

**NCC Level-4 Diploma in Computing**  
**Designing and Developing Object-Oriented**  
**Computer Program**

**Candidates Name : Fatema Akter**

**ID No : 00154713**

**Module Title : Designing and Developing Object-Oriented  
Computer Program**

**Assignment Title : QuickClicker**

**Examination Cycle: December 2015**

**Candidate attempting to gain an unfair advantage or colluding in anyway whatsoever (other than on joint assignments) are liable to be disqualified. Plagiarism is an offence.**

**Expected candidate time allocation: 35 to 40 hours**

Mark	Moderated	Final
	Mark	Mark

**Marker's comment:**

**Moderator's comment:**



## **Statement of Confirmation of Own Work**

### **Programmed /qualification name: Designing and Developing Object-Oriented Computer Program**

#### **Student Declaration:**

I have read and understood the NCC Education's policy on Academic Dishonesty and Plagiarism.

I can confirm the following details:

**Student ID/Registration number : 00154713**

**Name : Fatema Akter**

**Center Name : Daffodil Institute of Information Technology.**

**Module Name : Designing and Developing Object-Oriented  
Computer Program**

**Assignment Title : QuickClicker**

**Number of Words : 1,348**

I confirm that this is my own and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments.

**Due Date: 25/10/2015**

**Submitted Date: 24/10/2015**

**Student Signature: Fatema Akter**

## ACKNOWLEDGEMENT

At the beginning I would like to render thanks to the almighty Allah. And so I would wish to show my special thanks, gratitude to my teacher Tasnuva Ali well as all other teachers. Thanks to NCC education, who afforded me this tremendous task. I did a great deal of research and I came to know about so many recalls and it helped to increase my knowledge.

Once more, I would wish to give thanks all of them who helped me to complete this project.

## Table of Contents

Task-1 .....	6
The Application: .....	6
(1) Appropriate use of objects: .....	6
List of the class: .....	6
(2) Handling user interaction: .....	7
(3) Upgrades: .....	14
(4) Encapsulation and Abstraction: .....	15
Testing Data: .....	29
(1) Developing a test plan: .....	29
Testing Data .....	29
(2) Implementing test plan: .....	29
White Box Testing .....	29
Table for White Box Testing: .....	33
Black Box Testing.....	34
Table for Black Box Testing .....	36
Boundary Testing .....	37
Table for Boundary Testing.....	38
Exception Handling .....	39
Short discussion of Data Testing: .....	39
Task-3 .....	40
Design Documentation: .....	40
Class Relationships with Methods Attributes: .....	40
Conclusion: .....	41
Bibliography.....	42

**Introduction:**

In this assignment I am asked to create a incremental game. The game is performed by a player's simple actions to achieve points. These points can be upgrade to get point in faster rate. The development will take the form of one button each click will award one point to the player. This will also show how many points the player has earned. There will be a shop for the players to buy upgrades in cost. There will two types of upgrades.

.

## Task-1

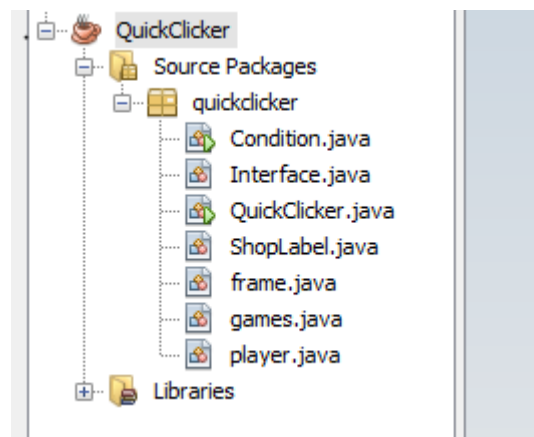
### The Application:

#### (1) Appropriate use of objects:

#### List of the class:

Here is all the class list that I have created for this game.

- Condition
- Interface
- QuickClicker
- ShopLable
- Frame
- Games
- Player



**Figure No: 1.1- interface of class list**

## (2) Handling user interaction:

- Here interface of when this program is open

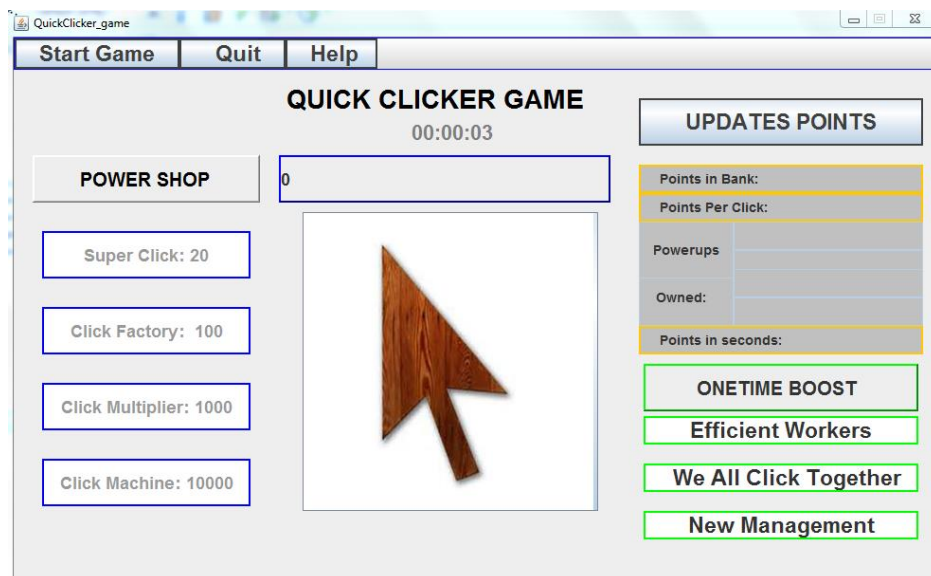


Figure No: 1.2- interface of when the program is open

- Here is screenshot that show game interface with there all system

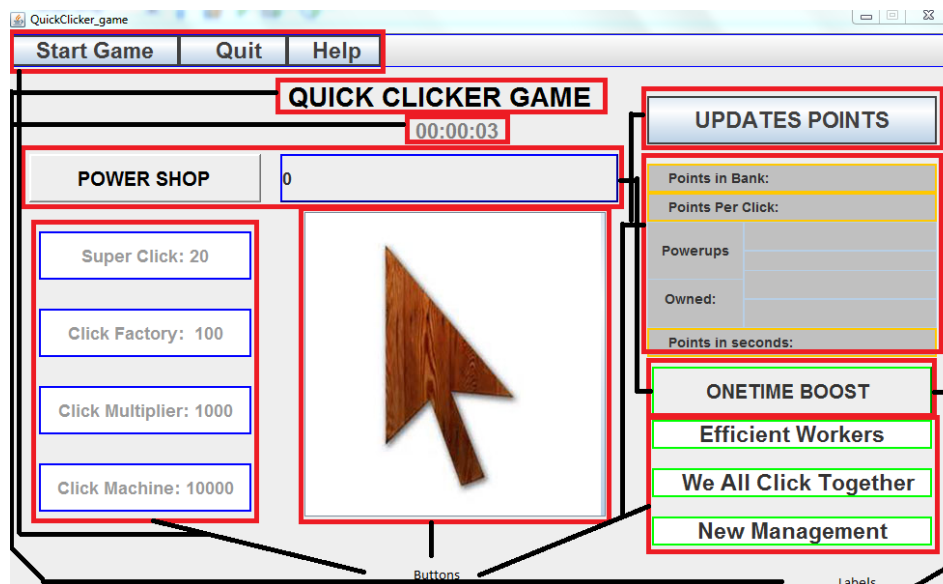


Figure No: 1.3- interface of game with there all system

- This interface for when the player will be start game.

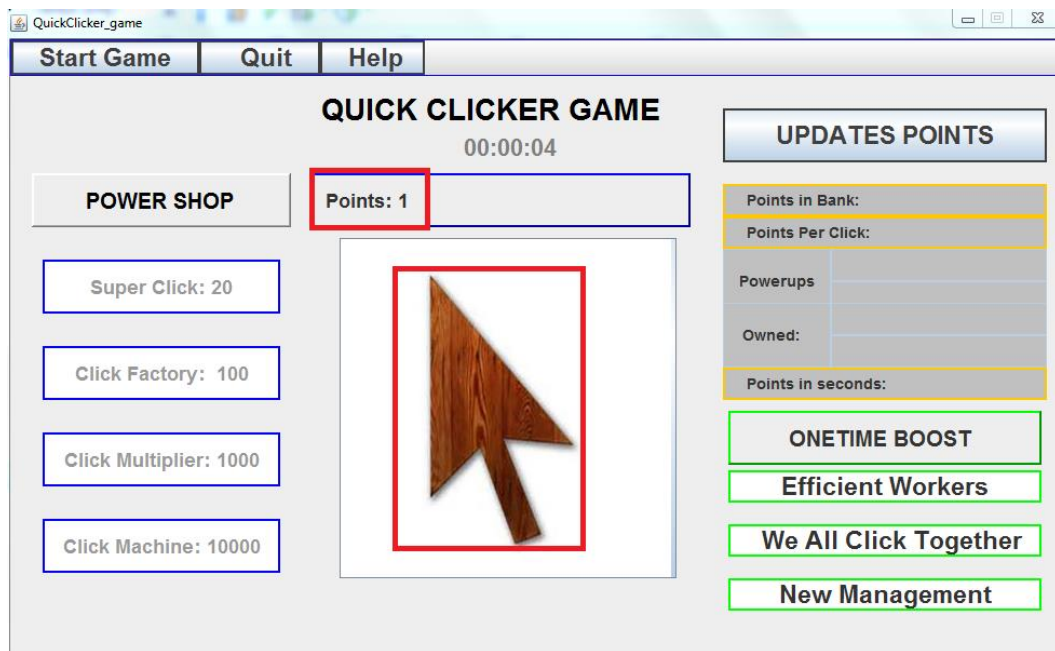


Figure No: 1.4- interface of Start game

- This interface is for buy points. When the player has gained 20 points then they can buy super click.

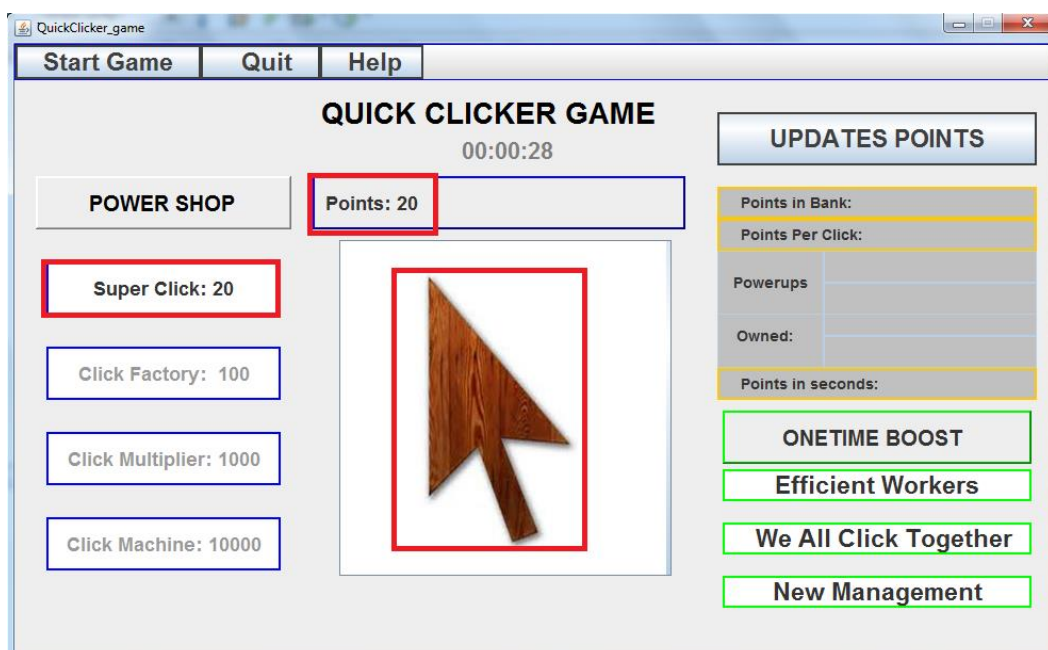


Figure No: 1.5- interface of Buy super click



- After this interface is show with updates points and the super click increase points



Figure No: 1.6- interface of updates super click point

- Same way the player has gained 100 Points then they can buy click Factory

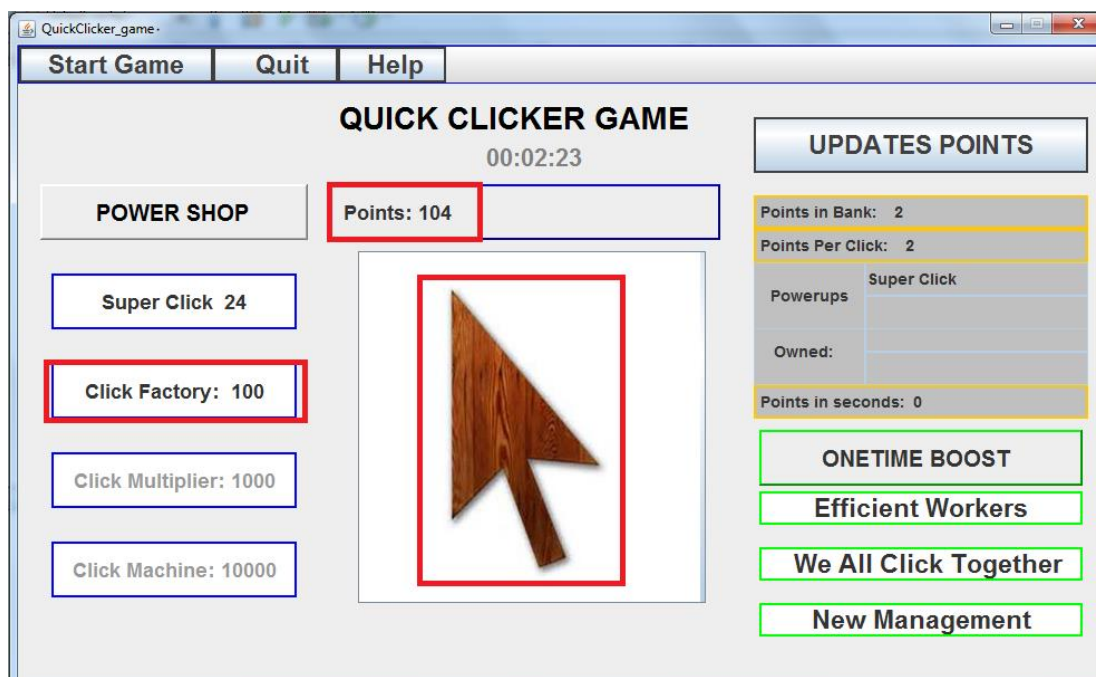


Figure No: 1.7- interface of Buy click factory

- After this interface is show with updates points and the click factory increase point

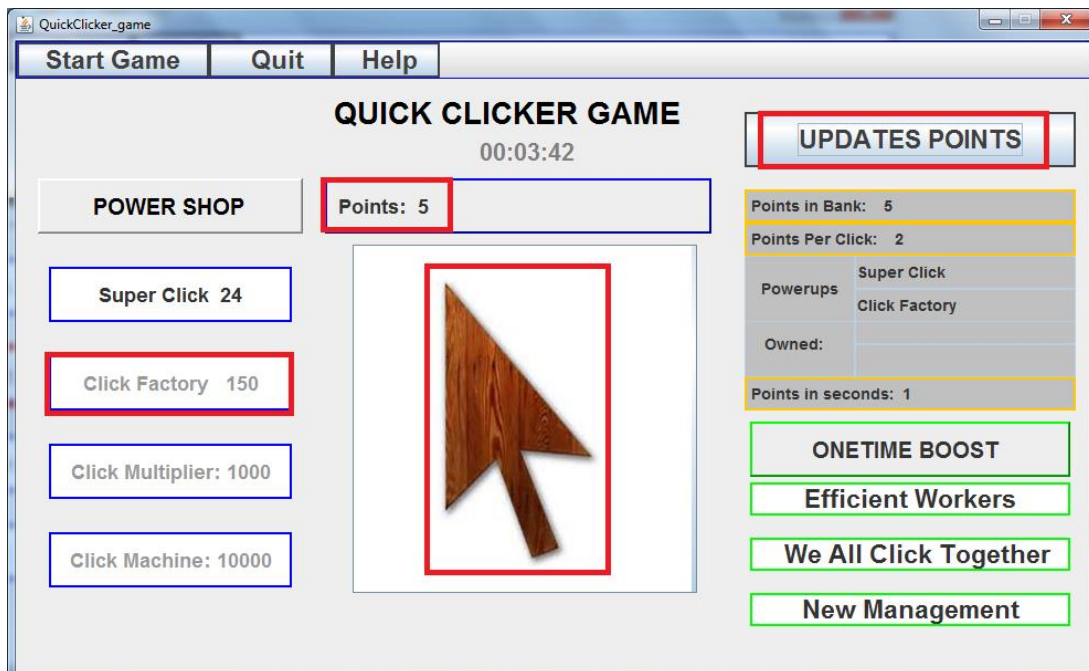


Figure No: 1.8- interface of updates click factory points

- Again the player has gained 1000 Points then they can buy click Multiplier

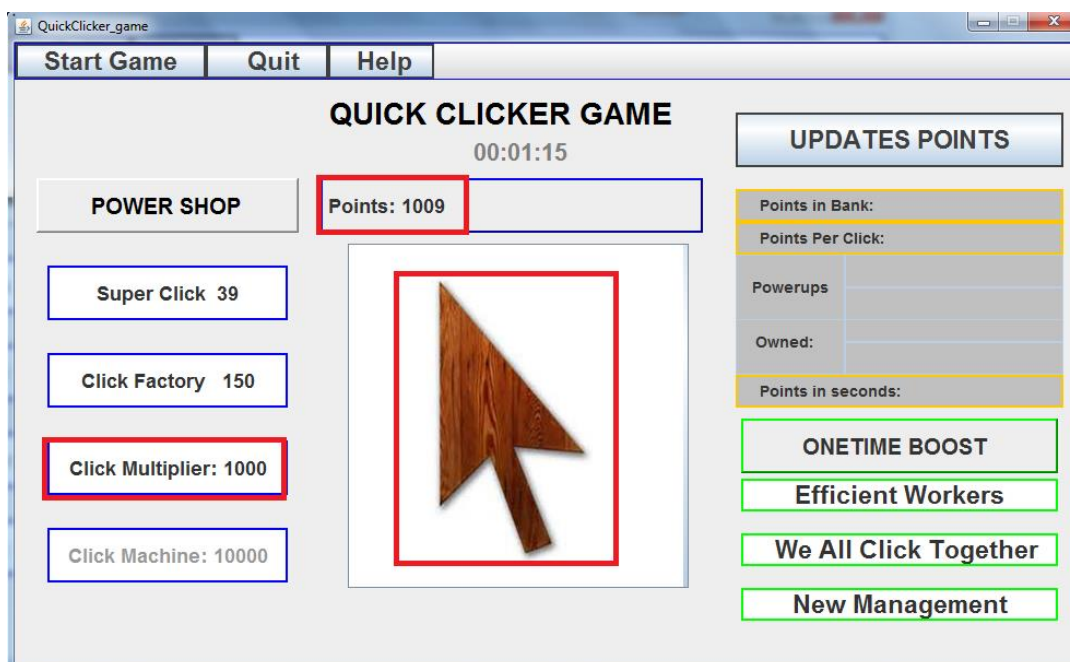


Figure No: 1.9- interface of Buy click Multiplier

- After this interface is show with updates points and the click Multiplier increase points

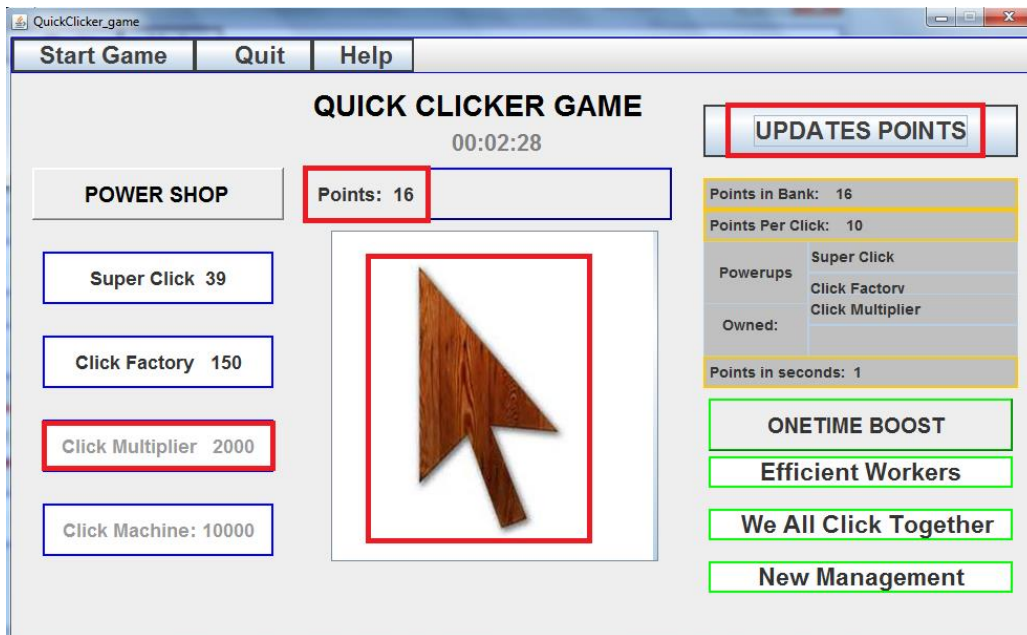


Figure No: 1.10- interface of updates click multiplier points

- Same way the player has gained 10000 Points then they can buy click Machine

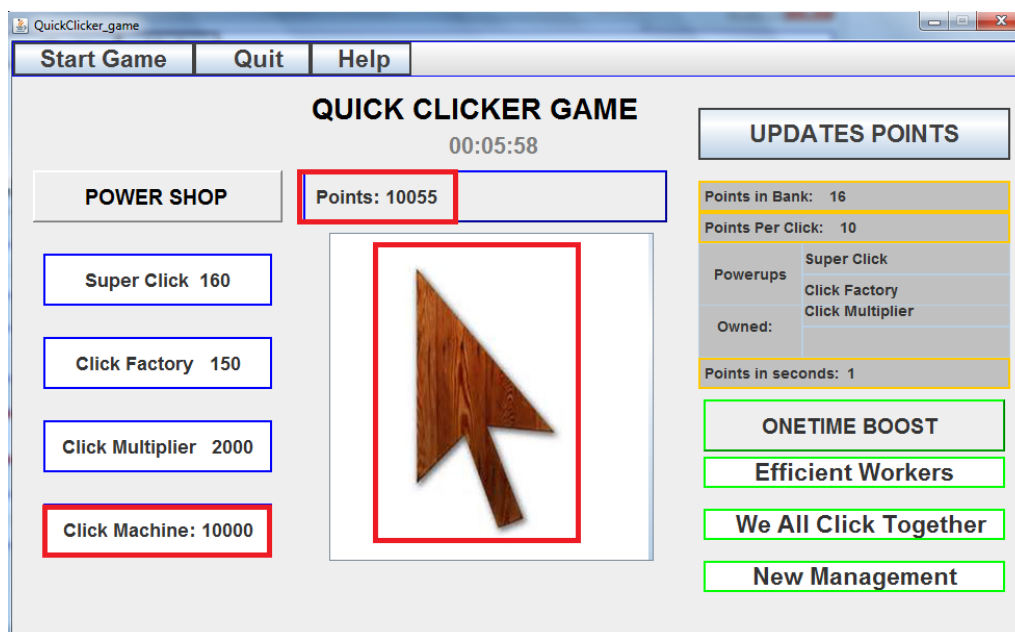


Figure No: 1.11- interface of buy click machine

- After this interface is show with updates points and the click Machine increase points

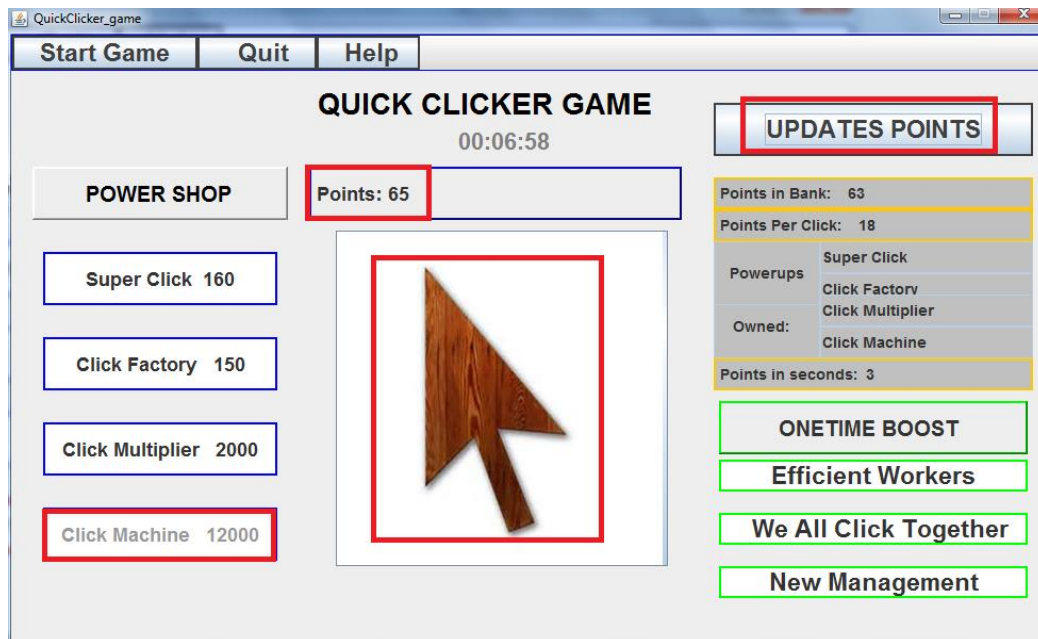


Figure No: 1.12- interface of updates Click Machine points

- This interface is show for game start newly. if the player click on “ Start Game ” then the program will be open newly.

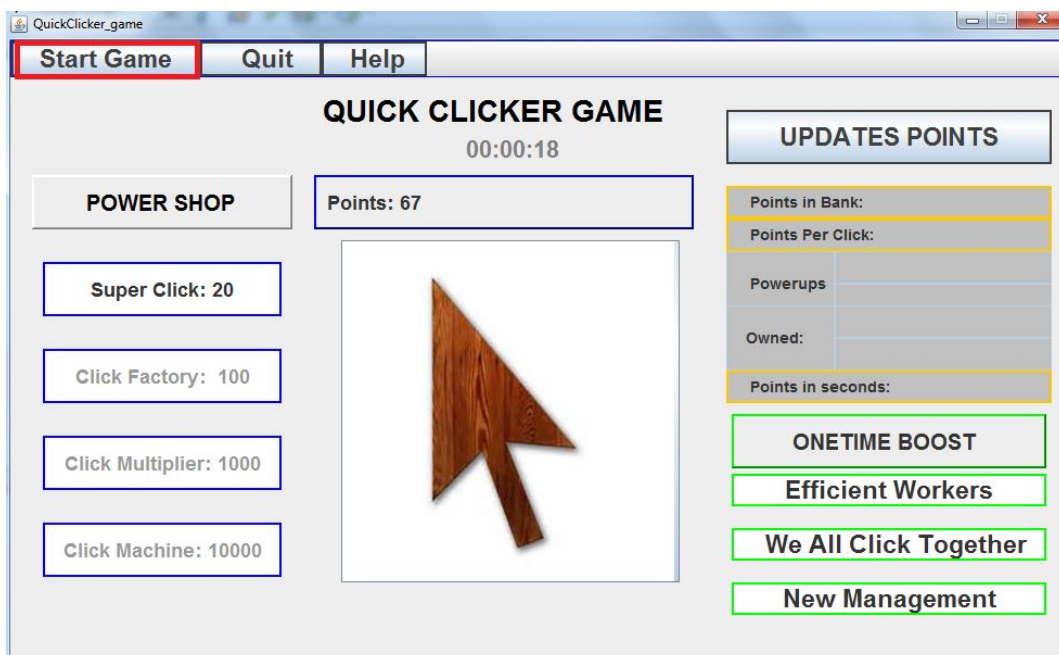


Figure No: 1.13- interface of start game newly

- This interface for Exit program .if the player wants to Exit program then they click on Quit button and then come out.

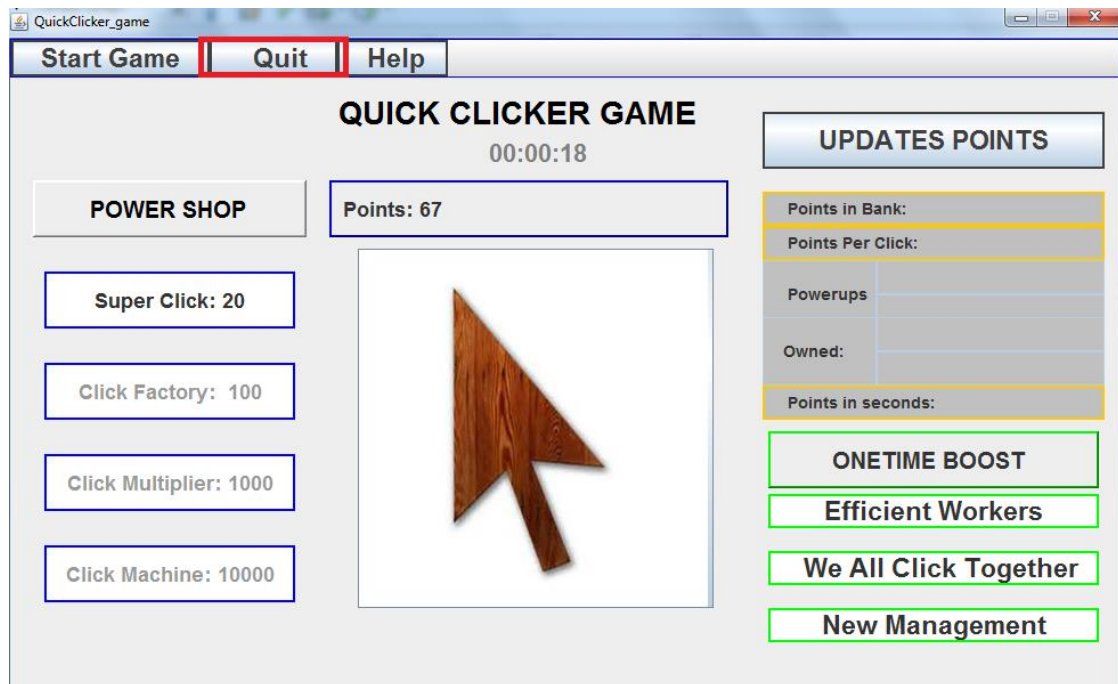


Figure No: 1.14- interface of Quit for come out game

### (3) Upgrades:

- This interface is for update button
- the player click on this button for updates points
- These interfaces are show on all labels. All labels are show on all updates points

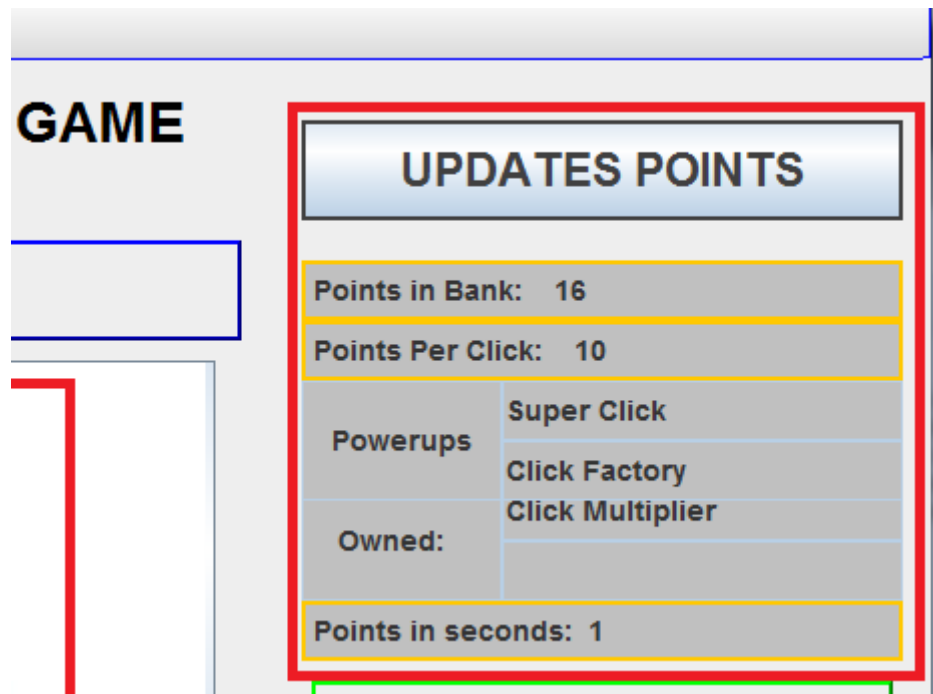


Figure No: 1.15- interface of Updates points

(Anon., n.d.)

#### (4) Encapsulation and Abstraction:

Code for Condition class:

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package quickclicker;
6
7
8  import java.awt.event.ActionEvent;
9  import java.awt.event.ActionListener;
10
11
12  /**
13   *
14   * @author Shawn
15   */
16  public class Condition extends Interface implements ActionListener {
17
18      public static void main(String[] args) {
19
20          // Interface obj= new Interface();
21          //
22          // obj=Interface();
23          //
24
25      }
26
27      private static Interface Interface() {
28          throw new UnsupportedOperationException("Not yet implemented");
29      }
30
31
32  }
33
34  @Override
35  public void actionPerformed(ActionEvent e) {
36      throw new UnsupportedOperationException("Not supported yet.");
37  }
38  }
```

Figure No: 1.16- interface for condition class coding



Code for Interface class:

```

1
2 package quickclicker;
3 import java.awt.*;
4 import java.awt.event.*;
5 import java.sql.Time;
6
7 import javax.swing.*;
8 import javax.swing.border.*;
9
10 import javax.swing.Timer;
11
12
13
14 public class Interface extends JFrame implements ActionListener {
15
16
17     Border bevelBorder = BorderFactory.createRaisedBevelBorder();
18     Font font = new Font("Arial",Font.BOLD, 23);
19
20     int point=0;
21     int point2=0;
22
23
24     String ownsuper="";
25     String ownfactory="";
26     String ownmulti="";
27     String ownmachine="";
28
29
30
31     int increase=1;
32     int countsuperclick=20;
33     int countClickFactory=100;
34     int countClickMultiplier=1000;
35     int countClickMachine=10000;
36
37
38     int superincrease=1;
39     int facincrease=1;
40     int multiincrease=2;
41     int macincrease=2;
42     int timepointincrease1;
43     int timepointincrease2;
44
45     JButton newgame = new JButton ("    Start Game    ");
46     JButton exitbtn = new JButton ("        Quit        ");
47     JButton helpbtn = new JButton ("    Help    ");
48

```



```

49     private JTextField update3,update31;
50     private JTextField update1,update2, update4, supupdate, facupdate, macupdate,multiupdate;;
51     private JButton clickhere;
52     private JButton btnClickFactory;
53     private JButton btnClickMultiplier;
54     private JButton btnclickmachine;
55     private JButton btnupdate;
56     private JButton btnWorkers;
57     private JButton btnClicktogether;
58     private JButton btnmanagement;
59     private JLabel lblcookies;
60     //private JLabel lblgamenam;
61     // private JLabel lblupgrade ;
62     // private JLabel onetimeboost;
63     private JButton btnsuperclick;
64     private JLabel settimer ;
65     private JLabel labeltimer ;
66     private Timer timer;
67     private int time;
68
69
70     public Interface    (){
71
72         games();
73         this.setResizable(false);
74         this.setSize(980, 600);
75         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
76         this.setTitle("QuickClicker_game");
77         this.setLayout(null);
78         this.setVisible( true);
79         this.setLocationRelativeTo(null);
80
81
82         clickhere = new JButton(" Points");
83         //clickhere.setBackground(Color.red);
84         clickhere.setBounds(305,150,312,315);
85
86         this.add(clickhere);
87
88         clickhere.setIcon(new ImageIcon("Cake.jpg"));
89         clickhere.addActionListener(this);
90
91
92         lblcookies = new JLabel(" 0");
93         this.add(lblcookies);
94         lblcookies.setBounds(280,90,350,50);
95         lblcookies.setBackground(Color.red);
96         lblcookies.setFont(new Font("Tekton Pro Ext", Font.BOLD, 18));
97         lblcookies.setBorder( bevelBorder );
98         // lblcookies.setForeground(Color.black);
99         lblcookies.setBackground(Color.blue );
100
101         lblcookies.setText(point+"");
102
103         btnsuperclick = new JButton("Super Click: "+""+countsuperclick);
104         this.add(btnsuperclick);
105         btnsuperclick.setBounds(30, 170, 220, 50);
106         btnsuperclick.setBorder(BorderFactory.createLineBorder(Color.blue,2));
107         btnsuperclick.setFont(new Font("Tekton Pro Ext", Font.BOLD, 18));
108         btnsuperclick.setBackground(Color.white);
109         btnsuperclick.setEnabled(false);
110         btnsuperclick.addActionListener(this);

```

```

111 btnClickFactory = new JButton("Click Factory: 100");
112 this.add(btnClickFactory);
113 btnClickFactory.setBounds(30, 250, 220, 50);
114 btnClickFactory.setBorder(BorderFactory.createLineBorder(Color.blue,2));
115 btnClickFactory.setFont(new Font("Tekton Pro Ext", Font.BOLD, 18));
116 btnClickFactory.setBackground(Color.white);
117 btnClickFactory.setEnabled(false);
118 btnClickFactory.addActionListener(this);
119
120
121 btnClickFactory.addActionListener(new ActionListener() {
122     Timer timer1 = new Timer(10000, this);
123     @Override
124     public void actionPerformed(ActionEvent e) {
125
126
127
128         timer1.start();
129
130         point=point+facincrease;
131
132         lblcookies.setText(" Points: "+point);
133         //timer1.stop();
134
135     }
136
137 });
138
139 btnClickMultiplier = new JButton("Click Multiplier: 1000");
140 this.add(btnClickMultiplier);
141 btnClickMultiplier.setBounds(30, 330, 220, 50);
142 btnClickMultiplier.setBorder(BorderFactory.createLineBorder(Color.blue,2));
143 btnClickMultiplier.setFont(new Font("Tekton Pro Ext", Font.BOLD, 18));
144 btnClickMultiplier.setBackground(Color.white);
145 btnClickMultiplier.setEnabled(false);
146 btnClickMultiplier.addActionListener(this);
147
148
149
150 btnClickMachine = new JButton("Click Machine: 10000");
151 this.add(btnClickMachine);
152 btnClickMachine.setBounds(30, 410, 220, 50);
153 btnClickMachine.setBorder(BorderFactory.createLineBorder(Color.blue,2));
154 btnClickMachine.setFont(new Font("Tekton Pro Ext", Font.BOLD, 18));
155 btnClickMachine.setBackground(Color.white);
156 btnClickMachine.setEnabled(false);
157 btnClickMachine.addActionListener(this);
158 btnClickMachine.addActionListener(new ActionListener() {
159     Timer timer2 = new Timer(5000, this);
160     @Override
161     public void actionPerformed(ActionEvent e) {
162
163
164
165         timer2.start();
166
167         point=point+macincrease;
168
169         lblcookies.setText(" Points: "+point);
170     }

```

```

169         });
170
171
172
173         JMenuBar menubar= new JMenuBar();
174
175         setJMenuBar(menubar);
176         menubar.add(newgame);
177         menubar.add(exitbtn);
178         menubar.add(helpbtn);
179
180
181         menubar.setBorder(BorderFactory.createLineBorder(Color.blue,1));
182         newgame.setBorder(BorderFactory.createLineBorder(Color.darkGray,2));
183         newgame.setFont(font);
184         newgame.setForeground(Color.darkGray);
185         newgame.addActionListener(this);
186
187
188         exitbtn.setBorder(BorderFactory.createLineBorder(Color.darkGray,2));
189         exitbtn.setFont(font);
190         exitbtn.setForeground(Color.darkGray);
191         exitbtn.addActionListener(this);
192
193
194         helpbtn.setBorder(BorderFactory.createLineBorder(Color.darkGray,2));
195         helpbtn.setFont(font);
196         helpbtn.setForeground(Color.darkGray);
197         helpbtn.addActionListener(this);
198
199         btnupdate = new JButton("UPDATES POINTS");
200         this.add(btnupdate);
201         btnupdate.setBounds(660,30,300,50);
202         btnupdate.setBorder(BorderFactory.createLineBorder(Color.darkGray,2));
203         btnupdate.setFont(font);
204         btnupdate.setForeground(Color.darkGray);
205         btnupdate.addActionListener(this);
206
207
208         btnWorkers = new JButton("Efficient Workers");
209         this.add(btnWorkers);
210         btnWorkers.setBounds(665, 365, 290, 30);
211         btnWorkers.setBorder(BorderFactory.createLineBorder(Color.green,2));
212         btnWorkers.setFont(font);
213         btnWorkers.setBackground(Color.white);
214         btnWorkers.addActionListener(this);
215
216         btnClicktogether = new JButton(" We All Click Together");
217         this.add(btnClicktogether);
218         btnClicktogether.setBounds(665, 415, 290, 30);
219         btnClicktogether.setBorder(BorderFactory.createLineBorder(Color.green,2));
220         btnClicktogether.setFont(font);
221         btnClicktogether.setBackground(Color.white);
222         btnClicktogether.addActionListener(this);
223
224         btnmanagement = new JButton("New Management");
225         this.add(btnmanagement);
226         btnmanagement.setBounds(665, 465, 290, 30);
227         btnmanagement.setBorder(BorderFactory.createLineBorder(Color.green,2));
228         btnmanagement.setFont(font);
229         btnmanagement.setBackground(Color.white);

```

```
btnmanagement.addActionListener(this);

update1 = new JTextField();
update2 = new JTextField();
update3 = new JTextField();
update31 = new JTextField();
update4 = new JTextField();

update1.setBorder(BorderFactory.createLineBorder(Color.orange,2));
update2.setBorder(BorderFactory.createLineBorder(Color.orange,2));
update4.setBorder(BorderFactory.createLineBorder(Color.orange,2));

suprupdate = new JTextField();
facupdate = new JTextField();
macupdate = new JTextField();
multiupdate = new JTextField();

this.add(update1);
this.add(update2);
this.add(update3);
this.add(update31);
this.add(update4);
this.add(suprupdate);
this.add(facupdate);
this.add(macupdate);
this.add(multiupdate);

update1.setEditable(false);
update2.setEditable(false);
update3.setEditable(false);
update31.setEditable(false);
update4.setEditable(false);
suprupdate.setEditable(false);
facupdate.setEditable(false);
macupdate.setEditable(false);
multiupdate.setEditable(false);

// suprupdate, facupdate, macupdate,multiupdate;

update1.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
update2.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
update3.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
update31.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
update4.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
suprupdate.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
facupdate.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
macupdate.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
multiupdate.setFont(new Font("Tekton Pro Ext", Font.BOLD, 14));
```

```

280
281     update1.setBounds(660, 100, 300, 30);
282     update2.setBounds(660, 130, 300, 30);
283     update3.setBounds(660, 160, 100, 60);
284     update31.setBounds(660, 210, 100, 60);
285     update4.setBounds(660, 270, 300, 30);
286     supupdate.setBounds(760, 160, 200, 30);
287     facupdate.setBounds(760, 190, 200, 30);
288     macupdate.setBounds(760, 210, 200, 30);
289     multiupdate.setBounds(760, 240, 200, 30);
290
291     update1.setBackground(Color.LIGHT_GRAY);
292     update2.setBackground(Color.LIGHT_GRAY);
293     update3.setBackground(Color.LIGHT_GRAY);
294     update31.setBackground(Color.LIGHT_GRAY);
295     update4.setBackground(Color.LIGHT_GRAY);
296     supupdate.setBackground(Color.LIGHT_GRAY);
297     facupdate.setBackground(Color.LIGHT_GRAY);
298     macupdate.setBackground(Color.LIGHT_GRAY);
299     multiupdate.setBackground(Color.LIGHT_GRAY);
300
301     update1.setText("    Points in Bank: ");
302     update2.setText("    Points Per Click: ");
303     update3.setText("    Powerups    ");
304     update31.setText("    Owned:    ");
305     update4.setText("    Points in seconds: ");
306     //labeltimer = new JLabel("    Time: ");
307     //this.add(labeltimer);
308     //labeltimer.setBounds(300,50,150,40);
309     //labeltimer.setFont(new Font("Time", Font.BOLD, 18));
310     //labeltimer.setForeground(Color.gray);
311
312     settimer = new JLabel(" ");
313     this.add(settimer);
314     settimer.setBounds(420,45,100,40);
315     settimer.setFont(new Font("Time", Font.BOLD, 22));
316     settimer.setForeground(Color.gray);
317
318     time=0;
319
320     timer=new Timer (1000, new ActionListener(){
321
322
323     public void actionPerformed(ActionEvent arg0){
324
325         settimer.setText(new Time(00,00,time++)+"");
326
327     }
328
329     });
330     timer.start();
331
332 }
333
334

```

```
.
336  @Override
337  public void actionPerformed(ActionEvent e) {
338
339
340      if(e.getSource() == clickhere){
341
342
343          point=point+increase;
344          lblcookies.setText(" Points: "+point);
345
346
347          if (point >=countsuperclick){
348              btnsuperclick.setEnabled(true);
349
350          }
351
352          else{
353              btnsuperclick.setEnabled(false);
354
355          }
356
357          if (point >=countClickFactory){
358              btnClickFactory.setEnabled( true);
359          }
360
361          else{
362              btnClickFactory.setEnabled(false);
363          }
364
365          if (point >=countClickMultiplier){
366              btnClickMultiplier.setEnabled( true);
367          }
368          else{
369              btnClickMultiplier.setEnabled(false);
370          }
371
372          if (point >=countclickmachine){
373              btnclickmachine.setEnabled( true);
374          }
375
376          else{
377              btnclickmachine.setEnabled(false);
378          }
379
380      }
381
382      if(e.getSource() == btnWorkers){
383
384          superincrease=superincrease+1;
385          multiincrease=multiincrease+1;
386          facincrease=facincrease+1;
387          macincrease=macincrease+1;
388          btnWorkers.setEnabled(false);
389      }
```



```

390
391
392         if(e.getSource() == btnClicktogether){
393
394             facincrease=facincrease+1;
395             btnClicktogether.setEnabled(false);
396             lblcookies.setText(" Points: "+point);
397         }
398
399
400     if(e.getSource() == btnsuperclick){
401
402         if (point >=countsuperclick){
403             btnClickFactory.setEnabled(false);
404             increase=increase+superincrease;
405             ownsuper="Super Click";
406             point-=countsuperclick ;
407             countsuperclick*=1.2;
408         }
409         else{
410             btnsuperclick.setEnabled(false);
411         }
412         lblcookies.setText(" Points: "+point);
413         btnsuperclick.setText("Super Click"+" "+countsuperclick);
414
415     }
416
417
418
419     if(e.getSource() == btnClickFactory){
420         btnClickFactory.setEnabled(false);
421
422         //clickfac=counter;
423         if (point >=countClickFactory){
424             ownfactory="Click Factory";
425             timepointincrease1=facincrease;
426             point-= countClickFactory ;
427         }
428         else{
429             btnClickFactory.setEnabled(false);
430         }
431         lblcookies.setText(" Points: "+point);
432         countClickFactory*=1.5;
433         btnClickFactory.setText("Click Factory "+countClickFactory);
434     }
435
436
437
438     if(e.getSource() == btnClickMultiplier){
439
440         btnClickMultiplier.setEnabled(false);
441         if (point >=countClickMultiplier){
442             ownmulti="Click Multiplier";
443
444             increase*=multiincrease;
445             point-= countClickMultiplier;
446         }

```

```

447         else{
448             btnClickMultiplier.setEnabled(false);
449         }
450         lblcookies.setText(" Points: "+point);
451         countClickMultiplier*=2;
452         btnClickMultiplier.setText("Click Multiplier "+countClickMultiplier);
453     }
454
455
456
457     if(e.getSource() == btnclickmachine){
458
459         btnclickmachine.setEnabled(false);
460
461         if (point >=countclickmachine){
462             point-= countclickmachine ;
463             ownmachine="Click Machine";
464             timepointincrease2=macincrease;
465         }
466         else{
467             btnclickmachine.setEnabled(false);
468         }
469         lblcookies.setText(" Points: "+point);
470         countclickmachine*=1.2;
471         btnclickmachine.setText("Click Machine "+countclickmachine);
472     }
473
474
475
476
477

```

```

478         if(e.getSource() == btnupdate){
479
480             update1.setText(" Points in Bank: " +point);
481             update2.setText(" Points Per Click: " +increase);
482
483             supupdate.setText(ownsuper);
484             facupdate.setText(ownfactory);
485             macupdate.setText(ownmulti);
486             multiupdate.setText(ownmachine);
487             update4.setText(" Points in seconds: " +(timepointincrease1+timepointincrease2));
488
489
490         }
491
492
493         if(e.getSource() == newgame){
494
495             this.dispose();
496             final frame framel =new frame();
497         }
498
499         if(e.getSource() == exitbtn){
500             this.dispose();
501         }
502
503         if(e.getSource() == btnmanagement){
504             point=(point+ (point*10)/100);
505             btnmanagement.setEnabled(false);
506             lblcookies.setText(" Points: "+point);
507         }
508
509

```



```

510 L                                     }
511
512 private void games() {
513     games a= new games();
514     this.add(a.lblgamenam);
515     player b= new player();
516     this.add(b.onetimeboost);
517
518     ShopLabel c= new ShopLabel();
519     this.add(c.blupgrade);
520 }
521
522
523
524
525 }
```

Figure No: 1.17- interface for interface class coding

Code for QuickClicker class:

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package quickclicker;
6
7  public class QuickClicker {
8
9      public static void main(String[] args) {
10         new frame();
11         // Interface o= new Interface();
12     }
13 }
14
```

Figure No: 1.18- interface for Quickclicker class coding

Cade for ShopLable class:

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package quickclicker;
6
7  import java.awt.Color;
8  import java.awt.Font;
9  import javax.swing.BorderFactory;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.border.Border;
13
14 /**
15 *
16 * @author Fatema
17 */
18 public class ShopLabel extends JFrame {
19     JLabel lblupgrade;
20     Border bevlBorder = BorderFactory.createRaisedBevelBorder();
21     public ShopLabel() {
22         lblupgrade = new JLabel("        POWER SHOP");
23         this.add(lblupgrade);
24         lblupgrade.setFont(new Font("Tekton Pro Ext", Font.BOLD, 20));
25         // upgraLB1.setForeground(Color.white);
26         lblupgrade.setBounds(20, 90, 240, 50);
27         lblupgrade.setForeground(Color.black);
28         lblupgrade.setBorder( bevlBorder );
29     }
30 }

```

Figure No: 1.19- interface for ShopLable class coding

Code for Frame class

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package quickclicker;
6
7  import javax.swing.*;
8
9  /**
10 *
11 * @author Shawn
12 */
13 public class frame extends JFrame {
14
15     public frame () {
16
17         new Interface();
18     }
19
20 }
21
22

```

Figure No: 1.20- interface for Frame class coding

Code for Games class:

```
1  /**
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5   package quickclicker;
6
7   import java.awt.*;
8   import java.awt.Font;
9   import javax.swing.*;
10
11  /**
12   *
13   * @author Fatema
14   */
15  public class games extends JFrame {
16      public JLabel lblgamenam;
17      public games() {
18          lblgamenam = new JLabel(" QUICK CLICKER GAME ");
19          this.add(lblgamenam);
20          lblgamenam.setBounds(280,5,400,50);
21          lblgamenam.setFont(new Font("Tekton Pro Ext", Font.BOLD, 28));
22          lblgamenam.setForeground(Color.black);
23      }
24  }
25
```

Figure No: 1.21- interface for game class coding

Code for Player class:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package quickclicker;
6
7  import java.awt.Color;
8  import java.awt.Font;
9  import javax.swing.*;
10 import javax.swing.border.Border;
11
12
13 public class player extends JFrame {
14     public JLabel onetimeboost;
15     Border bevlBorder = BorderFactory.createRaisedBevelBorder();
16
17
18     public player(){
19         try{
20             onetimeboost = new JLabel("          ONETIME BOOST");
21             this.add(onetimeboost);
22             onetimeboost.setFont(new Font("Tekton Pro Ext", Font.BOLD, 20));
23             //upgraLB1.setForeground(Color.white);
24             onetimeboost.setBounds(665, 310, 290, 50);
25             onetimeboost.setBackground(Color.green);
26             onetimeboost.setBorder( bevlBorder );
27
28         }catch(Exception e){
29             JOptionPane.showMessageDialog(null,"Error");
30
31         }
32     }
33 }
34
35
36
```

Figure No: 1.22- interface for player class coding

(Anon., n.d.)

## Task-2

### Testing Data:

#### (1) Developing a test plan:

### Testing Data

I have used 3 types of testing here;

- White box testing
- Black Box testing
- Boundary testing

#### (2) Implementing test plan:

### White Box Testing

The white box testing uncovers the errors in the conditions and internal program codes.

This testing is performed below;

### Test -1

Own 20 points by clicking on the main button.

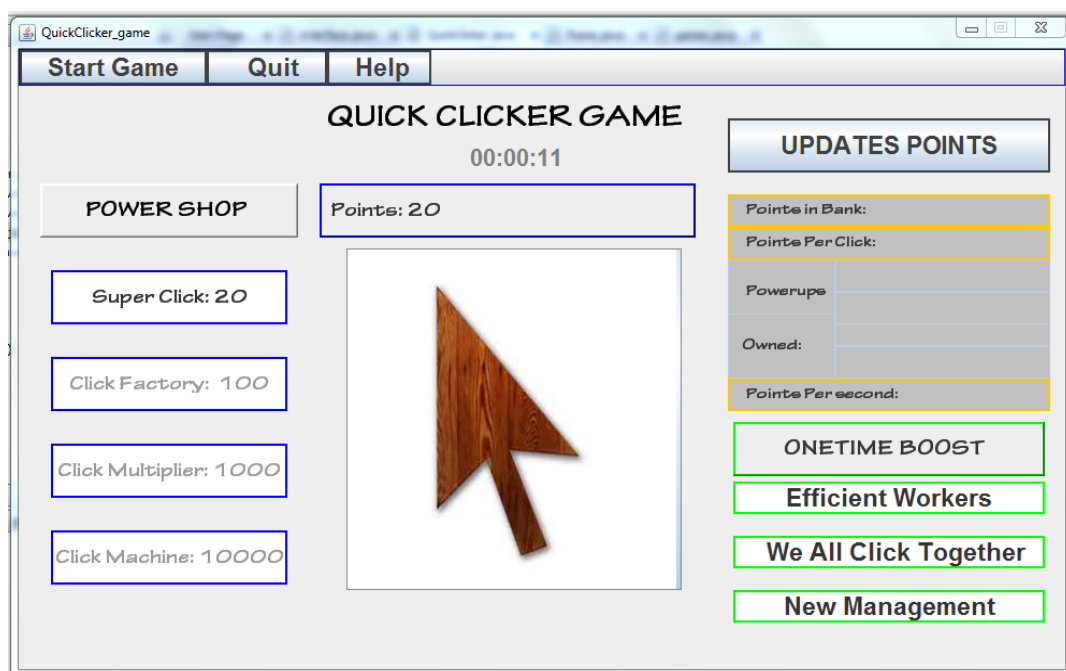


Image-1: Test-1

After that, buy super click. After buying it has cut 20 points from the total score.

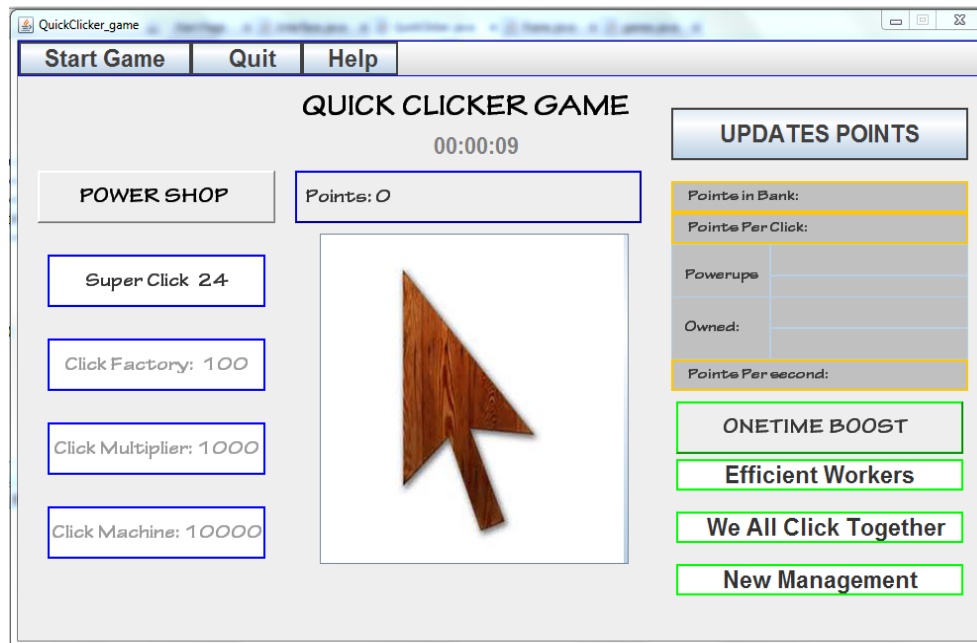


Image-2: Test-1

## Test -2

After clicking **Efficient Workers** click on **Super Click**.

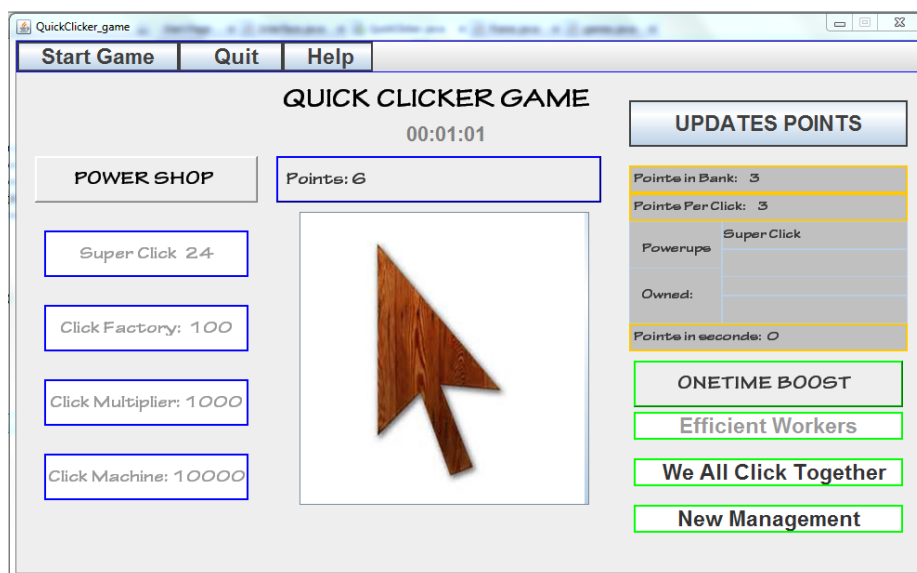


Image-3: Test-2

The **Super Clicker** gave 2 point increment in one purchase. **Efficient Workers** increased the Power ups power.

**Test -3**

Click on the **Click Factory** after **Efficient Workers**;

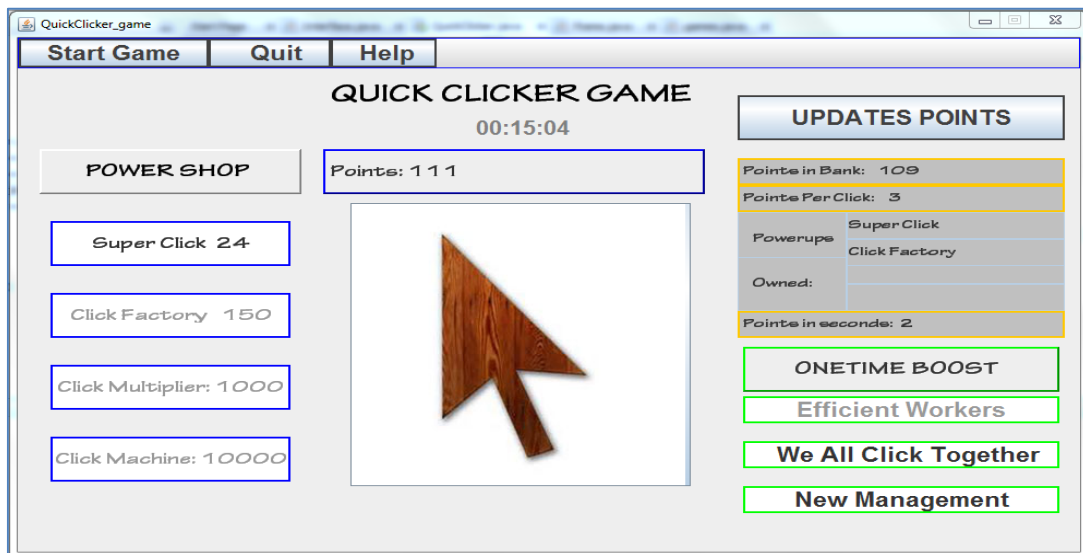


Image-4: Test-3

It is giving 2 points in every 10 seconds.

**Test-4**

Initial state;

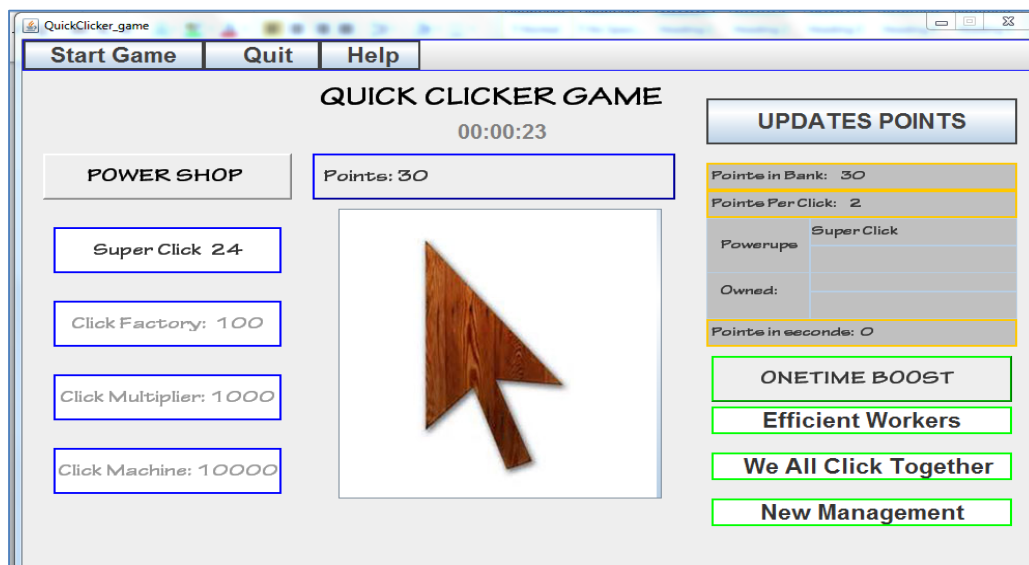


Image-5: Test-4

Now click **New Management**. It increased the points 10%.

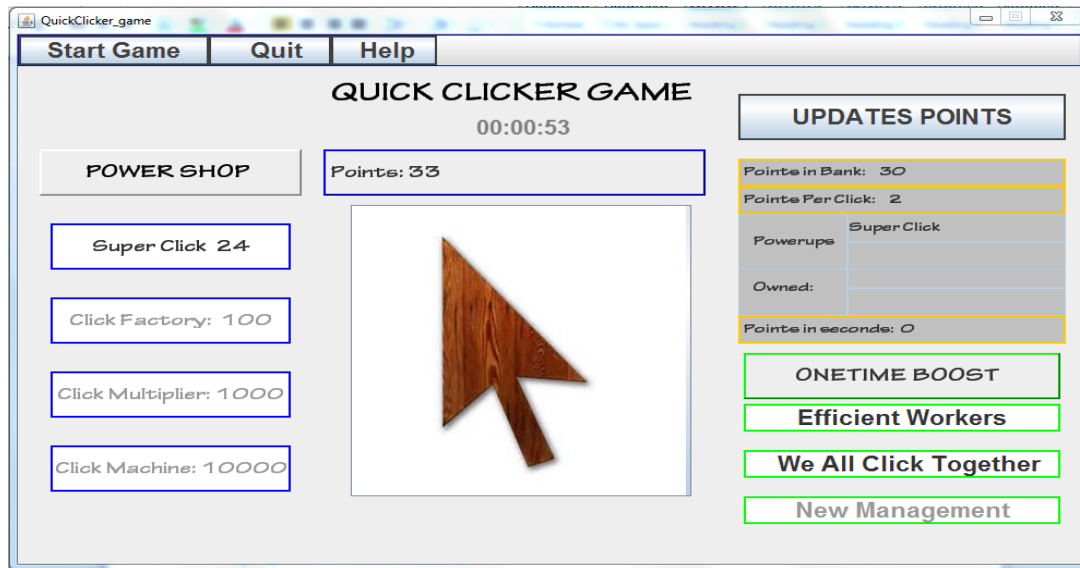


Image-6: Test-4

The initial value of **Click Multiplier** is 1000 points. After buying it the cost becomes 2000. So, the cost increased.

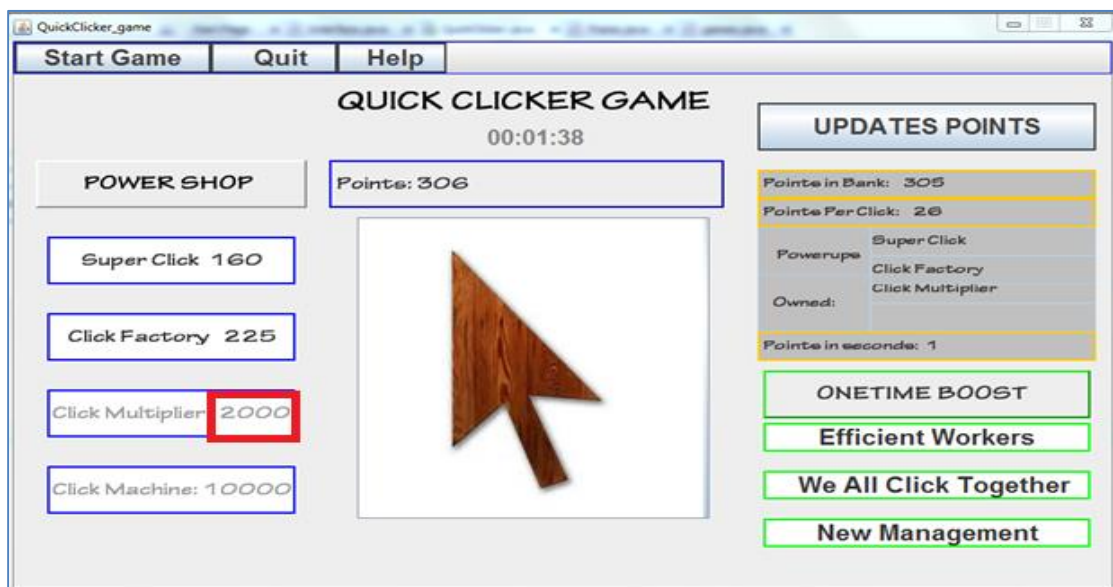


Image-7: Test-5



Table for White Box Testing:

Test No.	Input	Expected Outcome	Original Outcome	Comments
<b>Test-1</b>	Click on <b>Super Click button</b>	Decrease of 20 points from total point	20 points decreased	Right out come Image-1&2
<b>Test-2</b>	Click on <b>Efficient Worker button</b> then <b>Super click</b>	Risein the power of <b>Super Click</b>	The power of Super Click increased.	Right out come Image-3
<b>Test-3</b>	Click <b>Efficient Worker</b> then <b>Click Factory</b>	Risein the power of <b>Click Factory</b>	The power of Click Factory increased.	Right out come Image-4
<b>Test-4</b>	Click <b>New Management</b>	The rise I in the total point by 10%	Points increased 10%	Right out come Image-5 & 6
<b>Test-5</b>	Click <b>Click Multiplier</b>	The cost growth	The cost grew	Right out come Image-7 & 8

Figure No: 2.1- interface of White Box Testing

## Black Box Testing

### Testing-1

Click on Updates button. The Update shows all the points and powers gained.

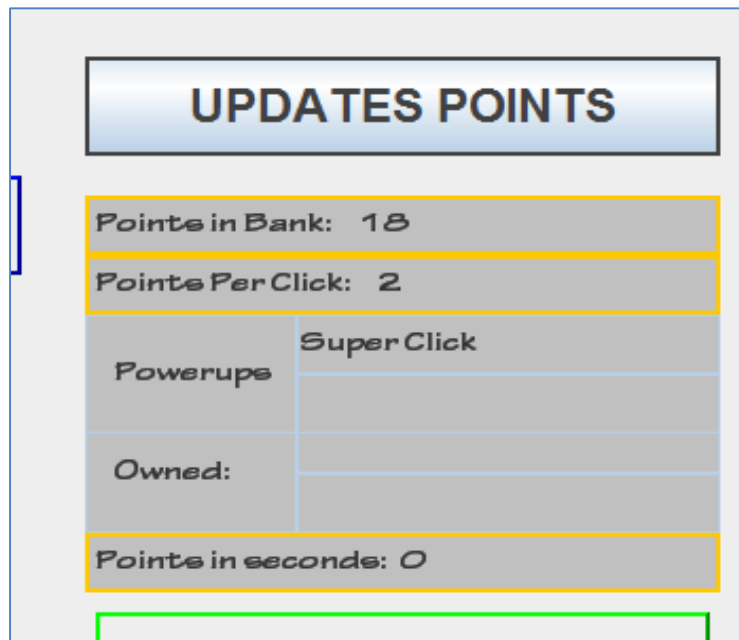


Image-1: Test-1

### Testing-2

Click on **Start Game**. A new game started.

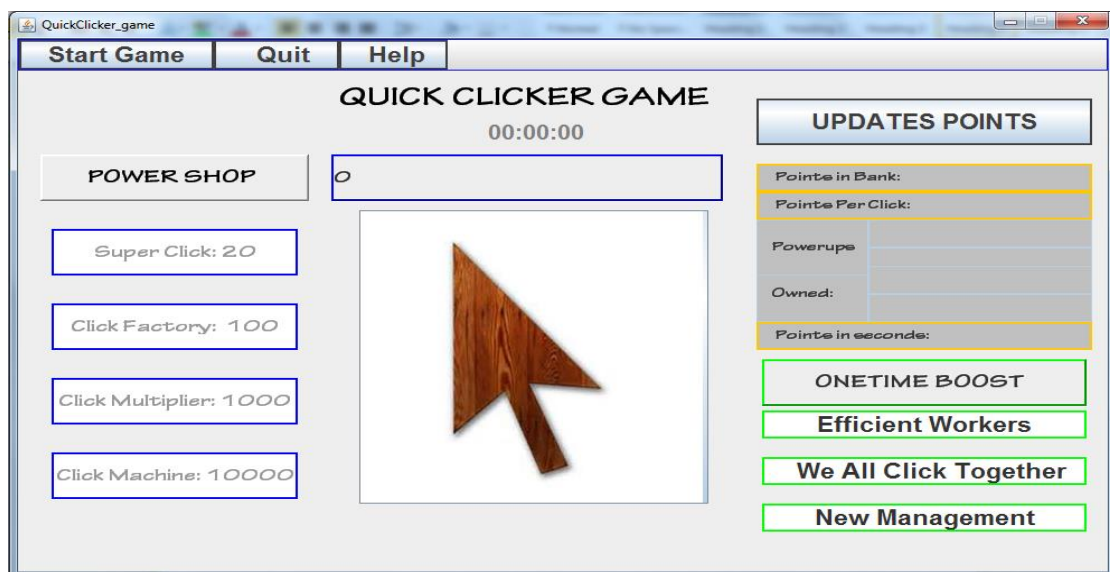


Image-2: Test-2

### Testing-3

If the point is less than 100 then the **Click Category** Remains disabled.

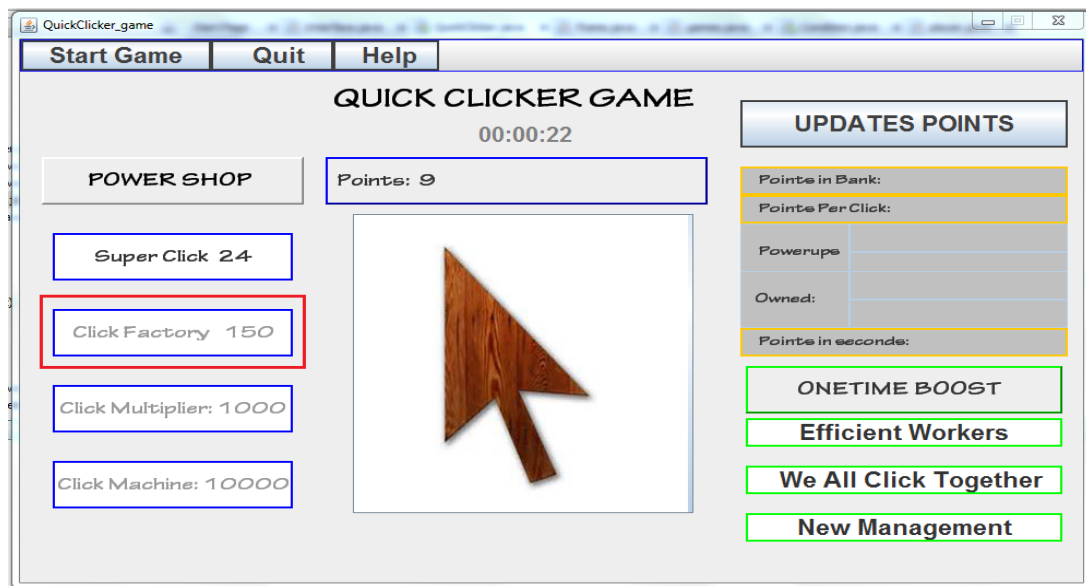


Image-3: Test-3

### Test-4

If the **Efficient Click** is clicked then it becomes Disabled.

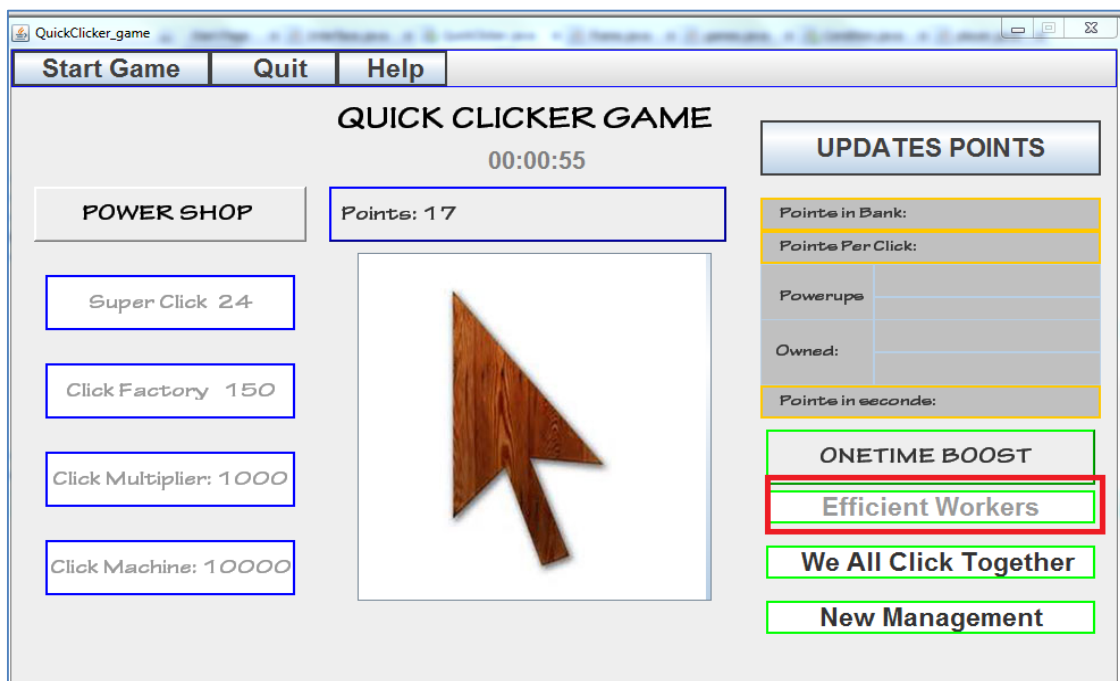


Image-4: Test-4

Table for Black Box Testing

Test No.	Input	Expected Outcome	Original Outcome	Comments
Test-1	Click on <b>Start Game</b>	A new games starting	A new game started.	Right out come Image-1
Test-2	Collect 100 points	Click Factory Enabling	Click Factory Enabled	Right out come Image-2
Test-3	Click <b>Efficient Worker</b>	Button Disabling	Button Disabled	Right out come Image-3

Figure No: 2.2- interface of Black Box Testing

## Boundary Testing

Boundary test finds out errors in the numbers and condition boundaries.

### Testing-1

After collecting 20 points the Super Click Enables.

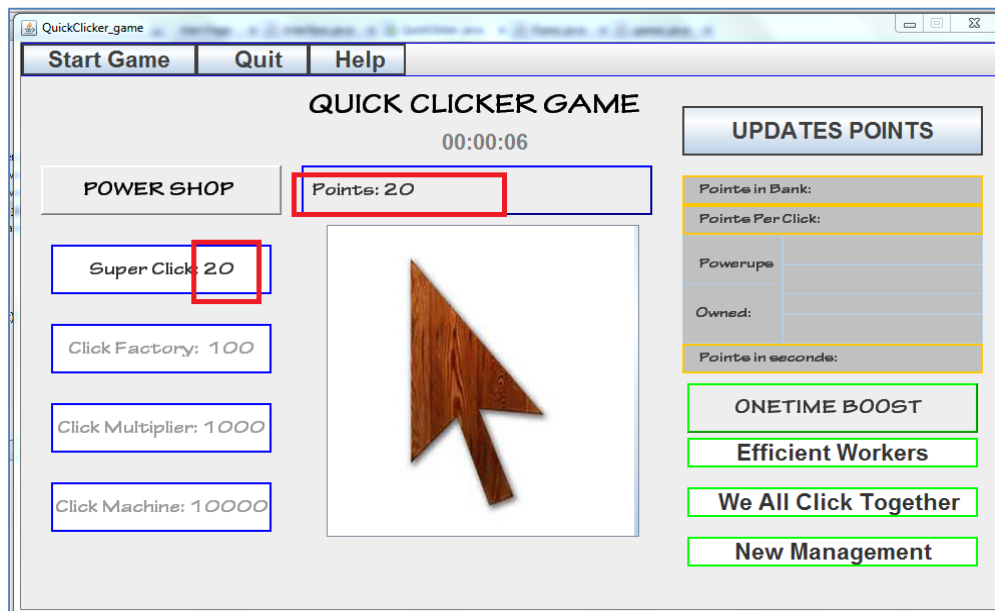


Image-1: Testing-1

### Testing-2

After gaining Click Factory the points rise by 1 in 10 seconds.

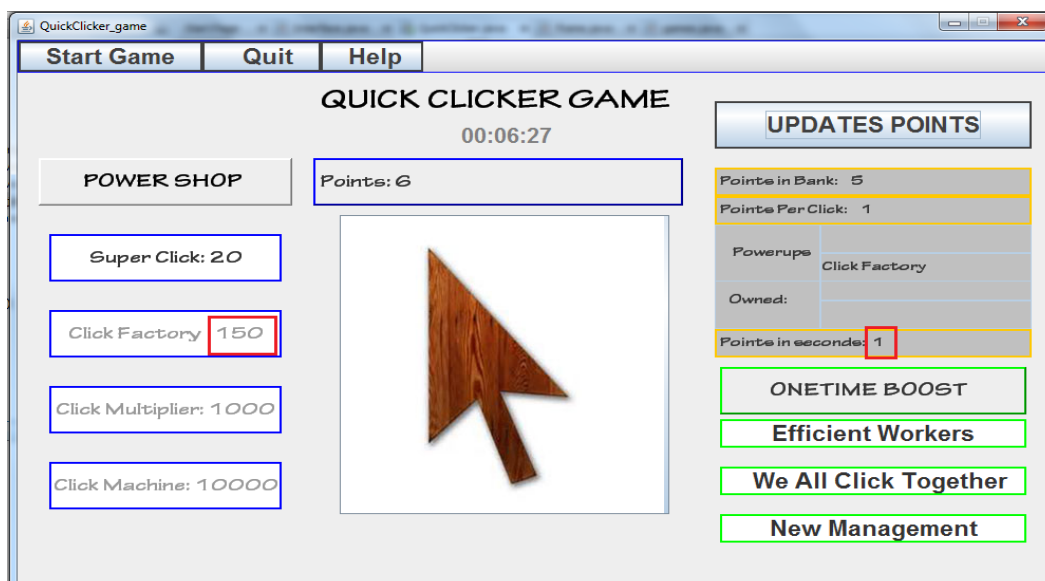


Image-2: Testing-2

### Testing-3

Click on **Click Multiplier**. It multiplied the points increase in per click by 2.

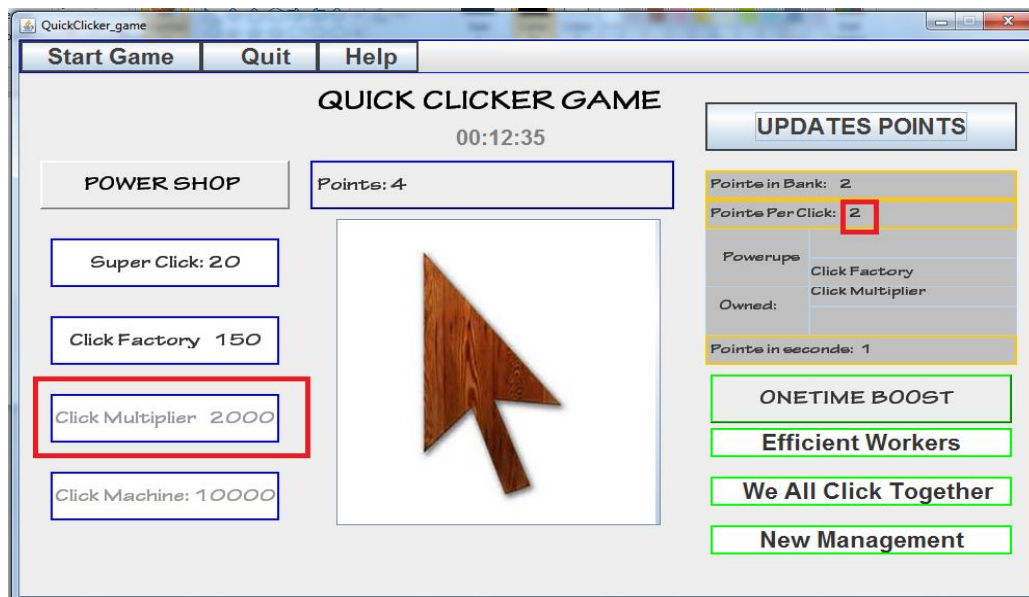


Image-2: Testing-3

### Table for Boundary Testing

Test No.	Input	Expected Outcome	Original Outcome	Comments
Test-1	Collect 20 points	Super Click Enable	Super Click Enable	Right Outcome Image-1
Test-2	Click on <b>Click Factory</b>	Points in 10 seconds	Grew 1 point in 10 seconds	Right Outcome Image-2
Test-3	Click on Click Multiplier	The points per click rising by 2.	The points per click rose by 2.	Right Outcome Image-3

Figure No: 2.3- interface of Boundary Testing

(Anon., n.d.)

## Exception Handling

I have handled an exception where it can have a hang or system drop.

```
public player() {  
    try{  
        onetimeboost = new JLabel("          ONETIME BOOST");  
        this.add(onetimeboost);  
        onetimeboost.setFont(new Font("Tekton Pro Ext", Font.BOLD, 20));  
        //upgraLB1.setForeground(Color.white);  
        onetimeboost.setBounds(665, 310, 290, 50);  
        onetimeboost.setBackground(Color.green);  
        onetimeboost.setBorder( bevelBorder );  
    }catch(Exception e){  
        JOptionPane.showMessageDialog(null,"Error");  
    }  
}
```

Figure No: 2.4- interface of Exception handling coding

### Short discussion of Data Testing:

First of all I have tested the program properly. I have tested this program in three ways. Such as: White Box testing, Black Box testing and Boundary testing. My program runs successfully. At First i have opened this program then clicked on point's buttons to increase the point. If the player has gained 20 points then the player can buy a super click. Same way I have tested all systems. That means I have tested **white box testing** by condition and **black box testing** by output test as well as **boundary testing** by the output points.

### Task-3

#### Design Documentation:

#### Class Relationships with Methods Attributes:

Here I have shown on all classes with their Relationships and attributes.

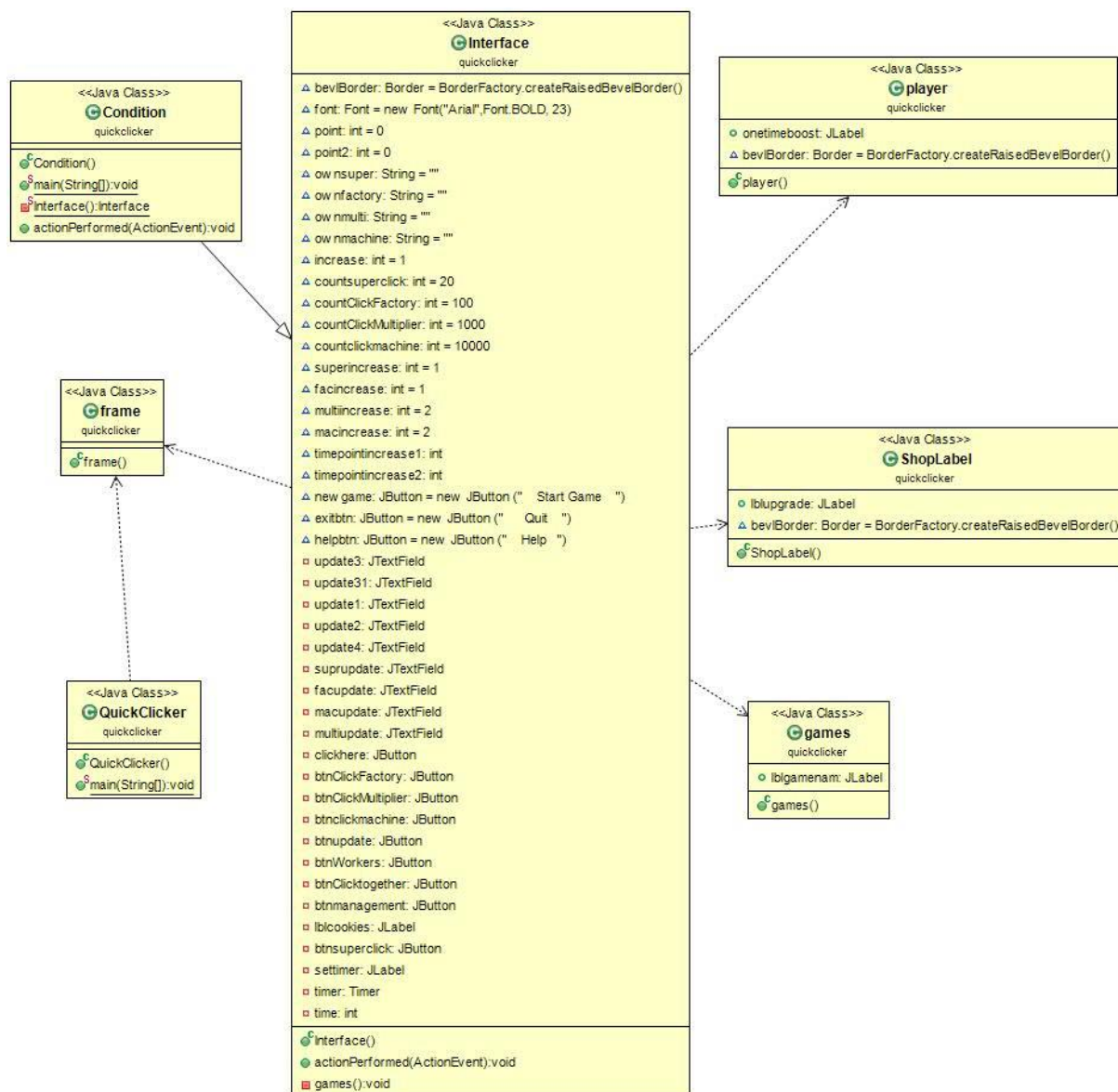


Figure No: 3.1- interface of all class Relationship with Methods Attributes

(Anon., n.d.)



**Conclusion:**

At the end of this assignment I have had a greatest working experience. I got to gained knowledge to increase. Self-confidently my experience will help me for better achievement in the near future where designing and developing object-oriented computer program problems will come out.

**Bibliography**

Anon., n.d. [Online] Available at: <http://orteil.dashnet.org/cookieclicker/> [Accessed 15 oct 2015].

Anon., n.d. [Online] Available at: <http://orteil.dashnet.org/experiments/cookie/> [Accessed 15 oct 2015].

Anon., n.d. [Online] Available at: <http://orteil.dashnet.org/cookieclicker/beta/> [Accessed 17 oct 2015].

Anon., n.d. [Online] Available at: <http://orteil.dashnet.org/cookieclicker/betadungeons/> [Accessed 20 oct 2015].