Sprint #0 Report

Instructions

CHECK GITHUB FOR BETTER REPORT
https://github.com/Jordinaa/cs449/tree/main/sprint0

Objectives

- Make decisions on the SOS software development project.
- Learn unit testing and GUI programming in the language of your choice.

Deliverables and Grading Policy

Read the "CS 449 Homework Overview" document **carefully** and make the key decisions for the software development. Use the following template to complete your report.

1. **Key Decisions of the SOS Project (2 points)**

Object-oriented programming language	Python
GUI library (strongly encouraged)	Tkinter
IDE (Integrated Development Environment)	VS Code
xUnit framework (e.g., JUnit for Java)	PyTest
Programming style guide (must read it carefully)	PEP 8
Project hosting site	Github - https://github.com/Jordinaa/cs449
Other decisions if applicable	None at the moment

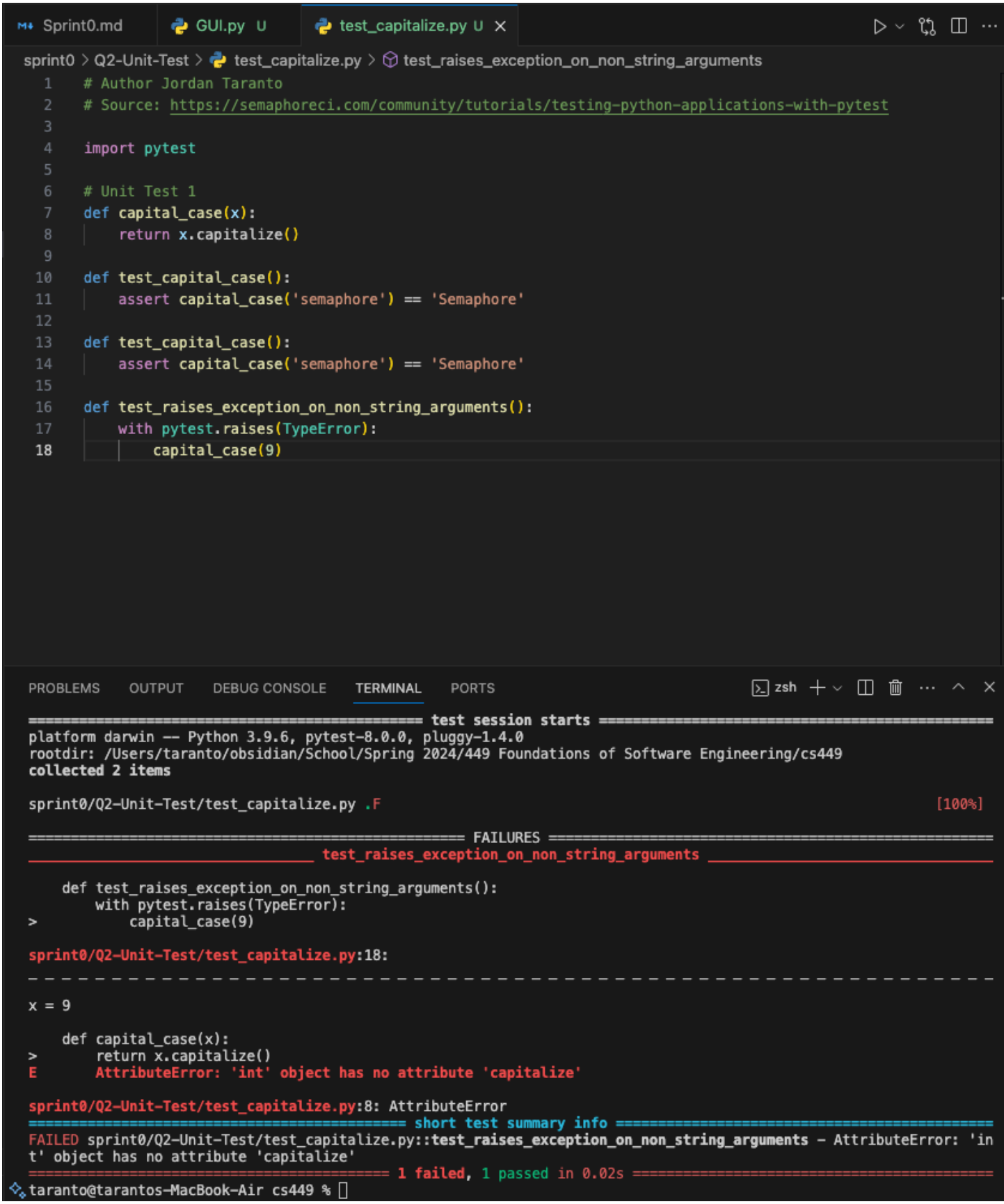
Sample programming style guides:

- Google Java Style Guide: https://google.github.io/styleguide/javaguide.html
- Google C++ Style Guide: https://google.github.io/styleguide/cppguide.html
- Google Python Style Guide: https://google.github.io/styleguide/pyguide.html

2. **Unit testing (4 points)**

Find a tutorial on the unit test framework you have chosen and write at least two xUnit tests of a program you have written or found elsewhere. Attach here (1) the screenshot of your program execution and (2) the source code of your program.

Failed Test


Sprint0.md
GUI.py U
test_capitalize.py U
sprint0 > Q2-Unit-Test > test_capitalize.py > test_raises_exception_on_non_string_arguments
1 # Author Jordan Taranto
2 # Source: https://semaphoreci.com/community/tutorials/testing-python-applications-with-pytest
3
4 import pytest
5
6 # Unit Test 1
7 def capital_case(x):
8 return x.capitalize()
9
10 def test_capital_case():
11 assert capital_case('semaphore') == 'Semaphore'
12
13 def test_capital_case():
14 assert capital_case('semaphore') == 'Semaphore'
15
16 def test_raises_exception_on_non_string_arguments():
17 with pytest.raises(TypeError):
18 capital_case(9)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
zsh
test session starts
platform darwin -- Python 3.9.6, pytest-8.0.0, pluggy-1.4.0
rootdir: /Users/taranto/obsidian/School/Spring 2024/449 Foundations of Software Engineering/cs449
collected 2 items
sprint0/Q2-Unit-Test/test_capitalize.py .F [100%]
FAILURES
test_raises_exception_on_non_string_arguments
def test_raises_exception_on_non_string_arguments():
with pytest.raises(TypeError):
> capital_case(9)
sprint0/Q2-Unit-Test/test_capitalize.py:18:
x = 9
def capital_case(x):
> return x.capitalize()
E AttributeError: 'int' object has no attribute 'capitalize'
sprint0/Q2-Unit-Test/test_capitalize.py:8: AttributeError
short test summary info
FAILED sprint0/Q2-Unit-Test/test_capitalize.py::test_raises_exception_on_non_string_arguments - AttributeError: 'int' object has no attribute 'capitalize'
1 failed, 1 passed in 0.02s
taranto@tarantos-MacBook-Air cs449 %

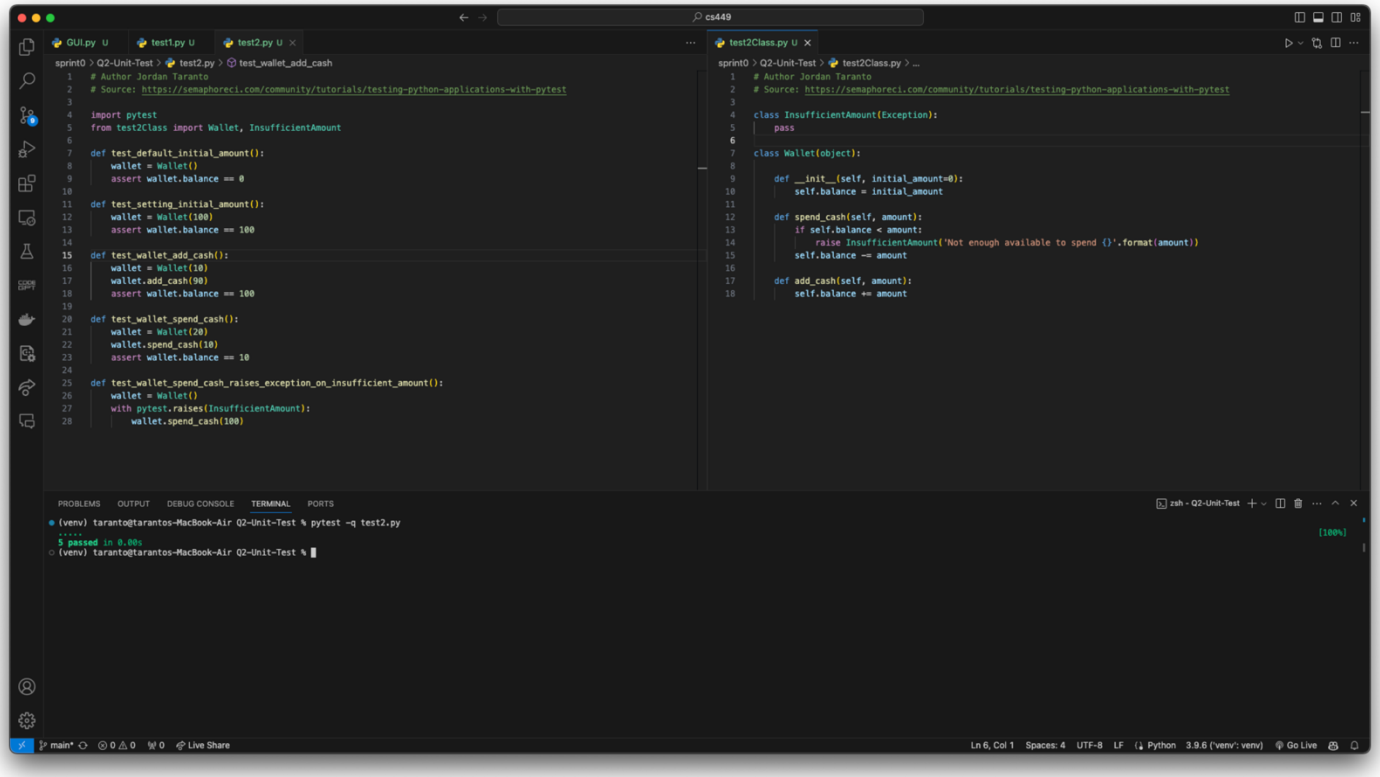
Sprint0.md | GUI.py U | test_capitalize.py U

sprint0 > Q2-Unit-Test > test_capitalize.py > ...

```python
# Author Jordan Taranto
# Source: https://semaphoreci.com/community/tutorials/testing-python-applications-with-pytest

import pytest

# Unit Test 1

# Rewrote Function because error was thrown back - 'int' object has no attribute 'capitalize'
# def capital_case(x):
#     return x.capitalize()

def capital_case(x):
    if not isinstance(x, str):
        raise TypeError('Please provide a string argument')
    return x.capitalize()

def test_capital_case():
    assert capital_case('semaphore') == 'Semaphore'

def test_capital_case():
    assert capital_case('semaphore') == 'Semaphore'

def test_raises_exception_on_non_string_arguments():
    with pytest.raises(TypeError):
        capital_case(9)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS zsh

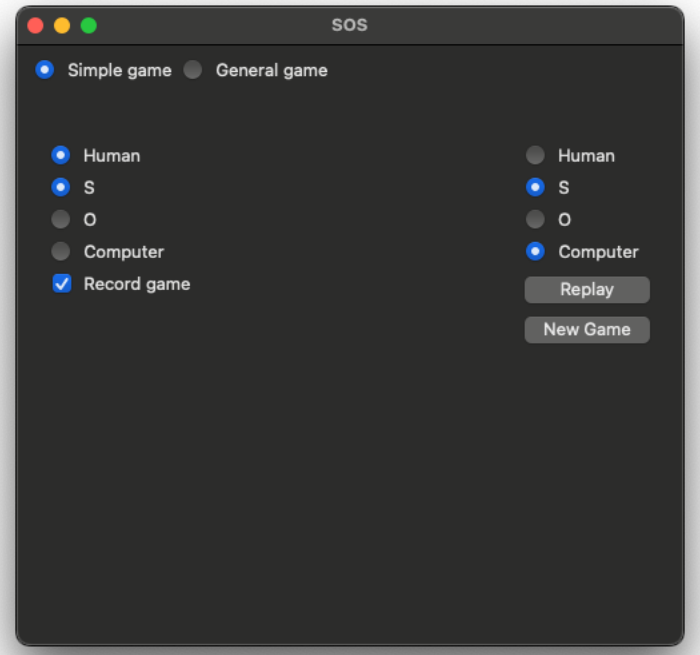
```
============================= test session starts ==============================
platform darwin -- Python 3.9.6, pytest-8.0.0, pluggy-1.4.0
rootdir: /Users/taranto/obsidian/School/Spring 2024/449 Foundations of Software Engineering/cs449
collected 2 items

sprint0/Q2-Unit-Test/test_capitalize.py ..                               [100%]

============================== 2 passed in 0.01s ===============================
```



3. GUI programming (4 points)

Write a GUI program in the language you have chosen for your SOS project. The GUI of your program must include text, lines, a check box, and radio buttons. While you are recommended to consider the GUI for the SOS game board, it is not required. In this assignment, any GUI program of your own work is acceptable.

Attach here (1) the screenshot of your program execution and (2) the source code of your program.


SOS
Simple game
General game
Human
S
O
Computer
Record game
Human
S
O
Computer
Replay
New Game

```python
# Sprint-0
# Author: Jordan Taranto
# Reference: https://medium.com/@fareedkhandev/modern-gui-using-tkinter-12da0b983e22

import tkinter as tk

# function for new game
def start_new_game():
    pass

# function for replay game
def replay_game():
    pass

# root window and size
root = tk.Tk()
root.geometry("500x450")
root.title("SOS")

# board size
board_size = tk.IntVar(value=8)
# record game
record_game = tk.BooleanVar()

# board
top_frame = tk.Frame(root)
top_frame.pack(side="top", fill="x", padx=10, pady=5)

# game type at the top
game_type_frame = tk.Frame(top_frame)
game_type_frame.pack(side="left", fill="x", expand=True)
tk.Radiobutton(game_type_frame, text="Simple game", value="simple").pack(side="left")
tk.Radiobutton(game_type_frame, text="General game", value="general").pack(side="left")

# board size
board_size_frame = tk.Frame(top_frame)
board_size_frame.pack(side="right", fill="x")
tk.Label(board_size_frame, text="Board size").pack(side="left")
tk.Entry(board_size_frame, textvariable=board_size, width=3).pack(side="left")

# blue player and radio buttons
blue_player_frame = tk.LabelFrame(root, text="Blue", padx=10, pady=10)
blue_player_frame.pack(side="left", fill="y", padx=10, pady=5)
blue_player_type = tk.StringVar(value="human")
blue_player_letter = tk.StringVar(value="S")
tk.Radiobutton(blue_player_frame, text="Human", variable=blue_player_type,
value="human").pack(anchor="w")
tk.Radiobutton(blue_player_frame, text="S", variable=blue_player_letter,
value="S").pack(anchor="w")
tk.Radiobutton(blue_player_frame, text="O", variable=blue_player_letter,
value="O").pack(anchor="w")
tk.Radiobutton(blue_player_frame, text="Computer", variable=blue_player_type,
value="computer").pack(anchor="w")
tk.Checkbutton(blue_player_frame, text="Record game", variable=record_game).pack(anchor="w")

# red player and radio buttons
red_player_frame = tk.LabelFrame(root, text="Red", padx=10, pady=10)
red_player_frame.pack(side="right", fill="y", padx=10, pady=5)
red_player_type = tk.StringVar(value="human")
red_player_letter = tk.StringVar(value="S")
tk.Radiobutton(red_player_frame, text="Human", variable=red_player_type,
value="human").pack(anchor="w")
tk.Radiobutton(red_player_frame, text="S", variable=red_player_letter,
value="S").pack(anchor="w")
tk.Radiobutton(red_player_frame, text="O", variable=red_player_letter,
value="O").pack(anchor="w")
tk.Radiobutton(red_player_frame, text="Computer", variable=red_player_type,
value="computer").pack(anchor="w")

# replay and new game buttons
tk.Button(red_player_frame, text="Replay", command=replay_game).pack(fill="x", pady=2)
tk.Button(red_player_frame, text="New Game", command=start_new_game).pack(fill="x")

# main loop
root.mainloop()
```