

**COE312 Software Design for Engineers**

**Homework 3: Design Patterns**

**Academic Integrity Pledge**

|  |
| --- |
| As a student of American University of Sharjah, I here by state that I will abide by the AUS Integrity Pledge that:   * I will hold myself accountable for all that I say and write. * I will hold myself responsible for the academic integrity of my work * I will not carry out unauthorized copying or printing of the work of others * I will not misrepresent my work nor give or receive unauthorized aid * I will behave in a manner that demonstrates concern for the personal dignity, rights and freedoms of all members of the community * I will respect university property and the property of others; and * I will not tolerate a lack of respect for these values.   **Student Name: Amir Mohideen**  **Student ID: 74559**  **Student Name:**  **Student ID:** |

**Note: You must submit the code files as well.**

**Github link:** https://github.com/COE312L/HW3

1. **State how the software design patterns can be used to implement the game.**

We should use the template pattern to implement the potion making process since it is a recipe with one similar instruction while others are different

We should use the Observer pattern where the main console that receives input should be the observer while the clock should be the subject

We can use command pattern for player commands like look around and displaying win message after spells are used

1. **Implement the game using the software design patterns and threads.**

* The Java code

*Please add subtitles with the class names. Paste only the new code, and* ***not*** *screenshots of the code.*

//main class that has all the thread initializations

//I assume Harry thread is not required since the question doesn’t mention him but only player throughout the question

**import** java.util.\*;

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** main {

**public** **static** **void** main(String[] args) **throws** InterruptedException{

// **TODO** Auto-generated method stub

//synchronized class objects for hermione and harry

loc l = **new** loc();//to know location

pot p = **new** pot();//to know how many potions made by player

win wi = **new** win();// to know if player defeated Malfoy in library

//time t = new time();

Watch w = **new** Watch(); //timer

mainthread main = **new** mainthread(l,p,wi); //main thread

Harry h = **new** Harry(w,l,p,wi); //harry thread

Hermione her = **new** Hermione(l,p,wi); // hermione thread

}

}

**// the main thread class.**

**//Changes: checks if player makes 2 potions of same type and stores it in a class that hermione checks constantly, also added the function call for when a spell is cast by player**

**import** java.util.Scanner;

**public** **class** mainthread **implements** Runnable{

**int** potion1=0,potion2 =0;

loc loc;

pot pot;

win win1;

Thread t;

mainthread( loc l1,pot p1,win wmain) {

**this**.loc = l1;

**this**.pot = p1;

**this**.win1 = wmain;

t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

//Initializations

Library l=**new** Library(win1);

DiningHall d=**new** DiningHall();

PotionsClassroom p=**new** PotionsClassroom();

//we assume the user types all input in lowercase

// For password

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("“Messrs Moony, Wormtail, Padfoot and Prongs Purveyors of Aids to magical Mischief-Makers are proud to present THE MARAUDER'S MAP”\n" +

"Say the magic phrase to reveal the map.\n");

String input = in.nextLine();

**while**(!input.contains("i solemnly swear that i am up for no good")) {

System.***out***.println("Wrong Password! Try Again:\n");

input = in.nextLine();

}

//while loop to continue asking user questions until user decides to exit by typing 'mischief managed'

loc.setState("Entrance"); //to enter switch case

**while**(!input.contains("mischief managed")) {

**switch**(loc.getState()) {

//contains 4 cases with 4 locations: Entrance, Dining Hall, Library, Potions Classroom

**case** "Entrance":

**while**(loc.getState().contains("Entrance")) {

System.***out***.println("The Entrance\n" +

"You arrive at the doors of Hogwarts. The door on the north wall leads to the dining hall,\n"+

"the door to the east leads to the Potions class, and the door to the west leads to the Library.");

System.***out***.println("Which direction would you like to walk to? (walk north/east/west)\n");

input = in.nextLine();

**if**(input.contains("north")) {

System.***out***.print("You have entered the dining hall.\n");

loc.setState("Dining Hall");

**break**;

}

**else** **if**(input.contains("west")) {

System.***out***.print("You have entered the Library.\n");

loc.setState("Library");

**break**;

}

**else** **if**(input.contains("east")) {

System.***out***.print("You have entered the potions class.\n");

loc.setState("Potions Classroom");

**break**;

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input! Try again\n");

}

}

**case** "Dining Hall":

**while**(loc.getState().contains("Dining Hall")) {

input = in.nextLine();

**if**(input.contains("look")) {

d.look();

}

**else** **if**(input.contains("expelliarmus") || input.contains("stupefy") ||input.contains("protego")||input.contains("expecto patronum"))

{d.win();}

// else if(input.contains("talk")) {

// if(input.contains("dumbledore")) {

// d.talk();

// }

// else {

// System.out.print("This character is not here.\n");

// }

// }

**else** **if**(input.contains("use")) {

**if**(input.contains("hat")) {

d.use("sorting hat");

}

**else** **if**(input.contains("plate")) { //changed to plate to make a meaningful sentence in output

d.use("plate");}

**else** {

System.***out***.print("You can only use a sorting hat or plate.\n");

}

}

**else** **if**(input.contains("north")||input.contains("east")||input.contains("west")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}

**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input");

}

}

**break**;

**case** "Library":

**while**(loc.getState()=="Library") {

input = in.nextLine();

**if** (input.contains("look")) {

l.look();

}

**else** **if**(input.contains("expelliarmus") || input.contains("stupefy") ||input.contains("protego")||input.contains("expecto patronum"))

l.win();

// else if(input.contains("talk")) {

// if(input.contains("hermione")) {

// l.talk();

// }

// else {

// System.out.print("This character is not here!\n");}

// }

**else** **if**(input.contains("use")) {

**if**(input.contains("book")) {

l.use("book");

}

**else** **if**(input.contains("quill")) {

l.use("quill");

}

**else** {

System.***out***.print("You can only use a book or quill.\n");}

}

**else** **if**(input.contains("north")||input.contains("west")||input.contains("east")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input!\n");

}

}

**break**;

**case** "Potions Classroom":

**while**(loc.getState()=="Potions Classroom") {

input = in.nextLine();

**if** (input.contains("look")) {

p.look();

}

**else** **if**(input.contains("expelliarmus") || input.contains("stupefy") ||input.contains("protego")||input.contains("expecto patronum"))

p.win();

// else if(input.contains("talk")) {

// if(input.contains("snape")) {

// p.talk();

// }

// else {

// System.out.print("This character is not here!\n");

// }

// }

**else** **if**(input.contains("make")) {

**if**(input.contains("polyjuice")) {

p.use("polyjuice");

potion1++;

**if**(potion1==2)

{pot.setState(potion1);}//indicates how many polyjuice potions player has for hermione to know

}

**else** **if**(input.contains("invisibility")){

p.use("invisibility");

potion2++;

**if**(potion2==2)

{pot.setState(potion2);}//indicates how many invisibility potions player has for hermione to know

}

**else** {

System.***out***.print("You can only make a polyjuice or a invisibility potion.\n");}

}

**else** **if**(input.contains("north")||input.contains("west")||input.contains("east")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}

**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if**(input.equals("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input!\n");

}

}

**break**;

**default**: System.***out***.print("Incorrect input!\n");

}

}

in.close();

System.***out***.print("Hiding map contents...end\n");

}

}

//harry thread that checks for time and also finds hermione and says he finds hermione only after he is in library with 2 potions and wins over malfoy

**public** **class** Harry **extends** ConcreteObserver **implements** Runnable{

loc l;

pot p;

win w;

Harry(Subject subject,loc l1,pot p1,win w1) {

**super**(subject);

l=l1;

p=p1;

w=w1;

Thread t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

**while**(**true**)

{

**if**(l.getState()=="Library") {

**if**((p.getState())== 2)

{

**if**((w.getState())== **true**){

System.***out***.println("Found Hermione!");

**break**;

}

}}

}

}

**public** **void** update(**int** sec) {

**if** (sec>5)

{

System.***out***.println( "You have been caught... and the Marauder’s map has been confiscated.");

System.*exit*(0);

}

System.***out***.println( sec +" minute passed...");

**if**(sec==5)

{

System.***out***.println( "Hide map contents before you get caught!...");

}

}

}

**//Hermione thread that reveals herself only if player is in library, has 2 potions, and gets past malfoy**

**public** **class** Hermione **implements** Runnable{

loc l;

pot p;

win w;

Hermione(loc l1,pot p1,win w1) {

Thread t = **new** Thread(**this**);

l=l1;

p=p1;

w=w1;

t.start();

}

**public** **void** run() {

**while**(**true**)

{

**if**(l.getState()=="Library")

{

**if**((p.getState())== 2)

{

**if**((w.getState())== **true**){//will only reveal herself if player has 2 potions and defeat malfoy

System.***out***.println("\nHermione: I am here!");

**break**;

}

}

}

}

}

}

**//watch subject**

**import** java.util.concurrent.TimeUnit;

**public** **class** Watch **extends** ConcreteSubject **implements** Runnable{

//time tim;

Thread t;

**int** sec=0;

Watch() {

//this.tim=t1;

t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

//long start = System.currentTimeMillis();

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=1;

notifyObservers(sec);

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=2;

notifyObservers(sec);

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=3;

notifyObservers(sec);

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=4;

notifyObservers(sec);

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=5;

notifyObservers(sec);

**try** {

TimeUnit.***SECONDS***.sleep(10);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

sec=6;

notifyObservers(sec);

}

}

**import** java.util.ArrayList;

**public** **class** ConcreteSubject **implements** Subject{

**private** ArrayList observers;

**public** ConcreteSubject(){

observers = **new** ArrayList();

}

**public** **void** notifyObservers(**int** sec) {

**for** (**int** i =0; i<observers.size(); i++)

{

Observer observer = (Observer)observers.get(i);

observer.update(sec);

}

}

**public** **void** registerObserver(Observer o) {

observers.add(o);

}

**public** **void** removeObsever(Observer o) {

**int** i = observers.indexOf(o);

**if** (i>=0) observers.remove(i);

}

}

**public** **interface** Subject {

**public** **void** registerObserver(Observer o);

**public** **void** removeObsever(Observer o);

**public** **void** notifyObservers(**int** sec);

}

**public** **class** ConcreteObserver **implements** Observer {

**private** Subject subject;

**public** ConcreteObserver(Subject subject)

{

**this**.subject = subject;

subject.registerObserver(**this**);

}

**public** **void** update(**int** sec) {

System.***out***.println( sec +" minute passed...");

}

}

**public** **interface** Observer {

**void** update(**int** sec);

}

//the class created for hermione to check constantly if 2 potions are created

**public** **class** pot {

**int** potion;

pot() {

potion = 0;

}

**public** **synchronized** **int** getState() {

**return** potion;

}

**public** **synchronized** **void** setState(**int** p) {

**this**.potion = p;

}

}

//similarly class created for hermione to check if player won against malfoy

**public** **class** win {

**boolean** win;

win() {

win = **false**;

}

**public** **synchronized** **boolean** getState() {

**return** win;

}

**public** **synchronized** **void** setState(**boolean** w) {

**this**.win = w;

}

}

**//potion class that follows the template pattern**

**public** **abstract** **class** Potion {

**void** Potion()

{

}

**final** **void** preparepotion() {

add();

stir();

heat();

**if**(polyjuice()){

wave();

}

status();

}

**private** **boolean** polyjuice() {

**return** **true**;

}

**protected** **void** heat() {

System.***out***.println("Heating at high temperature…");

}

**protected** **void** status() {

System.***out***.println("Potion is ready!");

}

**abstract** **void** add();

**abstract** **void** stir();

**abstract** **void** wave();

}

//polyjuice

**public** **class** PolyjuicePotion **extends** Potion{

**void** add() {

System.***out***.println("Adding fluxweed and lacewings in cauldron…");

}

**void** stir() {

System.***out***.println("Stirring 4 times clockwise…");

}

**void** wave() {

System.***out***.println("Waving hand to complete brewing…");

}

}

//invisibility

**public** **class** InvisibilityPotion **extends** Potion{

**void** add() {

System.***out***.println("Adding cherries and spiders in cauldron…");

}

**void** stir() {

System.***out***.println("Stirring 3 times anticlockwise…");

}

**void** wave() {

System.***out***.println("Waving hand to complete brewing…");

}

**public** **boolean** polyjuice(){

**return** **false**;

}

}

//class DiningHall

//changed the look function that displays the deatheater and the win function that displays the player winning against enemy message

**import** java.util.Random;

**public** **class** DiningHall **extends** Hogwarts **implements** look,use{

// Spells s = new Spells();

// // create the commands

// Spell1Command s1 = new Spell1Command(s);

// Spell2Command s2 = new Spell2Command(s);

// Spell3Command s3 = new Spell3Command(s);

// Spell4Command s4 = new Spell4Command(s);

// // create the panel

// Command[] cmds = { s1, s2, s3, s4 };

// ControlPanel p = new ControlPanel(cmds);

String[] randomfood= {"Pizza", "Pasta", "Lasagna"};

String[] house = {"Gryffindor!", "Slytherin!", "Ravenclaw!","Hufflepuff!"};

Random rand = **new** Random();

DiningHall()

{

locname = "Dining Hall";

}

**public** **void** look()

{

System.***out***.print("The "+locname+ " is the biggest open area inside Hogwarts. "

+ "Bellatrix Lestrange is here, get ready to fight!\n");

}

**public** **void** win()

{

System.***out***.print("You defeated Bellatrix! The door on the east leads to the Potions Classroom.\n");

}

**public** **void** use(String useinput)

{

**if** (useinput == "sorting hat")

{

System.***out***.println("You walk towards the chair and you sit on it. You are nervous. The sorting hat is \n"+

"placed on your head. He takes his sweet time choosing a house for you..." );

System.***out***.print("Sorting Hat: "+house[rand.nextInt(4)]+"\n");

}

**if** (useinput == "plate")

{

System.***out***.println("You find a huge table filled with all sorts of food. You spot your 3 most favourite food:\n" +

"Pizza, Pasta, and Lasagna. You take your plate and decide to have " + randomfood[rand.nextInt(3)]+"\n");

}

}

// public void talk()

// {

// System.out.println("Dumbledore: Welcome to the dining hall! Interact with the sorting hat to be sorted\n" +

// "into a house or grab a plate and enjoy the food served in the hall!\n");

// }

}

//class Library

// changed the look function that displays the deatheater and the win function that displays the player winning against enemy message. Also win class is passed to this class for hermione

**public** **class** Library **extends** Hogwarts **implements** look,use{

win win1;

Library(win w1)

{

locname="Library";

win1 = w1;

}

**public** **void** look()

{

System.***out***.print("You look around the magnificient "+locname+ " of Hogwarts."

+ " In the centre of the library, you find Draco Malfoy drawing his wand to attack you!\n");

}

**public** **void** win()

{

System.***out***.print("You defeated Malfoy!\n");

win1.setState(**true**);

}

**public** **void** use(String useinput)

{

**if** (useinput == "book")

{

System.***out***.println("You have picked up a book. The title says 'Diary of The Half-Blood Prince'.\n"+

"You open the book and start to read the pages. You don't understand anything. You keep it down.\n");

}

**if** (useinput == "quill")

{

System.***out***.println("You have picked up a quill. You write in a piece of paper the following,"+

" 'Hermione is a nerd...' \n");

}

}

// public void talk()

// {

// System.out.println("You: How do you pronounce Leviosa?");

// System.out.println("Hermione: OMG! It's pronounced Levi'oosa. Not leviosa!");

// System.out.println("You: \*giggles\*");

// System.out.println("Hermione: Instead of wasting my time, read that book or take that quill and complete your essay!");

// }

}

//class PotionsClassroom

// changed the look function that displays the deatheater and the win function that displays the player winning against enemy message. Also has the potions making function calls

**import** java.util.Random;

**public** **class** PotionsClassroom **extends** Hogwarts **implements** look,use{

**boolean**[] chance= {**true**,**false**};

Random rand = **new** Random();

PotionsClassroom()

{

locname= "Potions classroom";

}

**public** **void** look()

{

System.***out***.print("You are in the "+locname+"\n"+

"Potions is one of the core subjects that first year students are required to study. In the\n" +

"potions class there is Professor Snape who teaches the subject. He does not seem very happy\n" +

"to see you and starts attacking you!\n ");

}

**public** **void** win()

{

System.***out***.print("You defeated Snape! Make the potions and go to the door on the west leading to the Library.\n");

}

**public** **void** use(String useinput)

{

**if** (useinput == "polyjuice")

{

PolyjuicePotion p = **new** PolyjuicePotion();

p.preparepotion();

}

**if** (useinput == "invisibility")

{

InvisibilityPotion i = **new** InvisibilityPotion();

i.preparepotion();

}

}

// public void talk()

// {

//

// System.out.println("Professor Snape: You are here to learn the subtle science and exact art of potion-making.\n"

// + "You have two choices: Use a cauldron to understand the beauty of the softly simmering cauldron OR \n"

// + "Use a potion bottle to learn the dangerous procedure of potion making\n");

// }

}

1. **Show sample outputs**
2. Paste screenshots of output showing the player in all locations, fighting all villains and making the 2 potions.

Text, email

Description automatically generated

A picture containing background pattern

Description automatically generated

1. Paste screenshot of output showing that the player found Hermione before time is up and then hiding the map to end the game.

Graphical user interface, text, application, letter, email

Description automatically generated





1. Paste screenshot of output showing that the player got caught (time is up and s/he did not hide the map).

Graphical user interface, text, application, letter, email

Description automatically generated