CMP 2004 Term Project Report   
CannonGame

Derya ŞİŞMANYAZICI & Alper POSLU

# Project Purpose and Game Rules

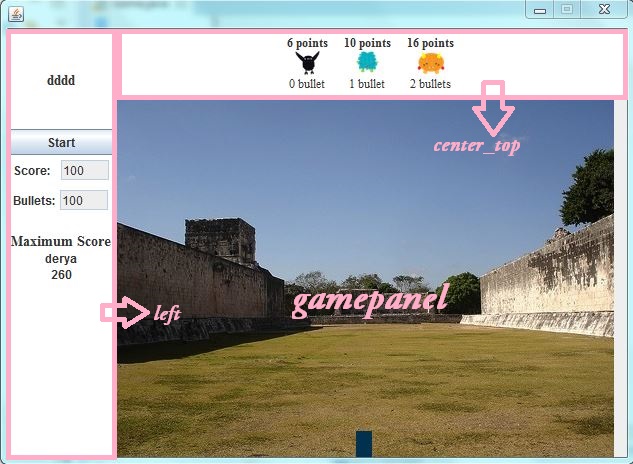
The aim of this game is shutting Bushy, Crow and Devil which come to our caste from garden.  
We have cannon which return max. Theta is pi/2 and sends a fire to creatures. When fire cross to creature its appearance returns to a ghost and with animation the ghost goes to up for a second and disappear.

All creatures have different points and affect our bullets differently.  
Firstly, shutting a Crow is 6 points, Bushy is 10 points and Devil is 16 points. According to kind of creature, gamer wins extra bullet such as a bullet for Bushy and two bullets for Devil.

The second rule is dangerous area. If creature’s randomly circle route won’t be gamer’s castle (ıf it will go to empty air), gamer will not take any point despite s/he kill the creature.

After, entering username and press “start” button you can begin and you can stop whenever you want.  
You will see the highest score & owner on the left and “max” txt file.

# Project Code (Bonus)

*Game.java (main part)*  
At first, we create a JOptionPane to ask username and checked it if it is in 3-18 characters.   
After that ;  
  
Firstly we created left because we wanted to add username, button, and score etc. one by one and vertically. If we used north center and south, there could be only three options but in this way we added them inside of other and created multiple in left. After left side, we just had center so we created center\_top so remaining area center is our “gamepanel”.  
  
**The important thing** we used synchronized functions in game.java.  
we used;  
**public** **synchronized** **void** addScore(**int** s) {

score += s;

score\_field.setText(""+score);

updateMaxScore();

**if** (score <= 0) {

gameOver = **true**;

JOptionPane.*showMessageDialog*(**this**, "Game is over: Your score is <= 0.");

}

}

**public** **synchronized** **void** addBullets(**int** b) {

bullets += b;

bullets\_field.setText("" + bullets);

}  
So it provides changing score and number of bullets at the same time.  
  
After that in populate\_left function, we added non editable textfield to write score and bullets, we added max score and username. At the same time we used scanner for reading and writing this information.  
In populate\_top function we gave information about creature icons.   
  
**Another important part is;**  
We used action listener for start/pause button. After that we used   
gamepanel.requestFocusInWindow()  
  
FocusInWindow provides us, when gamer open a new game start button will be active. After he clicked it our gamepanel will be active. Namely, when the user opens the game, he can’t begin game without click start button.  
At the end of game.java, we wrote main function.  
  
*VisualObject.java*  
we created visualObject.java to draw our creature icons, cannon and our ball.  
  
*Cannon.java*We implement VisualObject.java and with draw method we draw a renctangle.  
We set its color and gave it a position;  
 **public** **void** draw(Graphics g) {

**int** X=x-*W*/2;

g.setColor(**new** Color(0,50,80));

g.fillRect(x-30, y-1, 60, 2);

**while**(X<x+*W*/2){

g.drawLine(X, y, X-(**int**)(*H*\*Math.*cos*(theta)), y-(**int**)(*H*\*Math.*sin*(theta))) ;

X++;

}

g.setColor(Color.*BLACK*);

}  
and with increase and decrease functions we moved our cannon’s direction pi/16.  
  
*Creature.java*  
We created creatures such as giving their theta, their icon. We used Boolean isDangerous function. WE looked at creature’s slope and (x,y )coordinate. We added creature’s height and its slope\* x-y and divide it to it’s slope and foud creature’s place in x coordinate. So we determine if it is dangerous or not. (as we explained in game rules)  
We had other Boolean function hitBy. In this function we looked creature and ball’s coordinate and if they are same or not.  
  
We used creatures as a thread. In Thread we used CreatureAnimater.  
  
*Bushy.java / Devil.java / Crow.java*

Firstly we implemented VisualObject . We took its icon. We used   
**super**(panel, x, y, **new** ImageIcon(Game.**class**.getResource("/bushy.png")), theta);  
  
To make jar file we need a images folder. When we put images in project folder we couldn’t make a jar file so we put all images in images folder. To take our images in java file, we used (**“/iconname.png”)**  
With getPoints function we determine its point and getEarnedBullets determined if it will win bullet or not. And we used these steps for there of them.  
  
*Ball.java*We implemented VisualObjectJava and used same steps same as Devil/Bushy etc. and create our ball.  
We used thread for balls.   
**class** BallAnimater **extends** Thread{

Ball ball;

**public** BallAnimater(Ball ball) {

**this**.ball = ball;

}  
  
  
We had again synchronized function  
 **synchronized** **private** **void** destroyCreatures() {

ArrayList<VisualObject> v = **new** ArrayList(ball.panel.objects);

**for**(VisualObject o:v){

**if**(o **instanceof** Creature){

Creature creature = (Creature)o;

**if**(creature.hitBy(ball)){

creature.setKilled();

creature.theta=ball.theta;

**if**(creature.isDangereous()){

ball.panel.game.addScore(creature.getPoints());

ball.panel.game.addBullets(creature.getEarnedBullets());

}

ball.panel.repaint();

}

}

}

}

}  
If o is a creature, we checked if it is dangerous or not and if it hit to a creature we called setKilled function. Otherwife we called repaint method.  
  
*GamePanel.java*  
  
In there we wanted to show only 3 creatures on the screen, at the same time. And added a arraylist.  
**static** **int** *max\_num\_creature\_onscreen*=3;

ArrayList<VisualObject> objects = **new** ArrayList();  
we added our cannon.

With paint method we gave our background image.  
We used createCreature function. We generated random number and according to this number we determined which creature will be shown. We implemented

**public** **synchronized** **void** removeCreature(Creature c)  
function.  
We extended keyAdepter to manage our cannon. (we added comments in code)  
We implement our run function. We determine stop and gameover situation