Click Game part 3

Step 5: Randomize Circle

Now, let’s randomize the size and position of our circle. This will make the game more challenging.

This is a pretty simple change. We just need to update our `showCircle()` method so that it doesn’t set the value of the circle variable. Instead, we will give them a random value inside of our `toggleCircle()` method.

The code now looks like this:

```
int circleX;
int circleY;
int circleSize;
int score = 0;
boolean circleVisible = false;

void setup() {
  size(800, 600);
}

void draw() {
  background(0);
  showScore();
  toggleCircle();
  showCircle();
}

void toggleCircle() {
  if (frameCount % 60 == 0) {
    circleVisible = !circleVisible;
    circleSize = (int)random(25, 150);
    circleX = (int)random(circleSize, width-circleSize);
    circleY = (int)random(circleSize, height-circleSize);
  }
}

void showCircle() {
  if (circleVisible) {
    fill(255, 0, 0);
    circle(circleX, circleY, circleSize);
  }
}

void mouseReleased() {
  if (dist(mouseX, mouseY, circleX, circleY) < circleSize/2) {
    score++;
  } else {
    score--;
  }
}

void showScore() {
  fill(255); // make's text white
  textSize(24);
  textAlign(CENTER);
  text("Score: " + score, width/2, 50);
}
```

Notice that when setting the random values for the circle, it says **int** in parenthesis. `circleSize = (int)random(25, 150);` This is because `random()` generates a `float`, but our variable is an `int`.

Therefore, we need to tell Java to convert the float into an int for us.

Also, notice that I have the program choose a value for x, and y that are between the `circleSize` and the screen size - the `circleSize`. This way the circle will never be on the edge of the screen.

Step 6: Game Over

Let’s limit our game to 10 rounds. After that, we will display a game over screen.

For this we need:

1. A variable to keep track of the number of turns. This will be double the actual number of turns, because it will count both when the circle is visible and when it is invisible.
2. Our `toggleGame()` method needs to subtract 1 from the `turns` variable every time it toggle the circle on and off.
3. An `endGame()` method that will display the end game screen.
4. An `if / else` in the draw method to choose between the game screen and the end game screen. It will trigger the end game when turns equals 0.
5. A `boolean` to keep track of whether the game is over or not. We will use this to stop recording clicks when the game is over. Update your `mouseReleased()` method to only subtract points when the game is NOT over `(!gameOver)`.

The code now looks like this:

```
int circleX;
int circleY;
int circleSize;
int score = 0;
boolean circleVisible = false;
boolean gameOver = false;
int turns = 20;

void setup() {
  size(800, 600);
}

void draw() {
  background(0);
  showScore();
  if (turns > 0) {
    toggleCircle();
    showCircle();
  } else {
    endGame();
  }
}

void toggleCircle() {
  if (frameCount % 60 == 0) {
    circleVisible = !circleVisible;
    circleSize = (int)random(25, 150);
    circleX = (int)random(circleSize, width-circleSize);
    circleY = (int)random(circleSize, height-circleSize);
    turns--;
  }
}

void showCircle() {
  if (circleVisible) {
    fill(255, 0, 0);
    circle(circleX, circleY, circleSize);
  }
}

void mouseReleased() {
  if (dist(mouseX, mouseY, circleX, circleY) < circleSize/2) {
    score++;
  } else if (!gameOver){
    score--;
  }
}

void showScore() {
  fill(255); // make's text white
  textSize(24);
  textAlign(CENTER);
  text("Score: " + score, width/2, 50);
}

void endGame() {
  background(0, 0, 255);
  gameOver = true;
  text("Game Over!", width/2, height/2);
  showScore();
}
```

Step 7: New Game

Now we just need a way for players to play again.

For this, we'll use the Processing `keyPressed()` method to detect when a key is pressed. We will use the SPACEBAR to start a new game.

While we're at it, let's start the game with this screen, and wait to start playing until the player presses the SPACEBAR.

For this we need:

1. To create a `newGame()` method that resets all of the game variables.

2. Create a `keyPressed()` method that will call the `newGame()` method when pressed.
3. Start our program set to `gameOver` so that the end game screen displays right away, until the player presses SPACEBAR. To do this, set the gameOver variable to start as true. Change the `if` inside of draw.
4. Add an `if` to the `toggleCircle()` method that sets `gameOver` to true when `turns == 0`.

The Final code should look like this:

```
int circleX;
int circleY;
int circleSize;
int score = 0;
boolean circleVisible = false;
boolean gameOver = true;
int turns = 20;

void setup() {
  size(800, 600);
}

void draw() {
  background(0);
  showScore();
  if (gameOver) {
    endGame();
  } else {
    toggleCircle();
    showCircle();
  }
}

void toggleCircle() {
  if (frameCount % 60 == 0) {
    circleVisible = !circleVisible;
    circleSize = (int)random(25, 150);
    circleX = (int)random(circleSize, width-circleSize);
    circleY = (int)random(circleSize, height-circleSize);
    turns--;
  }
  if (turns == 0) {
    gameOver = true;
  }
}

void showCircle() {
  if (circleVisible) {
    fill(255, 0, 0);
    circle(circleX, circleY, circleSize);
  }
}
```

```
void mouseReleased() {
  if (dist(mouseX, mouseY, circleX, circleY) < circleSize/2) {
    score++;
  } else if (!gameOver) {
    score--;
  }
}

void showScore() {
  fill(255); // make's text white
  textSize(24);
  textAlign(CENTER);
  text("Score: " + score, width/2, 50);
}

void endGame() {
  background(0, 0, 255);
  gameOver = true;
  text("Game Over!", width/2, height/2);
  text("Press SPACE to play.", width/2, height-200);
  showScore();
}

void newGame() {
  turns = 20;
  score = 0;
  background(0);
  gameOver = false;
}

void keyPressed() {
  if (key == ' ') {
    newGame();
  }
}
```

Challenges

Although our game is mostly complete, there are some things to fix and improve, or just try out for fun.

1. Can you change the program so that instead of saying “Game Over” when the game starts, it displays a different message when the game starts, and only shows “Game Over” after a round is played? (Hint: You may want to try adding a `boolean` and an `if` statement.)
2. Can you modify the program so that the circles get smaller each round, instead of being a random size? (Hint: You can make the size some multiple of the current turn count.)
3. Can you modify the program so that it speeds up each round?
4. Can you make the circle a random color each round, or make the color get darker and darker each

round, making it harder to find?

5. Can you add a high score? (This will reset each time you start the program, but should save the highest score of all of your tries while the program is running.)