

Circle Clash

Instructor Guide

[Overview](#)

[Learning Goals](#)

[Personal Growth Goals](#)

[Skills Required](#)

[Resources Required](#)

[Instructor Preparation](#)

[In Depth Description of Lab Activities](#)

[Part 1 - Drawing Rectangles \(rectangles1.py\) \(20 minutes\)](#)

[Part 2 - Guessing Rectangles \(rectangles2.py\) \(10 minutes\)](#)

[Part 3 - Looping Rectangles \(rectangles3.py\) \(10 minutes\)](#)

[Part 4 - Random Rectangles \(rectangles4.py\) \(15 minutes\)](#)

[Part 5 - Bullseye \(bullseye.py\) \(15 minutes\)](#)

[Part 6 - Challenge Question: Color Board \(colorboard.py\) \(20 minutes\)](#)

[Lesson Plan](#)

[Additional Resources / Further Drawing Fun!](#)

Overview

Learning to make a Circle Clash game with animation and key input!



Learning Goals

- Learning the animation framework and what each function does
 - `init()`
 - `keyPressed()`
 - `timerFired()`
 - `redrawAll()`
- How to use data to store info (but not any detail on what a Class or Struct is yet)
 - `data.playerX`, `data.playerY`
 - `data.gameOver`
- Further using the drawing functions from last session:
 - `create_oval`
 - `create_text`

Personal Growth Goals

- Confidence in adding features to a project/game
- Navigating frustration in debugging a multi-part coding file

Skills Required

- How to input code into a file and run code
- Basic Geometry
- Distance Formula
- Using integers, booleans, strings
- Conditionals

Resources Required

- Personal Computers for each student
- Paper and pencil so mentors and students can draw out what the predicted drawings will be
- Small groups working on the labs together with 2-3 students and a mentor

Instructor Preparation

Pre-load the computers with the programming files, make sure tkinter works

In Depth Description of Lab Activities

Part 0 - Warmup Question (and learn create_text)

Open a python file with the drawing code from last week and ask the student to:

- Draw a rectangle
- Then add a square and a circle, none of the shapes touching, all different colors
- Then add text (new idea) with their name, like so:
 - `canvas.create_text(centerX, centerY, text="My Name", font="Arial 20", anchor="center")`

Part 1 - Changing a Basic Animated Game

1. Open "07_basic_animated_game.py"
2. Run the file, play the game
3. Have the student look at the code and try to change the following things:
 - a. make the text at the top left instead
 - b. Invert the controls (up is down, right is left, etc)
 - c. Make the player circle blue and a bigger size
 - d. Make the player need to be exactly in the middle of the text to win
4. Optional challenge problems
 - a. Make the text move once you reach it, incrementing a score
 - b. Make the game end when the player doesn't reach the text in a certain amount of time

Part 2 - Making a Basic Circle Clash

1. Run "08_basic_circle_clash.py" (trying to not let them see the code!)
2. Have them modify "08_circle_clash.py" to make that game.

Part 3 - Making an Advanced Circle Clash

1. Run "09_advanced_circle_clash.py" (trying not to let them see the code!)
2. Have them modify further what they have in "08_circle_clash.py" to make that game.
3. FYI you will need to teach them EXTRA CONCEPTS
 - a. Lists
 - i. Appending tuples to a list
 - ii. Accessing elements in a list
 - iii. Looping over them (for circle in data.circles...)
 - b. `random.choice` for choosing random circle colors (they used `random.randint` before, which could also work)
 - c. Modulo for things happening not EVERY frame

Lesson Plan

(:10) means that this part should be done by the tenth minute of the lesson

1. Part 0 - Warmup (:10)
2. Part 1 - Basic Animated Game (:30)
3. Part 2 - Basic Circle Clash (:55)
4. Part 3 - Advanced Circle Clash (:5 min to end)
5. Feedback (:last 5 min)

Further Reference

Tkinter functions:

<http://effbot.org/tkinterbook/canvas.htm>