need an access code
        - Once Tennessee is unlocked, the user can choose to talk to him and the appropriate ColChamber class function will be called to display dialogue
    - Void map1()
        - Displays map showing player in colony chamber

- Void map2()
  - Displays map showing player in comms room
- Void map3()
  - Displays map showing player in med bay
- Void map4()
  - Displays map showing player in loading bay
- Void map5()
  - Displays map showing player in bio lab
- Void map6()
  - Displays map showing player in navigation room
- Void space()
  - Displays 60 lines of blank space
- Void objective
  - Displays the player's current objective
  - Changes according to in game events