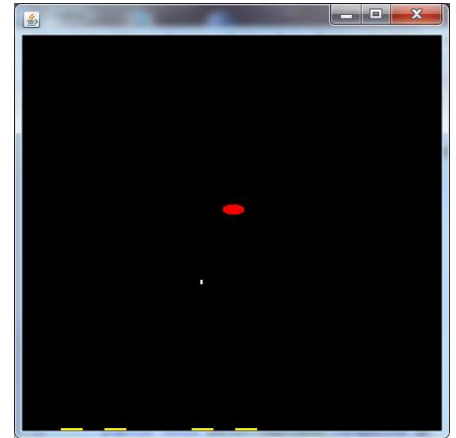


Department of Computer Science & CINSAM NKU Summer Programming Workshop 2015

Project 13: Animation



Enhance one or both of the Bouncing and Spaceship programs as follows.

Bouncing:

1. Add a variable called `blink`, set equal to 0. When a collision occurs, set `blink` to 5. In `paintComponent`, if `blink > 0` then “blink the screen” (change the background color) and decrement `blink`. If `blink == 0`, the background color will be black, but as collisions occur, the color changes for a few instances.
2. When a collision occurs, remove one of the colliding balls from the screen by setting it to null. For instance, `if(balls[i].collides(balls[j])) { ...; balls[j]=null;}` If you do this, you will have to change every occurrence of `balls[i].somemessage()` to be `if(balls[i]!=null) balls[i].somemessage()` to avoid getting `NullPointerExceptions`. You can also make `balls[i]` grow. Add a `grow` method in the `Ball` class which increases the size of the ball. Upon colliding, now do `if(balls[i].collides(balls[j])) { ...; balls[i].grow(); balls[j]=null;}`
3. Add 1 or more “holes” in the environment so that if a ball gets near a hole, it is removed from the screen (set `balls[i]` to null for that ball). Think of the hole as a pocket in a pool game.

Spaceship2

1. Enhance this program so that there are targets on the bottom of the screen. Have 8 targets at the bottom of the screen of size 30,5, equally spaced out. They will occur at 35-65, 135-165, etc in the x-direction and be between `Y_SIZE-50` and `Y_SIZE-45` in the y-direction. When a Spaceship’s bomb hits a target, add one to the user’s score. Display the score at the top of the screen. To determine if the bomb hits a target (in `actionPerformed`), use the following logic:

```
for(int i=0;i<8;i++) if(bx>...&&bx<=...&&by>=...&&by<=...) ...
```

You will need to figure out the values to compare where it has ... above. To draw the targets in `paintComponent` use `for(int i=0;i<8;i++) g.fillRect(..., Y_SIZE-50, 30, 5);`
2. Remove the target once it has been hit. Have the game run until either all of the targets have been hit or the spaceship has used some maximum number of bombs. In `actionPerformed`, you can test to see if `score==8` and if so, do `repaint();` and `t.stop();` See the note below about how to implement the targets so that they disappear after being hit.
3. Enhance the bomb dropping routine to only drop a bomb if the spaceship is not going to fast laterally (in the `dx` direction) and if the spaceship is in the bottom half of the screen. This will make it more likely to hit a target.

To do number 2, add an array called `targets=new boolean[8]` all initially set to true. Add to the if statement in `actionPerformed` (`targets[i]==true`) and then if it is a hit, part of the if clause should set `targets[i]=false;` In `paintComponent`, change the for loop so that it reads `for(int i=0;i<8&&targets[i]==true;i++)`. This ensures that the target must exist to be drawn.