

ASSIGNMENT OVERVIEW

In this assignment you'll be working on a computer game of your own choosing, subject to the approval of the instructor.

This assignment is worth a total of 50 points and is due during your last week of class.

BACKGROUND

For as long as there have been computers, there have been computer games. Whether allowing two people to play against each other, one person to play against the computer, or even one person just trying to play a solo game, playing computer games is fun. You might find that *programming* computer games is even *more* fun.

ASSIGNMENT SPECIFICATION

Write a computer-based project that allows one or two people to play a game on the computer.

General Possibilities

You've got your choice of a few different game categories. One of the things to consider is how the game will be delivered.

a. Computer Game - Graphics Based

The Processing platform has support for Python programming. If you choose this option, you'll write a graphics-based computer game: Craps? Snake? Tug of War? Poker? Flappy Bird? Lunar Lander?

See screenshots at the end of this document for visual examples of what these programs might look like.

b. Computer Game - Text Based

There are lots of games that can be programmed without having to resort to developing graphics: Rogue? Mastermind? Poker? Use Python to create a text-based game of your own design.

c. Game Suggestions

You are strongly urged to consider working on one of these games. They are reasonable for most people to complete within a few days of class, and if there are other people working on the same game in class, you'll have people to confer with as you develop your own version of the game.

- i. Hangman (text-based game)
- ii. Mugwumps (text-based game)
- iii. Snake (Processing-based game)
- iv. Lunar Lander (Processing-based game)

You may consider other options as well, subject to the approval of the instructor.

DELIVERABLES

`game_development1.jpg` (see description below), and `independent_project.zip`, a single compressed file including your game files and a `README.txt` file contained in a directory called `independent_project`.

You should keep a copy of your development files in your home folder on the server—see additional comments on backup strategies below. To submit your assignment for grading, zip your deliverables as specified above and upload it to the directory `/home/userID/forInstructor` at

crashwhite.polytechnic.org before the deadline.

ASSIGNMENT NOTES

- No outside references may be used for this assignment (reference programs online, Stack Overflow, etc.). This is an opportunity for you to perform your own development of a game program.
- Because the project will be challenging, you will almost certainly find yourself having to learn new things, and this often relies on learning from other people's coding projects or programs. As with the rest of this class, it is permissible to use coding strategies, techniques, or structures that you learn from inspecting other's work. DO read others' code, learn from it, and adapt it for your own purposes. Do NOT copy code verbatim from someone else's program.
- If you have any code that you yourself have already written—for this class, for another, or on your own—you are welcome to use that code as a base for your project. Make sure that you indicate what codebase you are starting with, and what you plan to add on top of that code base that makes it appropriate for an independent project in here.
- Now is an especially important time to consider your digital backup strategy. You should have:
 - Three copies of your data
 - Two of these copies on different types of media (magnetic disk, CD/DVD, flash drive, or hard copy)
 - One copy offsite (on the course server, for example)Thus, if you have one copy of important files on your laptop, one copy on a USB drive (that you regularly update), and one copy on the our server, in Google Drive, DropBox, Backblaze, or on GitHub (offsite), you'll be covered.
- For program-based projects, you can expect that there will be good days and bad days in terms of the progress you make. It's not uncommon for several days worth of work to be discarded after trying an approach to a problem that ultimately doesn't work—some of you have experienced this fact already. Be patient with yourself, and leave some time for unforeseen circumstances.

GETTING STARTED

1. *Before you start coding*, spend some time reflecting on the game you're about to write, and how you might think you'll need to organize your program. Talk to others, talk to the instructor... and write down your initial ideas on paper or on a whiteboard. Take a picture of that document and upload it to the server as **game_development1.jpg**.
2. In some cases the instructor may provide you with documents, game descriptions, and consultation to help you with your project.
3. Projects often take longer than you think. Don't start late on this one.
4. Don't forget to include a README.txt file. Information on this file is available on the course website.

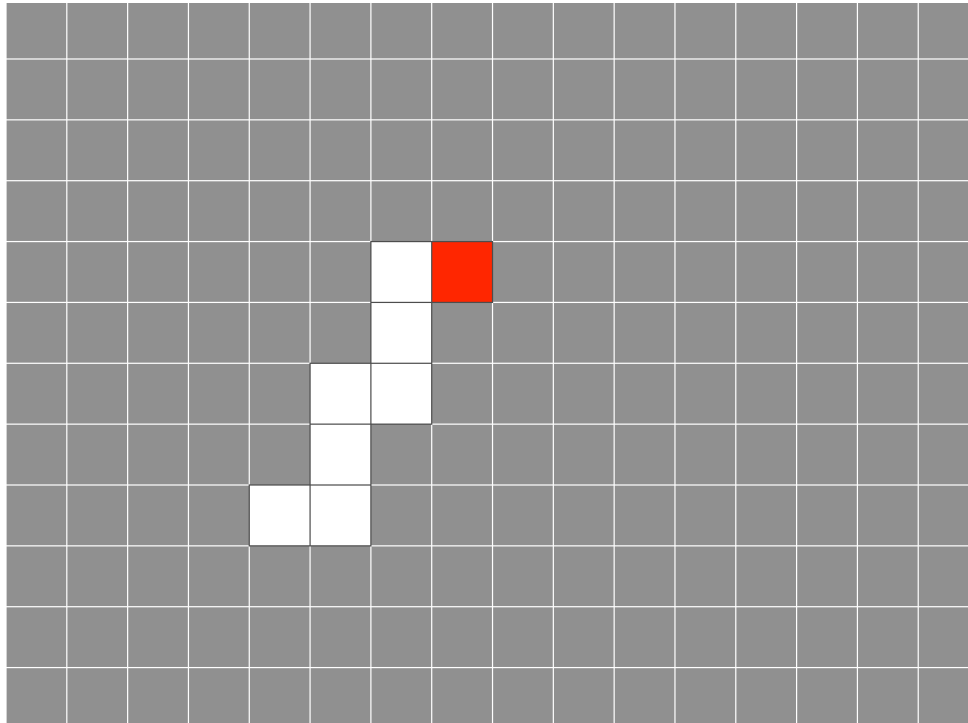
QUESTIONS FOR YOU TO CONSIDER (NOT HAND IN)

1. Working with computers is typically considered a solitary activity, and many people consider tech-savvy people to be “geeky” and socially awkward. Is there something inherent in technology or in computer programming that rewards working alone? What are the advantages and disadvantages to working alone on a project?
2. A large-scale project like this differs quite a bit from the day-to-day exercises that are part of this class. Which aspects of a larger-scale project do you enjoy? Which aspects are more challenging?

SAMPLE SCREENS

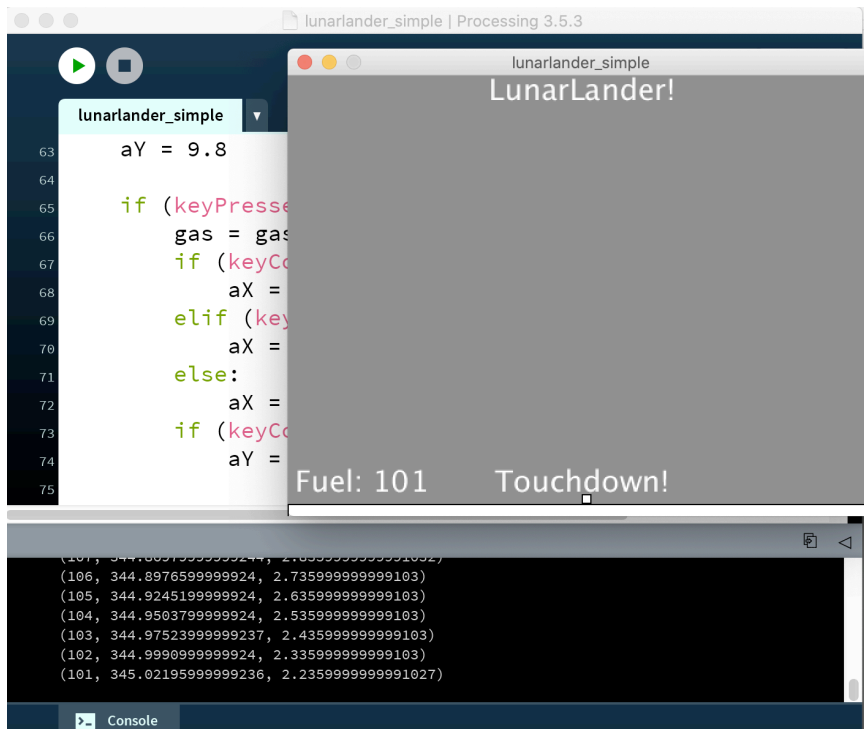
Snake

In the game of Snake, the user controls the direction of motion of a snake as it travels around the screen. The longer the snake travels, the more points you get, but be careful! The snake gets longer as it travels around the screen, and its speed increases as well. The game is over when the snake runs into a wall or runs into its own body.



Lunar Lander

In the Lunar Lander game, your Lunar Lander is falling toward the surface of the moon, and accelerating as it falls. You can use fuel in the lander to fire its thrusters to slow the descent, but you'll need to be careful: use too much fuel and you'll run out before the lander touches down, and if you use too little, the lander won't slow down in time for the landing.



Mugwumps

On a 10 x 10 grid, four mugwumps are hiding. Enter a coordinate where you think one might be—0,3 for example—and you'll get hints as to how far away the four mugwumps are located. Pay attention to these clues—you only have ten guesses to find all four mugwumps!

```

Guess # 5 Where do you think a mugwump might be (x,y)? 5,9
The distance to mugwump # 1 is 4.242640687119285
The distance to mugwump # 2 is 8.54400374531753
You've just found mugwump # 3
You've already found mugwump # 4
  * * * MUGWUMPS * * *
 9 | . . . . M . . . .
 8 | . . . . . . . . .
 7 | . . . . . . . . .
 6 | . . . . . . . . .
 5 | M . . . . * . . . .
 4 | . . . . . . . . .
 3 | . . . . . . . . .
 2 | . . * . . . . . .
 1 | . * . . . . . . .
 0 | . . . . . . . . .
+-----+
 0 1 2 3 4 5 6 7 8 9
Guess # 6 Where do you think a mugwump might be (x,y)?

```

Hangman

The classic children's game "Hangman" involves someone guessing letters in a secret word, and trying to

complete the word before a hangman is drawn, with an additional segment of the hangman being drawn each time a letter guess is incorrect.

This version reads in words from a large wordlist and selects one at random, so the programmer can play the game as well (once the program is written).

Example:

```
Enter a letter guess: o
Sorry, there is no o in the word.
m a _ i _ a _ _ _
```

```
Missed: ['s', 't', 'n', 'j', 'e', 'o']
```

```
----+
    |
    0
  +-+
  |
  /
```

```
Enter a letter guess: y
Yes, there is a y in the word!
m a _ i _ a _ _ y
```

```
Missed: ['s', 't', 'n', 'j', 'e', 'o']
```

```
----+
    |
    0
  +-+
  |
  /
```

```
Enter a letter guess: r
Sorry, there is no r in the word.
m a _ i _ a _ _ y
```

```
Missed: ['s', 't', 'n', 'j', 'e', 'o', 'r']
```

```
----+
    |
    0
  +-+
  | \
  / \
```

```
Game Over!
The word was magically!
```

```
Good game!
```