FIRIA LABS

LESSON: CodeX Winter Olympics Project

Time: 50 minutes

Overview:

This project uses the graphics capabilities of CodeX to draw circles, rectangles and text. The first part of the project (steps 1 through 5) are simple enough for beginners. During the first part, you will write code that displays the five Olympic rings with text for the host country and year.

You can stop after the first part, or continue to the second part (steps 6 through 10), which is more at the intermediate level. During the second part, you will write code that enables the user to type information on the console that is inputted to the CodeX. The data is used to create bar charts of the medals for a single country.

Several ideas for an extension are given on the last slide. These are optional, and instructions are not given for the extensions. They are suggestions for students who want to challenge themselves to do more and apply their learning to this project.

Coding Objectives:

- I can use the draw features of CodeX to draw circles and rectangles.
- I can display text on the screen using display.print().
- I can display text on the screen using display.draw_text().
- I can give input to the CodeX through the console.
- I can convert a string to an integer.
- I can define and call a function with a return.
- I can define and call a function with parameters.
- I can use a for loop to display stacked rectangles.

Grades 6-8 CS Standards:

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-CS-03 Systematically identify and fix problems with computing devices and their components.

2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.

Grades 9-10 CS Standards:

3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3A-DA-11 Create interactive data visualizations using software tools to help others better understand real-world phenomena.

3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.

3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, or independent but interrelated programs.

3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

Grades 11-12 CS Standards:

3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.

3B-AP-10 Use and adapt classic algorithms to solve computational problems.

3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.

Preparation:

- Download slides
- Be familiar with the final code
- Read through the teaching guide
- Optional: Use the accessibility guide as needed.

In the folder:

- Winter Olympics project slides
- Winter Olympics final code
- Winter Olympics code with an introduction

Agenda:

- Warm-up (5 minutes)
- Complete program using slides (45 minutes) – could be longer with extensions

Teacher Notes:

- This project can be completed individually or using pair programming.
- Almost all mistakes made by students are typing mistakes. If students get errors when they run their code, first look over the code for spelling, punctuation and especially indenting.
- This project accesses the console panel for input. Make sure you and your students are able to find and use the console panel.

Other Extension Ideas:

- Add an introduction so when the code starts, you know what to do.
- Add labels for the bar chart.
- Use variables for the logo's circles so the logo can be placed randomly.
- Use overlapping circles for the medals instead of rectangles.
- Use random circles for the medals.
- Have the medal bars start near the top of the screen and fill in downward.
- Display data for more than one country.

Cross-curricular Connections:

- **LANGUAGE ARTS:** Have students research an athlete or sport and write a paper on their topic.
- **MATH:** Getting the center location of each circle requires some math. Discuss how you can calculate the x and y position without guessing.
- **MATH:** One of the extension ideas is to use variables for the circles so that the entire logo can be randomly placed. This takes more math so that each circle stays in its proper place.
- **MATH:** One of the extension ideas is to use stacks of circles instead of rectangles. Use math to calculate the center (x and y) and radius of the circles.
- **VISUAL ARTS:** This project shows how to draw using circles and rectangles (and text). Use graph paper and design your own image. Then write the code to display your image.
- **SOCIAL STUDIES:** The Olympics bring together people from all over the world. Study different cultures and countries.

Teaching Guide

Warm-up (5 minutes)

Slide 2 – This project is a way to display the logo and information about the Winter Olympics. You can warm up by:

- Talking about the host country.
- Discussing the different sports in the winter olympics. Which are the students' favorites?
- Looking back at the Olympics from 4 years ago. Pick a country and see what their medal count was.

Create/Run the Program (45 minutes)

💻 This project can be completed individually or with pair programming. Students should already have an account in CodeSpace.

💡 **Teaching tip:**

This project is not included in CodeSpace. Download and follow the slides. They include step-by-step instructions as well as code segments to guide students through the program code creation.

This project can be completed as a class, with showing the slides on a large screen. Or you can give students the slides and let them work at their own pace. It is also a great project for students who finish work early.

💻 **Slides 3-4**

These slides give instructions on getting started. Students log in and use the sandbox. They need to create a new file and import the codex library.

PART 1: Beginner Level

💻 **Slides 6-7: Step 1**

Write code to fill the display a solid color. The suggested color is light gray. Students can pick a different color, but it should not be BLUE, BLACK, RED, YELLOW or DARK_GREEN, since they are the colors of the logo.

💻 **Slides 8-9: Step 2**

Add code to draw a single circle (the first circle in the logo, which is BLUE). Students can pick their own location and radius, but a suggestion is given in the code.

💻 **Slides 10-11: Step 3**

Add code to draw the next two circles of the logo. Use a little math to calculate the x position of the circles.

💻 **Slides 12-13: Step 4**

Add code to draw the two circles on the bottom of the logo. Again, use a little math to calculate the x position and y position of the circles.

💻 **Slides 14-16: Step 5**

Add code to display text. Use the display.draw_text() function. This will probably be new to most students. It uses many keyword parameters, so they can be in any order but need the keyword with the value.

The suggestion on the slide is to use three rows of text. Students can change this to what they want to display. The scale adjusts the size of the text.

PART 2: Intermediate Level

💻 **Slides 18-20: Step 6**

Students start a new function that will get input from the console. First, they will display instructions on the CodeX screen. Call the function during testing.

💻 **Slides 21-22: Step 7**

Add code to get input from the console panel. Students need to be very careful with typing. The first input is a country and is a string. The next three are the medal counts, so they need to be integers. Use the int() function with the input() function.

💻 **Slide 23: Try it!**

Students should be able to type information in the console panel, but they will not yet see it. They just need to make sure everything is working fine up to this point.

💻 **Slides 24-25: Step 8**

Students will write a return statement for the get_info() function. Then they write a new function that calls the get_info() function. Change the function call to the new function – stacks(). The new function will just display the name of the country at first.

💻 **Slides 26-32: Step 9**

This step has lots of little steps that define a function for creating the bar chart and calling the function three times – one for each medal. This function will have several parameters. The order of the parameters must match the order of the arguments during a function call. The example shows the bars always starting at the bottom (y = 220). Alternatively, you could have the bars starting near the top and going down.

💻 **Slides 33-34: Step 10**

Add the main program with button presses so the user can choose either the logo or the medals.

💻 **Slide 35: Finished!**

Students do the final testing for their code. They should display the logo when a button is pressed, and input data and display the medals when another button is pressed.

💻 **Slide 36: Extensions**

Several ideas are given for students who want to challenge themselves and apply their learning. Instructions are not given for the extensions, but several have sample code available for the teacher.

Wrap-up / Optional (5 minutes)

💻 The project doesn’t require a wrap-up. You can use an extra-curricular activity from the list above, or have students reflect on the project and something they learned.