Course: Video Game Design in JavaScript (2023) | Module: Control Structures

Lesson 5.8: While Loops and Graphics

https://codehs.com/course/20517/lesson/5.8

Description	In this lesson, students will apply their understanding of while loops to graphics programs.
Objective	Students will be able to: • Write graphics programs that use while loops • Use variables to update the position and size of graphics objects within a while loop
Activities	5.8.1 Notes: While Loops and Graphics 5.8.2 Example: Lots of Circles 5.8.3 Example: Corners on Corners 5.8.4 Exercise: Concentric Circles 5.8.5 Debugging: Debugging: Circle Positions 5.8.6 Exercise: Growing Squares
Prior Knowledge	• Students should be familiar with conditionals and while loops.
Planning Notes	• This is a longer lesson that may need to be extended across two class periods. • This lesson does not have a lesson video. Instead, it uses notes to introduce students to while loops in graphics programs. • It might be helpful for students to draw out a program with pencil and paper to visualize how a shapes' position and/or size change each loop iteration. • This lesson is great for any students who are interested in art and in generative art. If time permits, give students some time to explore generative art and find an artist who interests them.
Standards Addressed	
Teaching and Learning Strategies	**Lesson Opener:** • Have students brainstorm and write down answers to the discussion questions listed below. Students can work individually or in groups/pairs. Have them share their responses. [5 mins]

- Connect while loops to generative art. Show students this [lovely video](#) of artist Vera Molnar as she discusses the relationship between art and computers.

Activities:

- Read through the *While Loops and Graphics* note. [10-20 mins]
  - This note introduces two important concepts when it comes to graphics and while loops: using a counter variable and using variables to set position and size. To ensure students fully comprehend the information presented in the note, consider reading through the note with students or using checks for understanding before moving on.
  - Students will have a chance to explore the two example programs in the following activities. You may want to have students open the *Lots of Circles* and *Corners on Corners* example activities so they can run the programs while reading through the note.
- Explore the *Lots of Circles* example. [5 mins]
  - Encourage students to connect this example program back to the flowchart in the previous note activity.
  - Have students adjust the value of `MAX_CIRCLES` and see how it impacts the program output.
- Explore the *Corners on Corners* example. [5 mins]
  - Encourage students to connect this example program back to the flowchart in the previous note activity.
  - Have students adjust the value of `DECREMENT` and see how it impacts the program output.
- Complete the *Concentric Circles* activity. [10-15 mins]
  - When it comes to determining the color of each circle, you can help students by making the connection of alternating to even and odd values.
- Complete the *Debugging: Circle Positions* activity. [10-15 mins]
  - It might help to have students draw out the desired program and then label the distance between circle centers.
  - Encourage students to keep debug mode on!
- Complete the *Growing Squares* activity. [20-30 mins]
  - Note that students have some flexibility in this program when it comes to the colors and title of their program. Encourage them to have fun with it!

Lesson Closer:

- Have students reflect and discuss their responses to the end of class discussion questions. [5 mins]

Discussion Questions	

Beginning of Class:

- Generative art is art that is created by using automated, non-human processes. Artists will set the rules and conditions for the art, but the art itself is created by a computer. Show students this piece of generative art by Mark J. Stock: CF11_1179. He describes this piece [here](#): "This piece is the result of an intricate numerical

computation on an initially-banded field of color. The amount of energy and detail in this piece must be seen in person." Then use the [See, Think, Wonder](#) thinking routine to start a discussion about the relationship between computers and art. Ask:

- What do you see?
- What do you think about that?
- What does it make you wonder?

- What is a similarity between a while loop and an if statement? What is a difference?

    - *Both an if statement and a while loop execute code only if a condition is true. They are different because an if statement only executes once, while a while loop will repeat as long as the condition is true or, put another way, until the condition is false.*
- Take a look at the code snippet below. What will happen when the program runs?
    - *Nothing will happen because the condition for the while loop is never true: 10 is not less than 10.*

```
let counter = 10;
while (counter < 10) {
    console.log(counter);
    counter--;
}
```

End of Class:

- What is the purpose of creating a counter variable outside of a while loop?
    - *You can use a counter variable to keep track of the number of loops, and you can use that to set the condition for when to end the loop.*
- Take a look at the output of a program below that draws 6 randomly colored horizontal stripes. In pseudocode, how might you use a

while loop to produce that output?


```
let counter = 0;
let x = 0;
let y = 0;
while (counter <=6) {
    draw randomly colored rectangle at x, y
    y = y + rectangle height
    counter++
}
```

Resources/Handouts	

Vocabulary

Term	Definition
While Loop	Lets us repeat code as long as something is true.

Modification: Advanced	Modification: Special Education	Modification: English Language Learners
• Virginia Garcia Costa is an Argentine artist who creates artwork that is inspired by nature. Much of her work is geometric and can be modeled using a while loop! Have students take a look at some of her work and have them try and recreate her art digitally.	• Strategically pair students and use pair programming for the *Growing Squares* exercise. • Print out the *While Loops and Graphics* note so that students can reference it throughout the lesson. • Provide students with starter code that breaks down the *Growing Squares* exercise to make it more accessible.	• Print out the *While Loops and Graphics* note so that students can reference it throughout the lesson. • Provide students with a dictionary so they can look up known words.