CS65: Introduction to Computer Science

Final Project
Graphics and Animation

Md Alimoor Reza
Assistant Professor of Computer Science

Final Project

For the final project, you will develop a program of your choosing that incorporates many different programming concepts you have learned throughout the semester.

The final project is to be completed individually

Grading and requirements:

- *Homework Assignments (20%).*
  - homework Python programming activities (3-4 in total).
- *Programming Labs (35%).*
  - A series of short exercises accompanying nearly each class topic. Relatively easier than homework assignments (10-12 in total).
- *Content Quizzes (30%).*
  - "paper-and-pencil exams"quizzes based on the lecture contents. (4-6 in total)
- *Final project (10%).*
  - Individual proposed project. You also need to prepare a powerpoint presentation by the end of semester.
- *Attendance (5%).*
  - Counted based on your signature.

Final Project Proposal Requirements

To receive full credit on the final project you will need to use at least 12 of the following 20 elements:

- variables
- comments
- if-else statement
- Getting input from the user
- Printing output to the console
- One of the following operators //, %, +=, or -=
- Nested if-statement
- While loop
- For loop
- a graphics element
- a random number
- at least 1 user-defined function
- Keyword argument
- Default parameter
- docstring for each function
- List
- Nested List (2D list)
- Reading from a file
- Dictionary
- A Class you define

Final Project Proposal

Due before class on **Tuesday, December 16**

It will be counted as Lab #9

The final project in CS 65 provides opportunity for you to develop a Python program of your own choosing.

The proposal requires a short (a few sentences) written description for your final project

Final Project Proposal

Your project proposal should include the following information:

1. The idea (the game, simulation, visualization, etc. you plan to implement for your final project)
2. A development plan
    a. What will you do first, second, third, etc.
    b. What functions will you use or develop?

Final Project Submission

Your final project must include the following:

- Source Code
- A proposed grade (out of 100)
- A list of the items (12 out of 20) that you implemented
- A 2-minute (approximate) video tour of your code and demo

Final Project Proposal: Example 1

Idea: Sorting Algorithm Visualizer

For our project, we plan to create a sorting algorithm visualizer. We intend to use Python's built-in GUI library t-kinter to achieve this. T-kinter will not create the most visually pleasing software however it will create a very functional and fast system that allows for a lot of modifiability in the parameters, options, and display of the algorithms.

The algorithms we intend to use for sorting are bubble sort, selection sort, insertion sort, merge sort, quick sort, random quick sort, counting sort, and radix sort.

We intend to use rectangles to represent the magnitude of a value in the list and then visually display which boxes are being interacted with and where they are being moved to. We intend to simply swap places of the boxes instead of implementing animations as we want to focus on learning how to create GUIs and understand the sorting algorithms behind them.

Development Plan:

1. Learn for ourselves how each sort functions and what it may be used for
2. Figure out the order in which these sorts will be written/done
3. Write the functions for each individual sort
4. Write the main function where the order will be set and/or the user will be able to choose which sort to execute
5. Run main function
6. See visualization of each sort algorithm by run through

- Functions being used or developed: defined, input, int, main(), for loop, nested for loop, while loop, lists, classes, Python libraries

This is a sort of example based off a more complex idea of what we want to do: https://visualgo.net/en/sorting

Drake UNIVERSITY

Final Project Proposal: Example 2

Dataset: US Unemployment Dataset (2010 - 2020) | Kaggle

Project Idea & Summary:

In this project, we will be taking a look at a dataset containing 10 years of unemployment data from the US. The variables included in this dataset are year and month the data were recorded, the unemployment rate based on the qualification of education of adults at that time (primary school, high school, associates' degree, and professional degree), as well as demographic data looking at the race and gender of the people in the US who were observed in the data. This dataset also contains state-wide unemployment rate data for the year 2020.

Specifically, we will be looking at the trends in unemployment across education qualifications over the 10-year period, and comparing the trends to the more recent effects on US unemployment from the COVID-19 pandemic. By performing such analysis, we will be able to better understand how the COVID-19 pandemic has impacted the unemployment rate since 2020, by viewing the trends across race and education qualification since 2010.

With Python, we will be using data visualization functions to create charts and graphs to visualize the dataset and compare trends across the 10-year period. By using such packages in Python, we can create a dashboard that provides our audience with visualizations to better understand the dataset and the conclusions we have arrived at from performing the analysis. Our Python code will also use functions learned in class to help aggregate the data, such as if/elif functions, to compare variables in the dataset and make understandable conclusions.

Final Project Proposal: Example 3

Idea: Ride the Bus / Irish Poker

Ride the bus or as some others know it Irish poker is a game that is played with more than one person. Note, this does not have to be played as a drinking game, that is just most commonly how it is played.

Step 1) Round 1: The game starts with one of the players guessing if the card is going to be black or red, if the guess is wrong the player has to take a sip of his drink and starts over again, if the guess is correct the user does not have to take a sip and moves on to step 2. Note that all the players have to select their choice before the output is revealed.

Step 2) Round 2: For the second part of the game the code will prompt the users with a question of "higher" or "lower". Depending on the user choice the user will be prompted to take a sip and restart at round 1, or move to the next round. For example, if the user selects "higher" and the card has a higher value than the card that we previously pulled the user will pass on without taking a sip, and if the card has a lower value the user takes a sip and restarts.

Step 3) Round 3: In the third round, the player will input "between" or "outside", to guess whether the next card they flip will be within the range or outside the range of the two cards from the first two rounds. If they guess correctly they move on to the final round, if not they restart at round 1.

Step 4) Round 4: In the final round, the player will input "Spade" or "Club" or "Heart" or "Diamond", guessing what the suit of the final card will be. If the suit of the flipped card is correctly guessed, the win the game, if not they restart at round 1.

Datasets We Will Use: For our final project we will use def, while, for, list, random, dictionary.

Final Project WARNING

Final projects that have any element of plagiarism will be awarded 0 points

You are free (encouraged even) to look at external sources of information

Just CITE, CITE, CITE

Put the webpage you used in comments

You will be assessed on the original code you add to the project

If you use CHAT GPT or DEEPSEEK, AI Assisted Learning Reflection

Quick Review

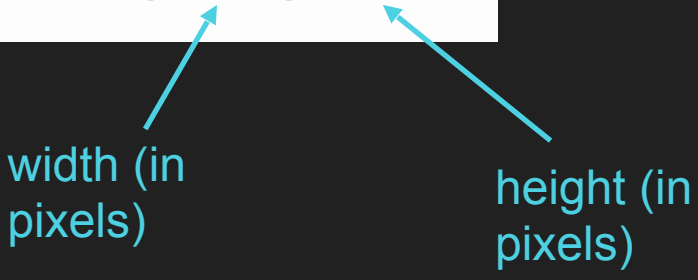
Creating a GraphWin Window

You start using the Graphics library by creating a window and specifying the width and height (in pixels)

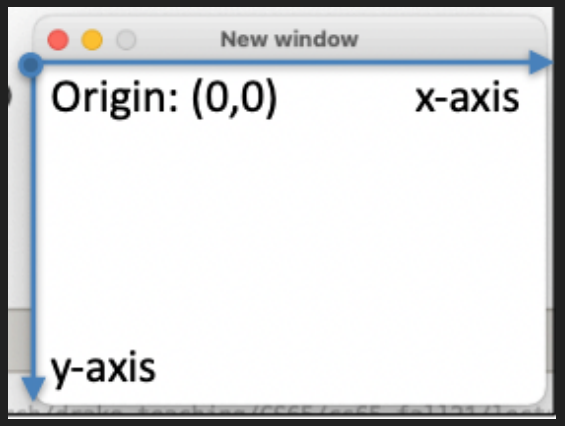
The **`win`** variable will be used throughout the rest of the code to reference the window (or “canvas”) on which we will be drawing things.

```
from graphics import *

# create a window
win = GraphWin("Hello World", 640, 480)
```



In most window-based graphics systems, the origin (0,0) is the upper-left corner of the window



Moving things in the +x direction => moves

Moving things in the +y direction => moves

Extra Challenges

try out the Polygon method--can you draw a triangle?

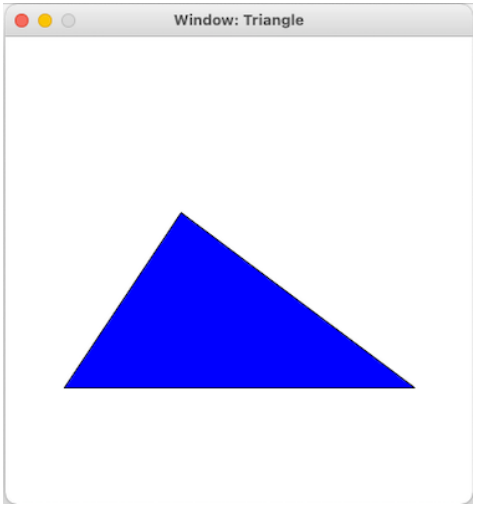
Draw something else cool--can you draw a scene? Maybe a dog, robot, beach, house, etc?

Extra Challenges

- Write a function that draws a **Triangle** based on
  - challenge 1: find out what how many variables do you need to draw it?
  - challenge 2: then receive that many user inputs

- Hints: read the specification below and try to figure out what might be a useful graphical object for this task

  https://mcsp.wartburg.edu/zelle/python/graphics/graphics/graphref.html



Interaction with the user using Mouse

- GraphWin() has a function that allows us to identify the location of your mouse-click
  - 2D coordinate
  - represented by Point object

```
from graphics import *


window = GraphWin("Mouse interaction", 400, 400)

mouse_point1 = window.getMouse()
mouse_point2 = window.getMouse()

print(mouse_point1)
print(mouse_point2)
```

Random Numbers

We can incorporate random numbers into our program by

```
import random
```

At the top of our code

Then, we can use a function such randint as follows:

```
# generate a random integer between 1 and 100 (including 1 and 100)
number = random.randint(1,100)
```

Computer Colors

Each pixel (picture element) has three components

- Red
- Green
- Blue
  - Values for each component range between 0 and 255
  - red = 0, green=0, blue = 0 results in **black**
  - red = 255, green=255, blue = 255 results in **white**
  - red = 255, green=0, blue = 255 results in **purple**

Random Colors

We can create a “random” color by selecting three different numbers to make the color like this:

```
# a random color
r = random.randint(0,255)
g = random.randint(0,255)
b = random.randint(0,255)

color = color_rgb(r,g,b)
```

Random Locations

Similarly, we can select a random x and y values and a random radius for a circle by using **`randint`**:

```
# a random location
x = random.randint(0,350)
y = random.randint(0,350)

# random radius
radius = random.randint(10,50)
```

Classes and Objects

Object Oriented Programming (OOP) is centered on creating *Objects*

Object: a ***combination*** of data and associated procedures

- Data components are called attributes or fields
- Functions are called methods

A ***class*** is the blueprint for creating an object

- A class defines the fields and methods that are associated with an object

These concepts lead to easily developing code that can be **reusable**

Object Oriented Programming – Main Ideas

A class creates a blueprint

- Many objects can be created from a class

The `__init__` function is automatically called when an object is created

- Called a *initializer* method
- It gives initial values to the attributes for the class

`self` parameter: required in every method in the class – references the specific object that the method is working on

Storing Classes in Modules

Classes can be stored in modules

- Filename for module must end in .py
- Module can be imported to programs that use the class

“Driver” class will need to `import` Class module

Example:

```
from Person import *
```

Ball Demo

Exercise #1

When the user presses the space bar, create a new ball object, add it to the ball_list, and draw it. The result should be another ball bouncing around the screen

Hint: look at lines 13-15

Exercise #1

When the user presses the space bar, create a new ball object, add it to the ball_list, and draw it. The result should be another ball bouncing around the screen

Hint: look at lines 13-15

Exercise #2

Make the initial location of the ball a random value

Exercise #3

The Ball class has a **`change_color`** method. Implement it so that the ball's color will change when it is called.

Then, add the code in the Driver (Example1.py) so that whenever a ball bounces, it also changes color

Exercise #4

Add the functionality so the ball also increases speed after it bounces. Create a new method called `increase_speed` in the Ball class.