ECS724 Interactive Digital Media Techniques · Autumn 2014

Mini-Assignment 2: Processing-Tank Wars

Beici Liang 140253278

Step 1: Draw the terrain and the tanks

1. Firstly, I just used three times of *line()* function to draw the boundary of the terrain. Then I found it was hard to colour the whole area of terrain. Thus I used *line()* in a *for* loop to draw a vertical line at each X location.
2. The stroke of each item should be declared before drawing. Otherwise it would show a colour or weight I did not mean to.
3. In *arc()* function, the angle starts going clockwise, so it should be *PI* to *TWO_PI* rather than *0* to *PI* if I would like to draw a semicircle above the terrain.
4. When drawing the cannon in *line()* function, *sin()* and *cos()* should be used to calculate the coordinates of the second point. It is noticed that we operate addition to calculate x-coordinate, but subtraction to calculate y-coordinate.

Step 2 & 3 & 4: Handling events at the keyboard & Fire a projectile & Gravity

1. Since there are two players, use *if()* function to decide which player's turn is going on in the *keyPressed()* function. Then we can adjust the angle and strength of each tank's cannon by arrow keys separately.
2. Unlike the arrow keys, which show up as *CODED* because of no printable character, the space bar is shown as *key == ' '*. When it is pressed, remember to turn the *projectileInMotio*n to *true*.
3. The initial position and velocity of the projectile are set before the *projectileInMotio*n turns *true*. I set the initial position as the muzzle of cannon, and velocity is decided by strength.
4. Move the projectile in *updateProjectilePositionAndCheckCollision()*. Update the position by adding velocity, and vertical velocity is influenced by *gravity*.

Step 5: Collision Detection

1. If I just consider the situation that when projectile hits the ground, I cannot continue to play when the projectile hits the left, right, or top border. Therefore I set boundaries for all the directions. Whenever the projectile hits the border, it stops moving. Especially when the distance between projectile and one tank is less than or equal to the tank's radius, the other player wins. Otherwise, call *nextPlayersTurn()*.

Step 6: Customise your game

1. For the appearance of the game, I add a colour-changing ribbon and draw a blue sky in *drawGround()*, and change colour of the text in *drawStatus()* as well.
2. I add *wind* to update the horizontal velocity. Remember to randomise the *wind* again when it comes to the next player's turn, and make the value displayed by adding code to *drawStatus()*.
3. When the projectile hits something, call *goBoom()* to draw the explosion, which will get bigger and turn from red to yellow over time.
4. When a tank is hit by the projectile (i.e. the projectile is within the area of the tank's semicircle), this tank will turn black, which is operated by *if()* function in *drawTanks()*.
5. Finally, I upload an image of sun to the blue sky, and make it rotate.