

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Takeaways

If we were to have more time and more Python experience, we would've implemented error checking. To assure the program does not crash after a logic error by converting a string to an integer improperly.

Link to Video

Description of Video	Link to Video
<ol style="list-style-type: none"> 1. We showcase the game itself 2. We showcase the features of our game 3. We show the completion of the game 4. We some specific code segments of our game Code 	<p><u>Video Link</u></p> <p>If it does not load Please click file → copy to dropbox</p>

Highlighted Code Segment

The reason we chose to contain the game within a main method is the team lead was familiar with Java more than Python and the team wanted to make it easier for him to adapt to Python.

```
def main():
    global score
    global gameOver
    global listOfQuestionNum
    global debugMode
    if (not debugMode):
        while (not gameOver):
            n = 0
            if (score > 16):
                n = rand.randint(11, 16)
            elif (score > 8):
                n = rand.randint(6, 11)
            else:
                n = rand.randint(2, 6)
            print("What is the sum of all the numbers from 1 to", n)
            answer = input()
            game(answer, n)
        else:
            while (not gameOver):
                score = score + 1
                display()
            if (gameOver):
                print("Game Over")
                print("You got", score, "correct")
                sum = 0
                for i in listOfQuestionNum:
                    sum = sum + i
                print(f"You summed up this total {sum}")
```

Purpose and Function of Program

String Code Segment

We manipulate the print line String to explain the problem with respect to the given summation max variable

```
90     print(
91         f"Therefore the , the Sigma Notation of {n} is {n} *
({n} + 1) / 2 = {sum}"
92     )
```

Event Code Segment

Our game didn't use an event per say so we used a looped input request to accept the answer which would determine whether to continue forward or not by using a Function.

```
120     n = 0
121     if (score > 16):
122         n = rand.randint(11, 16)
123     elif (score > 8):
124         n = rand.randint(6, 11)
125     else:
126         n = rand.randint(2, 6)
127     print("What is the sum of all the
numbers from 1 to", n)
128     answer = input()
129     game(answer, n)
```

Function Code Segment

This particular function accepts parameter n and explains how to take the sum of a number. It uses formatted strings to explain how to solve the problem in context.

```
78     # Explaining the why you got the answer wrong
79     def explain(n):
80         print(f"--- Sigma of {n} ---")
81         print(
82             "The Sigma Notation is a way of
writing down the sum of a sequence of
numbers."
83         )
84         sum = int(sigma_formula(n))
85         time.sleep(1)
86         print("The Sigma Notation is written as
follows: ")
87         time.sleep(1)
88         print("Sigma(n) = n * (n + 1) / 2")
89         time.sleep(1)
90         print(
91             f"Therefore the , the Sigma Notation
of {n} is {n} * ({n} + 1) / 2 = {sum}"
92         )
```

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Link to Codebase

Link to Codebase
<u>Link</u>

Industry Connects One 8: 1.2.5 Shall We Play A Game

Pick one or two partners of your own choice. The groups should be no larger than three. I would prefer that no one works alone. If alone, you will be expected to explain how a team approach would benefit your project and investigate all parts of this document.

Team members: Up to three members per team, So nice you see it seven times

```
nameArr = []
nameArr.append("Alex Kovalski")
nameArr.append("Eli Turcotte")
nameArr.append("Jordan Real.")
for step in range(7):
    for name in nameArr:
        print(name)
```

Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.
Alex Kovalski	Eli Turcotte	Jordan Real.

Learning Targets and SWBAT:

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

- Apply all you have learned to plan and create new artwork, a game, or an interactive story of your choosing.
- Explain how a code segment or program functions.
- Identify data storage and usage in your program.
- Explain how an abstraction manages complexity in your program.
- Describe a procedure call in your program.

Project:

Computing for Change, a non-profit organization in your community, plans to send digital interactive cards to various members of your community. You have been hired to design a card for one of their causes that includes one of the following components:

- A kind, encouraging message
- A fun animated image
- An educational interactive animation

Requirements:

In addition to the requirements set up by your project manager, the encouraging message, animated image, or educational animation should use the algorithmic thinking and computational skills you have learned.

Basic functionality:

- ☒ ~~Allow for user input.~~
- ☒ ~~Use color and movement.~~

In your code:

- ☒ ~~Use iteration (looping) and conditional execution (if statements) to control the drawing.~~
- ☒ ~~Manipulate lists or strings.~~
- ☒ ~~Make use of a boolean expression~~
- ☒ ~~Storing data from a list and manipulating that list (might want to allow an input statement)~~
- ☒ ~~Include conditional behavior.~~
- ☒ ~~Respond to events. IS INPUT AN EVENT? WE WENT BASED OFF OF THAT BEING TRUE~~
- ☒ ~~Create functions (to reduce duplication of code when appropriate)~~

To help you in creating your artwork

- ☒ ~~Read the entire project before starting!!!~~
- ☒ ~~Use existing turtle methods.~~
- ☒ ~~Choose descriptive variable names.~~
- ☒ ~~Comment code segments or blocks of statements.~~

Industry Connects Requirement:

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

- ☒ ~~Link to a video of the artistic expression~~
- ☒ ~~Screenshots of code~~
- ☒ ~~Link to a video of the program's output.~~
- ☒ ~~Purpose of the program and explanation of what the video/screenshots illustrate~~
- ☒ ~~Description of one code segment related to strings (i.e., what data do the strings represent and how they manipulate them?)~~
- ☒ ~~Description of one code segment related to events (i.e., what events are recognized in the program and how they are managed?)~~
- ☒ ~~Description of one code segment related to a function that they created (i.e., how is the function used and how does it reduce duplication of code?)~~
- ☒ ~~Link to python (.py) file with code in a Google Drive or dropbox that is shared so that all can view~~

If you need to include links, embed the links in the PDF and double-check that they are accessible to all.

Team rules

Some examples are open communication, mutual respect, shared accountability, clear goals, positive support for teammates, active listening, constructive criticism, timely practice attendance, commitment to improvement, and upholding team values. You can research your own, and make sure you have goals that keep your team members accountable, engaged, appreciated, and their voices are heard.

Give me at least 5 agreed-upon rules you plan to follow on your team:

Norm	The consequence if a norm is broken
Everyone will work together when needed.	No work possibly won't get done.
Everybody has access to the code	Most likely no code to access.
If a group member is absent they will be caught up when they are next in class	They'll be caught up on what happened in their absences
Everyone won't lose their code.	Code will be lost.
Someone is out	Email the person

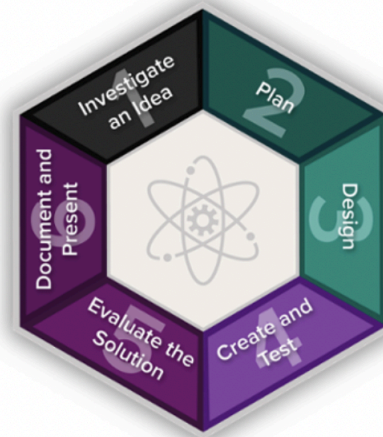
Step 1: Review the software development process

(<https://pltw.read.inkling.com/a/b/f328a8095ccf4e15b482f866496c941d/p/d1507b1dc7ec4799a6393047390916e6>)

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Development Process

- Investigate an Idea
- Plan
- Design
- Create and Test
- Evaluate the Solution
- Document and Present



Step 1: Join a team and investigate the project together

Your artistic artifact should represent the algorithmic thinking and computational skills you have learned thus far. It should be something worthy of a team. What does that mean? Maybe your team liked bugs...then create a moving bug that can use code from before but has to be your own. Maybe you want to do a race? Maybe something like whack a mole (catch a turtle)...maybe another simple game?

- ☒ ~~Make sure you read this entire document, the requirements, and PLTW 1.3.1 before starting!!!~~

Step 2-6: Brainstorming

Brainstorm a few ideas that you would like to create. Avoid going into too much detail, especially how you will create your artifact. You have probably done a fair amount of brainstorming or idea generation throughout your education. Here are some basic “rules” about brainstorming.

Visit to learn about brainstorming techniques and pick one for your team:

<https://pltw.read.inkling.com/a/b/f328a8095ccf4e15b482f866496c941d/p/46f04e11a2f143d1a71bdf3a57dff4ac>

Basic Brainstorming Rules

- Capture all ideas.
- Allow all members to contribute.
- No criticism allowed.
- Work for quantity to allow quality ideas to surface.
- Welcome piling on. Use others' ideas as launch pads.
- Allow a free-for-all.
- Encourage big thinking.

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Be sure to capture all ideas during a brainstorming session. Designate a good note taker who can compile all the notes from the session.

What technique did you use from the Brainstorming document to gather ideas?

Alex had an idea already. And the group developed on it together by making it more cool instead of boring math games.

Show me all your ideas as a team (everyone should have at least two ideas of their own)

Jordan - Suggests possibly to add in debug mode, suggests to work close (Not that close.) To complete this.

Alexander - Wants to create a 'connect the dots' game, Wishes to name it 'sigma'.

Eli - Suggested doing the connect the dots, Suggested using a shark drawing.

Once you are done brainstorming, evaluate your ideas by choosing two of your ideas and elaborate on them with further documentation and rough sketches.

Done on paper

Choose one idea to fully develop and explain your project and how it relates to the requirements

Sigma Game

Step 7-9: Plan Software development projects are often organized into milestones, a list of tasks that need to get done by a certain date. Breaking down your project into smaller tasks requires problem decomposition. Problem decomposition is the process of breaking a complex problem or system into parts that are easier to conceive, understand, program, and maintain. In this part of the incremental development process, you decompose your artistic artifact into smaller features and then define milestones for developing those features.

Decompose (break down) your problem. Identify the major parts of your program. Decide what the various parts will do. These are its features. Remember you have more than one member on your team and you might all want to work on different parts of this project. Make an exhaustive list of what you would like to do (even if you don't feel you have the time to do your project fully realized)

Drawing of a shark
The Sigma answers
Explanations if the input is false

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Methods for game sets

Decide whether each feature is feasible, meaning decide if it is doable. Be sure that all features are feasible and that you will be able to develop them in the time allowed. If necessary, modify your list, moving some features to an optional or “nice to have” list. Remember, you can do iterations. Get the basic functionality going and add to make it more robust.

All these features are feasible and have been worked on already by Alex.

Most features have been tested out already as well.

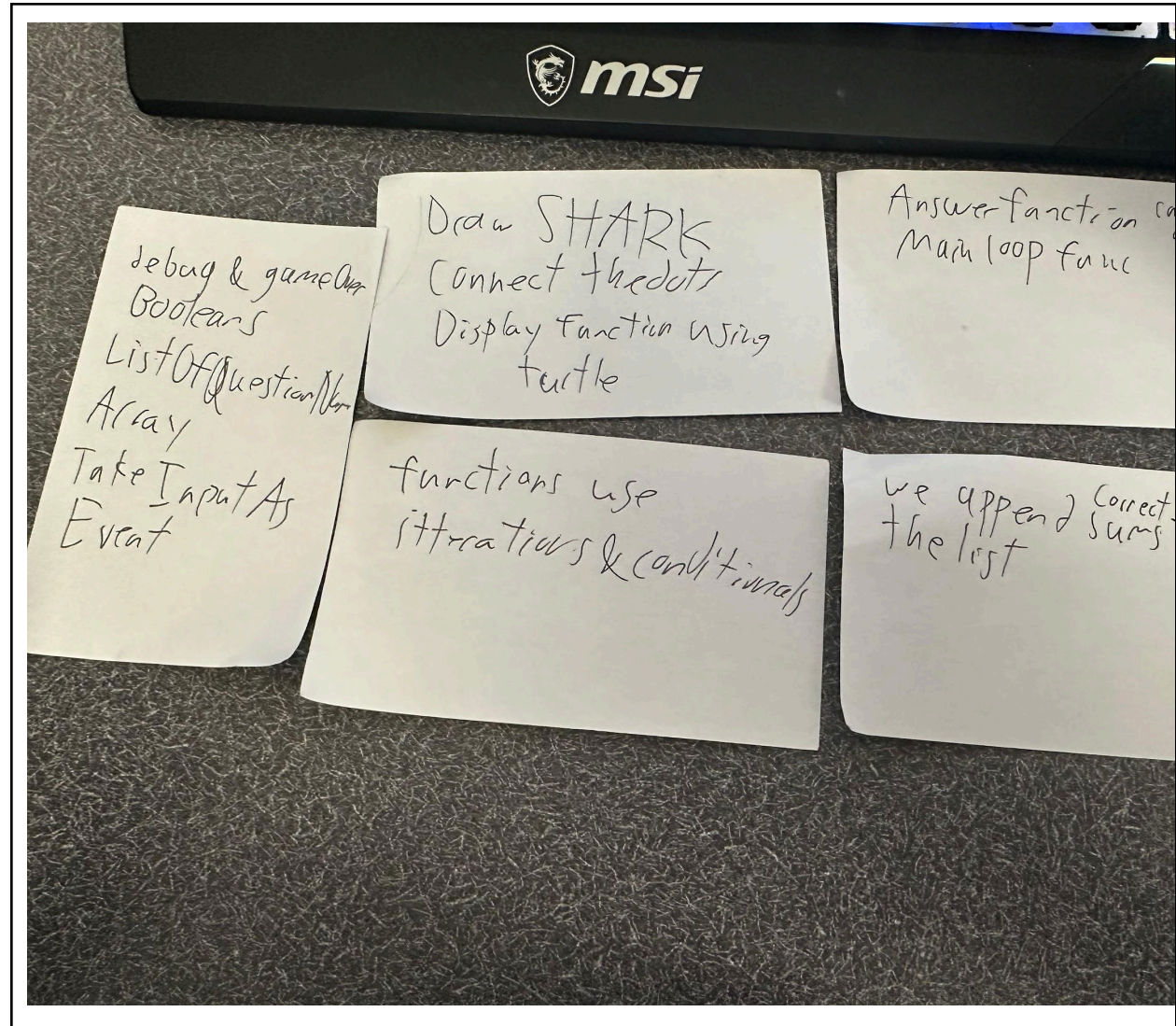
Order your list. Choose what feature to develop first, what to develop last, and what to develop in between. Estimate the time you think it will take to develop each feature. These are your development milestones. Document them and assign them dates. You should have at least one milestone for each day you work on your project.

The Sigma answers
 Explanations if the input is false
 Drawing of a shark
 Methods for game over
 Methods for game sets

Step 10-11: Design Now that you have planned what you want to create, it is time to design its specific features.

Best flowchart: you can sketch out what you want to do and then ask permission to take a photo. You can use the whiteboards or the magnetic whiteboards to solidify your idea.

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------



Revisit your requirements to make sure your project covers all requirements. Use iteration to add features to enhance a fully developed project.

Feature	Components Needed	Which requirement is met	The time it will take
Shark drawing Display	Plug and chug	Turtle methods and conditionals	1 class per
Sigma Answer	Function with parameters	Function with parameters	10 mins
Explanation	Function that explains how to solve when	Function	5 mins

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

	wrong		
main	Method	iteration	30 mins

Write pseudocode for the major features of your program. This should give you a broad idea of how to develop them, but some of the details may not be completely clear.

While the game isnt over:

- If the score is level 3 - gen num 11-16
- If the score is level 2 - gen num 6-11
- If the score is level 1 - gen num 2-6
- Ask the player what the sum of num
- Call game with parameter answer and n

Step 12-17: Create and Test

In this section, you will begin to develop a prototype. Remember to test your prototype throughout the development. This is when you start coding!

Refer to your list of milestones and develop your first feature:

- Use your pseudocode as comments, and if necessary, adjust them as you develop your solution.
- Remember to use your debugging techniques when you run into problems.
- Check that your feature meets the requirement(s) that you documented as part of your project milestones.

Screenshot of code and output Iteration 1

No to screen output at this time

```

112 # Main Loop for executing a game instance
113 def main():
114     global score
115     global gameOver
116     global listOfQuestionNum
117     global debugMode
118     if (not debugMode):
119         while (not gameOver):
120             n = 0
121             if (score > 16):
122                 n = rand.randint(11, 16)
123             elif (score > 8):
124                 n = rand.randint(6, 11)
125             else:
126                 n = rand.randint(2, 6)
127             print("What is the sum of all the numbers from 1 to", n)
128             answer = input()
129             game(answer, n)

```

What is the sum of all the numbers from 1 to 5
5

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Actual Code Iteration 1

```

# Programmers: Alexander P. Kovalski, Elijah A. F. Turcotte, Jordan E. Real
# Date Last Modified: 23JAN2025
# Purpose: Make winning educational project by teaching students Sigma Notation
# Function: Using globals, functions, input, and iteration to create a game that teaches students
how to do sigma notation.

# Imports
import turtle as trtl
import random as rand
import time

# Define Global Vars
debugMode = False
score = 0
gameOver = False
listOfQuestionNum = []
participant = trtl.Turtle()
participant.fillcolor("red")
participant.begin_fill()
participant.penup()
participant.goto(-100, 0)
participant.pendown()
wn = trtl.Screen()

# Display the connect the dots on the screen
def display():
    global gameOver
    global score
    if (score == 1):
        participant.goto(-90, 20)
    elif (score == 2):
        participant.goto(-60, 40)
    elif (score == 3):
        participant.goto(-30, 45)
    elif (score == 4):
        participant.goto(10, 100)
    elif (score == 5):
        participant.goto(0, 50)
    elif (score == 6):
        participant.goto(40, 45)
    elif (score == 7):
        participant.goto(60, 30)
    elif (score == 8):
        participant.goto(90, 10)
    elif (score == 9):
        participant.goto(120, 30)
    elif (score == 10):

```

```
    participant.goto(120, -30)
elif (score == 11):
    participant.goto(90, -10)
elif (score == 12):
    participant.goto(60, -30)
elif (score == 13):
    participant.goto(-30, -35)
elif (score == 14):
    participant.goto(0, -60)
elif (score == 15):
    participant.goto(-45, -35)
elif (score == 16):
    participant.goto(-60, -30)
elif (score == 17):
    participant.goto(-80, -20)
elif (score == 18):
    participant.goto(-100, 0)
    participant.end_fill()
elif (score == 19):
    participant.penup()
    participant.goto(-80, 20)
    participant.pensize(3)
    participant.dot()
    participant.hideturtle()
    gameOver = True
```

```
# Calculate Answer Formula
```

```
def sigma_formula(n):
    sum = n * (n + 1) / 2
    return sum
```

```
# Explaining the why you got the answer wrong
```

```
def explain(n):
    print(f"--- Sigma of {n} ---")
    print(
        "The Sigma Notation is a way of writing down the sum of a sequence of numbers."
    )
    sum = int(sigma_formula(n))
    time.sleep(1)
    print("The Sigma Notation is written as follows: ")
    time.sleep(1)
    print("Sigma(n) = n * (n + 1) / 2")
    time.sleep(1)
    print(
        f"Therefore the , the Sigma Notation of {n} is {n} * ({n} + 1) / 2 = {sum}"
    )
```

```
# A Single execution of a question
```

```
def game(answer, n):
    global gameover
    global score
    global listOfQuestionNum
    answer = int(answer)
    if answer == sigma_formula(n):
        print("Correct")
        score = score + 1
        listOfQuestionNum.append(n)
        display()
    else:
        print("Incorrect")
        explanation = input("Would you like an explanation? (y/n) ")
        if (explanation == "y"):
            explain(n)
        print()

# Main Loop for executing a game instance
def main():
    global score
    global gameOver
    global listOfQuestionNum
    global debugMode
    if (not debugMode):
        while (not gameOver):
            n = 0
            if (score > 16):
                n = rand.randint(11, 16)
            elif (score > 8):
                n = rand.randint(6, 11)
            else:
                n = rand.randint(2, 6)
            print("What is the sum of all the numbers from 1 to", n)
            answer = input()
            game(answer, n)
        else:
            while (not gameOver):
                score = score + 1
                display()
            if (gameOver):
                print("Game Over")
                print("You got", score, "correct")
                sum = 0
                for i in listOfQuestionNum:
                    sum = sum + i
                print(f"You summed up this total {sum}")

# Calling the Main (Loop) Function
```

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

```
main()
wn.mainloop()
```

Test that the feature works and record the milestone as complete in the table above. What did you complete?

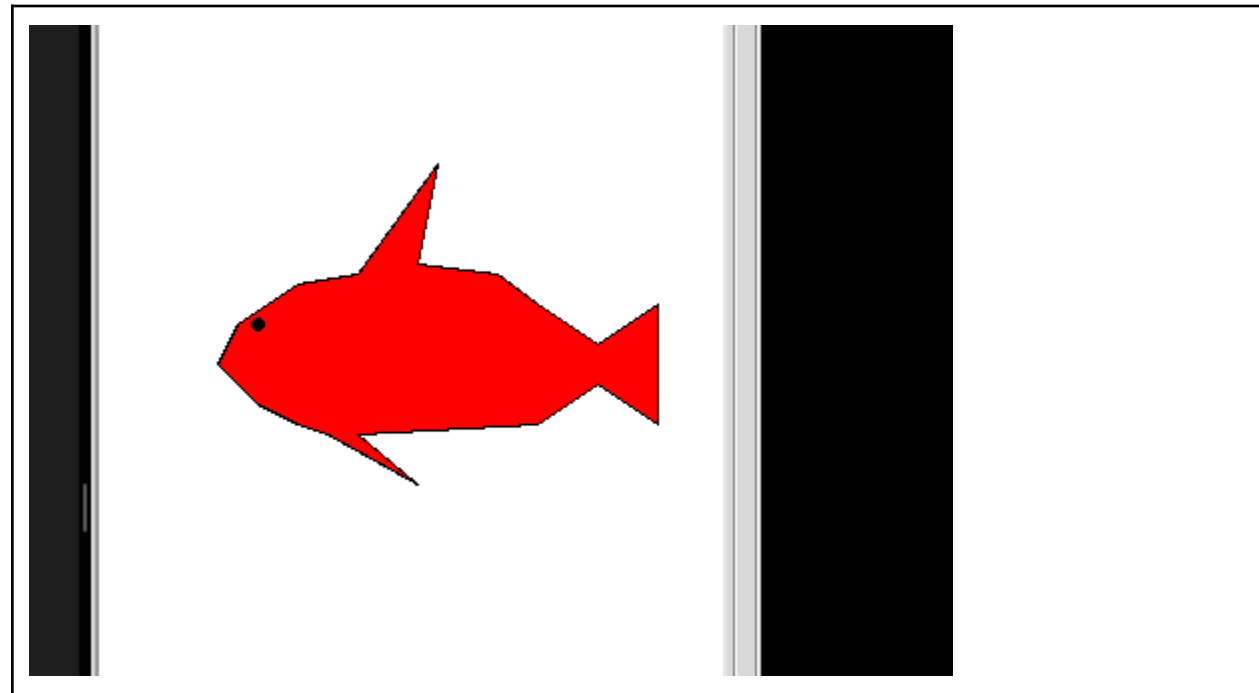
Main | iteration

Make note of any challenges that you faced and how you overcame them or worked around them.

Being absentminded while importing lack there of globals

Repeat the previous steps to develop and test each feature until you have completed your project.

Screenshot of code and output Iteration 2



Actual Code Iteration 2

```
# Programmers: Alexander P. Kovalski, Elijah A. F. Turcotte, Jordan E. Real
# Date Last Modified: 23JAN2025
# Purpose: Make winning educational project by teaching students Sigma Notation
# Function: Using globals, functions, input, and iteration to create a game that teaches students
how to do sigma notation.
```

```
# Imports
import turtle as trtl
import random as rand
import time

# Define Global Vars
debugMode = True
score = 0
gameOver = False
listOfQuestionNum = []
participant = trtl.Turtle()
participant.fillcolor("red")
participant.begin_fill()
participant.penup()
participant.goto(-100, 0)
participant.pendown()
wn = trtl.Screen()

# Display the connect the dots on the screen
def display():
    global gameOver
    global score
    if (score == 1):
        participant.goto(-90, 20)
    elif (score == 2):
        participant.goto(-60, 40)
    elif (score == 3):
        participant.goto(-30, 45)
    elif (score == 4):
        participant.goto(10, 100)
    elif (score == 5):
        participant.goto(0, 50)
    elif (score == 6):
        participant.goto(40, 45)
    elif (score == 7):
        participant.goto(60, 30)
    elif (score == 8):
        participant.goto(90, 10)
    elif (score == 9):
        participant.goto(120, 30)
    elif (score == 10):
        participant.goto(120, -30)
    elif (score == 11):
        participant.goto(90, -10)
    elif (score == 12):
        participant.goto(60, -30)
    elif (score == 13):
        participant.goto(-30, -35)
```



```
elif (score == 14):
    participant.goto(0, -60)
elif (score == 15):
    participant.goto(-45, -35)
elif (score == 16):
    participant.goto(-60, -30)
elif (score == 17):
    participant.goto(-80, -20)
elif (score == 18):
    participant.goto(-100, 0)
    participant.end_fill()
elif (score == 19):
    participant.penup()
    participant.goto(-80, 20)
    participant.pensize(3)
    participant.dot()
    participant.hideturtle()
    gameOver = True

# Calculate Answer Formula
def sigma_formula(n):
    sum = n * (n + 1) / 2
    return sum

# Explaining the why you got the answer wrong
def explain(n):
    print(f"--- Sigma of {n} ---")
    print(
        "The Sigma Notation is a way of writing down the sum of a sequence of numbers."
    )
    sum = int(sigma_formula(n))
    time.sleep(1)
    print("The Sigma Notation is written as follows: ")
    time.sleep(1)
    print("Sigma(n) =  $n * (n + 1) / 2$ ")
    time.sleep(1)
    print(
        f"Therefore the , the Sigma Notation of {n} is {n} * ({n} + 1) / 2 = {sum}"
    )

# A Single execution of a question
def game(answer, n):
    global gameOver
    global score
    global listOfQuestionNum
    answer = int(answer)
    if answer == sigma_formula(n):
        print("Correct")
```

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

```

    score = score + 1
    listOfQuestionNum.append(n)
    display()
else:
    print("Incorrect")
    explanation = input("Would you like an explanation? (y/n) ")
    if (explanation == "y"):
        explain(n)
    print()

# Main Loop for executing a game instance
def main():
    global score
    global gameOver
    global listOfQuestionNum
    global debugMode
    if (not debugMode):
        while (not gameOver):
            n = 0
            if (score > 16):
                n = rand.randint(11, 16)
            elif (score > 8):
                n = rand.randint(6, 11)
            else:
                n = rand.randint(2, 6)
            print("What is the sum of all the numbers from 1 to", n)
            answer = input()
            game(answer, n)
        else:
            while (not gameOver):
                score = score + 1
                display()
            if (gameOver):
                print("Game Over")
                print("You got", score, "correct")
                sum = 0
                for i in listOfQuestionNum:
                    sum = sum + i
                print(f"You summed up this total {sum}")

# Calling the Main (Loop) Function
main()

wn.mainloop()

```

Test that the feature works and record the milestone as complete. What did you complete?

Shark display

Alex Kovalski

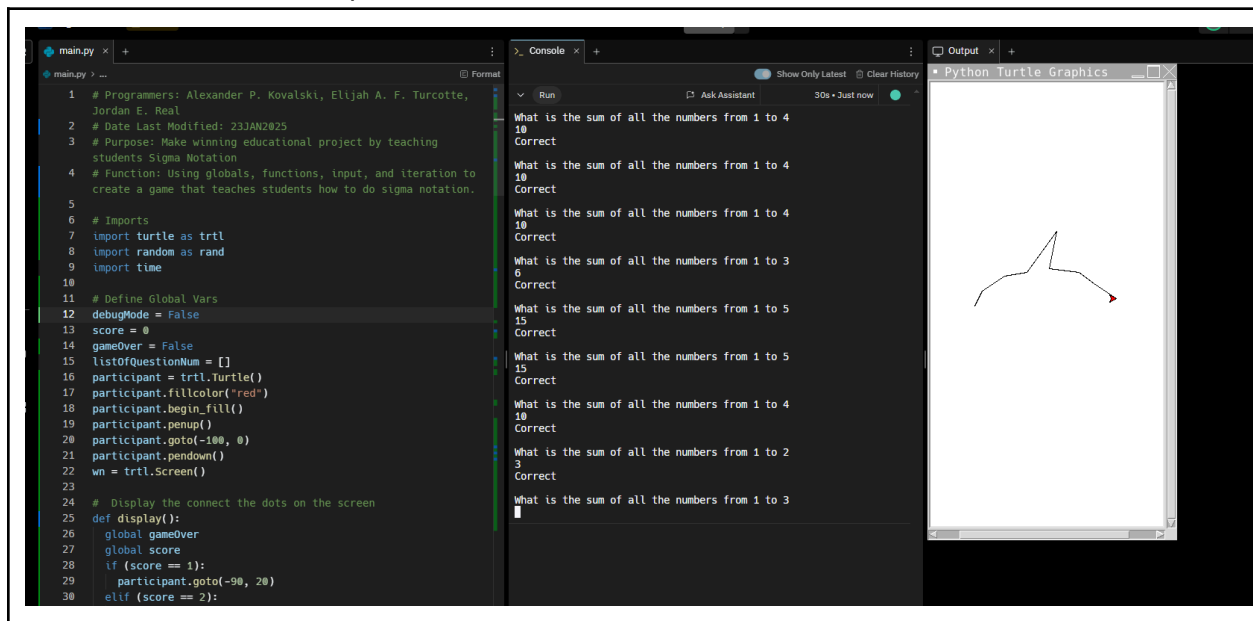
Eli Turcotte

Jordan Real

Make note of any challenges that you faced and how you overcame them or worked around them.

Debugging the display was a pain

Screenshot of code and output final Iteration



Actual Code Final Iteration

```
# Programmers: Alexander P. Kovalski, Elijah A. F. Turcotte, Jordan E. Real
# Date Last Modified: 23JAN2025
# Purpose: Make winning educational project by teaching students Sigma Notation
# Function: Using globals, functions, input, and iteration to create a game that teaches students how to do sigma notation.
```

```
# Imports
import turtle as trtl
import random as rand
import time
```

```
# Define Global Vars
debugMode = False
score = 0
gameOver = False
listOfQuestionNum = []
participant = trtl.Turtle()
participant.fillcolor("red")
participant.begin_fill()
participant.penup()
```

```
participant.goto(-100, 0)
participant.pendown()
wn = turtle.Screen()

# Display the connect the dots on the screen
def display():
    global gameOver
    global score
    if (score == 1):
        participant.goto(-90, 20)
    elif (score == 2):
        participant.goto(-60, 40)
    elif (score == 3):
        participant.goto(-30, 45)
    elif (score == 4):
        participant.goto(10, 100)
    elif (score == 5):
        participant.goto(0, 50)
    elif (score == 6):
        participant.goto(40, 45)
    elif (score == 7):
        participant.goto(60, 30)
    elif (score == 8):
        participant.goto(90, 10)
    elif (score == 9):
        participant.goto(120, 30)
    elif (score == 10):
        participant.goto(120, -30)
    elif (score == 11):
        participant.goto(90, -10)
    elif (score == 12):
        participant.goto(60, -30)
    elif (score == 13):
        participant.goto(-30, -35)
    elif (score == 14):
        participant.goto(0, -60)
    elif (score == 15):
        participant.goto(-45, -35)
    elif (score == 16):
        participant.goto(-60, -30)
    elif (score == 17):
        participant.goto(-80, -20)
    elif (score == 18):
        participant.goto(-100, 0)
        participant.end_fill()
    elif (score == 19):
        participant.penup()
        participant.goto(-80, 20)
```

```
    participant.pensize(3)
    participant.dot()
    participant.hideturtle()
    gameOver = True

# Calculate Answer Formula
def sigma_formula(n):
    sum = n * (n + 1) / 2
    return sum

# Explaining the why you got the answer wrong
def explain(n):
    print(f"--- Sigma of {n} ---")
    print(
        "The Sigma Notation is a way of writing down the sum of a sequence of numbers."
    )
    sum = int(sigma_formula(n))
    time.sleep(1)
    print("The Sigma Notation is written as follows: ")
    time.sleep(1)
    print("Sigma(n) =  $n * (n + 1) / 2$ ")
    time.sleep(1)
    print(
        f"Therefore the , the Sigma Notation of {n} is  $\{n\} * (\{n\} + 1) / 2 = \{sum\}$ "
    )

# A Single execution of a question
def game(answer, n):
    global gameover
    global score
    global listOfQuestionNum
    answer = int(answer)
    if answer == sigma_formula(n):
        print("Correct")
        score = score + 1
        listOfQuestionNum.append(n)
        display()
    else:
        print("Incorrect")
        explanation = input("Would you like an explanation? (y/n) ")
        if (explanation == "y"):
            explain(n)
        print()

# Main Loop for executing a game instance
def main():
    global score
    global gameOver
```

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

```

global listOfQuestionNum
global debugMode
if (not debugMode):
    while (not gameOver):
        n = 0
        if (score > 16):
            n = rand.randint(11, 16)
        elif (score > 8):
            n = rand.randint(6, 11)
        else:
            n = rand.randint(2, 6)
        print("What is the sum of all the numbers from 1 to", n)
        answer = input()
        game(answer, n)
    else:
        while (not gameOver):
            score = score + 1
            display()
        if (gameOver):
            print("Game Over")
            print("You got", score, "correct")
            sum = 0
            for i in listOfQuestionNum:
                sum = sum + i
            print(f"You summed up this total {sum}")

# Calling the Main (Loop) Function
main()

wn.mainloop()

```

Test that all the features work and record the milestone as complete when done. What did you complete?

Check

Make note of any challenges that you faced and how you overcame them or worked around them.

Making it challenging without being impossible

Make note of any features you did not have time to develop and any milestones you did not reach.

Completed everything that was intended.

☒ ~~You will need to make a video in the last section~~

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Step 18: Evaluate the Solution: Share your artistic artifact with a classmate.

Can you read and understand your classmate's program?

Yes

Is it commented/well-named and easy to follow? How?

Yes

Is it well-structured and logical?

Yes although it could be better due to lack of time

Step 20-21: Document and Present

- ☒ ~~Review the program requirements for this project to ensure your solution meets all of the requirements.~~
- ☒ ~~Make sure the document is filled out and ALL team members contributed.~~

What did each member of the team contribute to?

Alex Sigma
Eli Connect the Dots
Jordan Code Review

Create a video demonstrating your program's execution and functionality. Your video can contain comments in your code. We need to produce a good-quality video.

Complete and attached in GCR. Please review so we can adjust if needed

Take a screenshot of a procedure that contributes to achieving the intended purpose of your program and a screenshot of the part of your program that calls the procedure. Ensure your screenshots have no comments.

```

74  def sigma_formula(n):
75      sum = n * (n + 1) / 2
76      return sum

```

Take a screenshot of a code segment showing how you store data in your program and a screenshot that shows how that data is used. Ensure your screenshots have no comments.

```
score = 0
```

```
global score
if (score == 1):
    participant.goto(-90, 20)
elif (score == 2):
    participant.goto(-60, 40)
elif (score == 3):
    participant.goto(-30, 45)
elif (score == 4):
```

Purpose of the program and explanation of what the video/screenshots illustrate

Make a winning educational project by teaching students Sigma Notation.

Describe one code segment related to selection (if statement). Selection determines which parts of the program are executed based on a condition being true or false.

```
global score
if (score == 1):
    participant.goto(-90, 20)
elif (score == 2):
    participant.goto(-60, 40)
elif (score == 3):
    participant.goto(-30, 45)
elif (score == 4):
```

This code segment determines which point the turtle goes to complete the drawing

Describe one code segment related to iteration.

```
while (not gameOver):
    n = 0
    if (score > 16):
        n = rand.randint(11, 16)
    elif (score > 8):
        n = rand.randint(6, 11)
    else:
        n = rand.randint(2, 6)
    print("What is the sum of all the numbers from 1 to", n)
    answer = input()
    game(answer, n)
```


Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Description of one code segment related to strings (i.e., what data do the strings represent and how they manipulate them?)

We manipulate the print line String to explain the problem with respect to the given summation max variable

```

90     print(
91         f"Therefore the , the Sigma Notation of {n} is {n} *
92         ({n} + 1) / 2 = {sum}"
93     )

```

Description of one code segment related to events (i.e., what events are recognized in the program and how they are managed?)

The main event we use is an input event which determines whether the answer is right or wrong

Description of one code segment related to a function that they created (i.e., how is the function used and how does it reduce duplication of code?)

This particular function accepts parameter n and explains how to take the sum of a number. It uses formatted strings to explain how to solve the problem in context.

Development Process

- Investigate an Idea
- Plan
- Design
- Create and Test
- Evaluate the Solution
- Document and Present



Each team member should reflect on the design process. How did it help you in your development process?

Alex	Eli	Jordan	Together
I designed and created the main file structure and methods	I was the one who handled the document and the presentation. I was reminding my teammates what needed to go into the document and how the video should look.	I helped with the development process by just reviewing the code throughout the process while evaluating by making sure the algorithms were proper.	We created a game that teaches students how to use sigma notation

Each team member should reflect on your collaboration during the entire development process. What are some strategies that worked well and what are some that can be improved?

Alex Kovalski	Eli Turcotte	Jordan Real
---------------	--------------	-------------

Good	Bad
We brought together our ideas very well Sharing code and responsibilities within Replit was very useful as we could all do specific roles within our team	With our time we didn't get to expand on our ideas and make multiple drawings Finding a better solution to Replit as you can only have a single collaborator at a time other than the owner

What questions would you want to ask the industry professionals who are kindly reviewing your project? At least one question per team member, please!

Alex - How do you recommend we improve the game to be more user friendly?
Eli - How do you make the math easier for the student to comprehend to complete the game?
Jordan - How do you make it easier to follow the development process/life cycle?