|  |  |
| --- | --- |
| Course | Advanced Software Design – CS525 |
| Assignment | Lab 6 |
| Week | 06 |
| Due | Mar 23, 2020 |
| Student name | Quan Hong Doan |
| Student ID | 986956 |

Online version:

<https://github.com/zithiat/asd/blob/master/labs%20%26%20assignments/Answers/Assignment_CS525_Week06_986956.docx>

**Problem 1**:

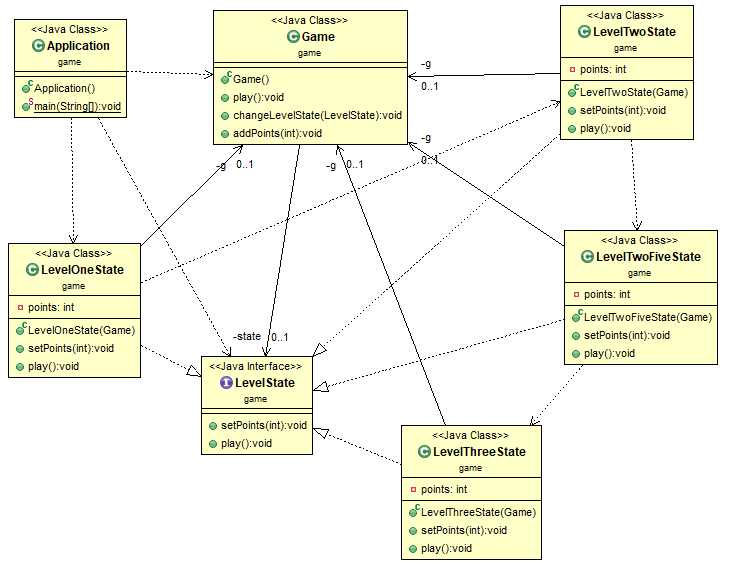
Problem 1 is the question about the **game** problem from the PDF file.

**Answer**:

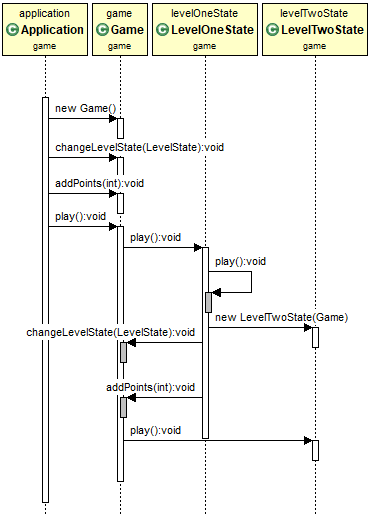
The **Game** source code on my GitHub

<https://github.com/zithiat/asd/tree/master/codes/code%20for%20labs/lab6/game>

**Class diagram**



**Sequence diagram** how to move from level 1 to level 2



Interface **LevelState**

**package** game;

**public** **interface** LevelState {

**void** setPoints(**int** points);

**void** play();

}

**LevelOneState**

**public** **class** LevelOneState **implements** LevelState{

**private** **int** points = 0;

**private** Game g;

**public** LevelOneState(Game g) {

**this**.g = g;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points += points;

}

@Override

**public** **void** play() {

**if** (**this**.points > 10) {

g.changeLevelState(**new** LevelTwoState(g));

g.addPoints(points);

g.play();

}

**else** {

System.***out***.println("Total points = " + points + " --- level = 1");

}

}

}

**LevelTwoState**

**public** **class** LevelTwoState **implements** LevelState {

**private** Game g;

**private** **int** points = 0;

**public** LevelTwoState(Game g) {

**this**.g = g;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points += points;

}

@Override

**public** **void** play() {

**if** (**this**.points > 10 && **this**.points <= 15) {

System.***out***.println("Total points = " + points + " --- level = 2");

} **else** {

g.changeLevelState(**new** LevelTwoFiveState(g));

g.addPoints(points);

g.play();

}

}

}

**LevelTwoFiveState**

**public** **class** LevelTwoFiveState **implements** LevelState {

**private** **int** points = 0;

**private** Game g;

**public** LevelTwoFiveState(Game g) {

**this**.g = g;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points += points;

}

@Override

**public** **void** play() {

**if** (**this**.points > 15 && **this**.points <=20) {

**this**.points += 1;

System.***out***.println("Total points = " + points + " --- level = 2-5");

}

**if** (**this**.points > 20) {

g.changeLevelState(**new** LevelThreeState(g));

g.addPoints(points);

g.play();

}

}

}

**LevelThreeState**

**public** **class** LevelThreeState **implements** LevelState {

**private** **int** points = 0;

**private** Game g;

**public** LevelThreeState(Game g) {

**this**.g = g;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points += points;

}

@Override

**public** **void** play() {

System.***out***.println("Total points = " + points + " --- level = 3");

}

}

Changes in the **Game** class

**public** **class** Game {

**private** LevelState state;

**public** **void** play() {

**this**.state.play();

}

**public** **void** changeLevelState(LevelState state) {

**this**.state = state;

}

**public** **void** addPoints(**int** points) {

**this**.state.setPoints(points);

}

}

**Application** class

**public** **class** Application {

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

Random r = **new** Random();

Game g = **new** Game();

LevelState l1 = **new** LevelOneState(g);

**for** (**int** i = 1; i <= 5; i++) {

g.changeLevelState(l1);

g.addPoints(r.nextInt(10));

g.play();

}

}

}

**Problem 2**:

Problem 2 is the question about the **counter** problem from the PDF file.

**Answer**:

The **Counter** source on my GitHub

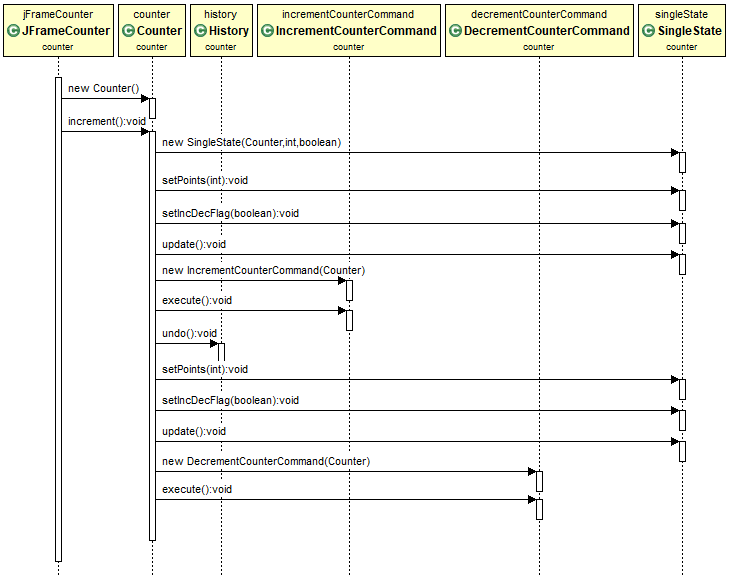
<https://github.com/zithiat/asd/tree/master/codes/code%20for%20labs/lab6/counter>

**Class diagram**

A close up of a map

Description automatically generated

**Sequence diagram** for click increment and undo button



**State** interface

**public** **interface** State {

**void** update();

**void** setIncDecFlag(**boolean** f);

**void** setPoints(**int** points);

}

**SingleState**

**package** counter;

**public** **class** SingleState **implements** State {

Counter c;

**private** **int** points = 0;

**private** **boolean** incDecFlag = **true**;

**public** SingleState(Counter c, **int** points, **boolean** f) {

**this**.c = c;

}

@Override

**public** **void** update() {

**if** (incDecFlag) {

**this**.points++;

} **else** {

**this**.points--;

}

System.***out***.println("SingleState updated points: " + **this**.points);

**if** (**this**.points <= 0) {

**this**.points = 0;

} **else** **if** (**this**.points >= 10) {

c.changeState(**new** DoubleState(c, **this**.points, incDecFlag));

}

c.setCount(**this**.points);

}

@Override

**public** **void** setIncDecFlag(**boolean** f) {

**this**.incDecFlag = f;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

}

**DoubleState**

**public** **class** DoubleState **implements** State {

Counter c;

**private** **int** points = 0;

**private** **boolean** incDecFlag = **true**;

**public** DoubleState(Counter c, **int** points, **boolean** f) {

**this**.c = c;

}

@Override

**public** **void** update() {

**if** (incDecFlag) {

**this**.points += 2;

} **else** {

**this**.points -= 2;

}

System.***out***.println("DoubleState Received points: " + **this**.points);

**if** (**this**.points < 10) {

c.changeState(**new** SingleState(c, **this**.points, incDecFlag));

} **else** **if** (**this**.points >= 100) {

c.changeState(**new** TripleState(c, **this**.points));

}

c.setCount(**this**.points);

}

@Override

**public** **void** setIncDecFlag(**boolean** f) {

**this**.incDecFlag = f;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

}

**TripleState**

**public** **class** TripleState **implements** State {

Counter c;

**private** **int** points = 0;

**private** **boolean** incDecFlag = **true**;

**public** TripleState(Counter c, **int** points) {

**this**.c = c;

}

@Override

**public** **void** update() {

**if** (incDecFlag) {

**this**.points += 3;

} **else** {

**this**.points -= 3;

}

System.***out***.println("TripleState Received points: " + **this**.points);

**if** (**this**.points < 100) {

c.changeState(**new** DoubleState(c, **this**.points, incDecFlag));

}

c.setCount(**this**.points);

}

@Override

**public** **void** setIncDecFlag(**boolean** f) {

**this**.incDecFlag = f;

}

@Override

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

}

Changes in the **Counter** class

State state;

**public** Counter() {

observerList = **new** ArrayList<Observer>();

**this**.state = **new** SingleState(**this**, count, **true**);

}

**…**

**public** **void** increment() {

count++;

**this**.state.setIncDecFlag(**true**);

**this**.state.setPoints(count);

**this**.state.update();

**this**.observerList.stream().forEach(e -> e.update(count));

}

**public** **void** decrement() {

count--;

**this**.state.setIncDecFlag(**false**);

**this**.state.setPoints(count);

**this**.state.update();

**this**.observerList.stream().forEach(e -> e.update(count));

}

**public** **void** changeState(State s) {

**this**.state = s;

}

**public** **void** setCount(**int** count) {

**this**.count = count;

}