Assignment 1 - UFO Tracker

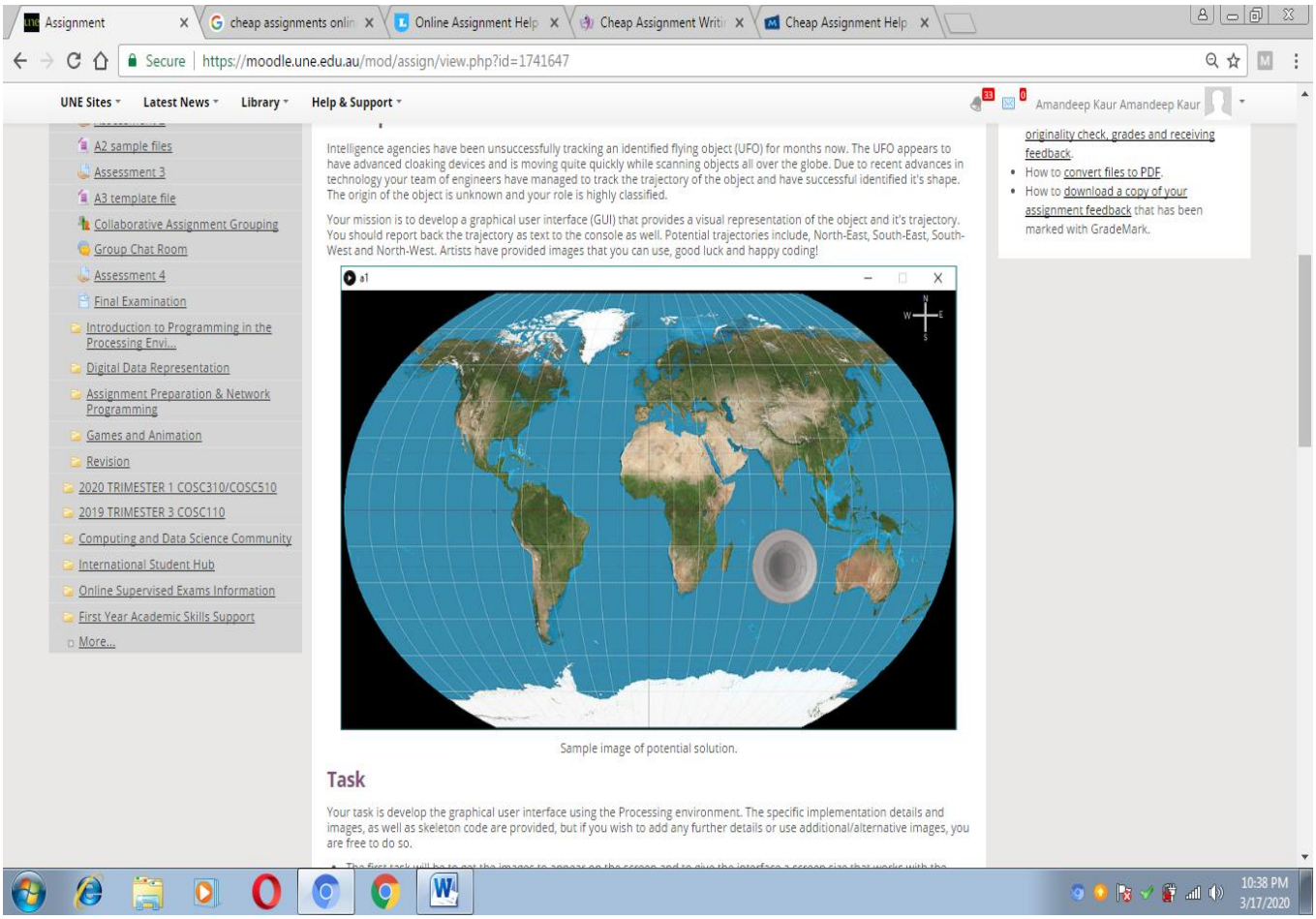
Aims

- Develop a piece of software that matches the task description.
- Solve problems and design software solutions using a high-level programming language and a range of technologies, protocols and algorithms.
- Build a program with visual elements.
- Move an object around the screen.

Description

Intelligence agencies have been unsuccessfully tracking an identified flying object (UFO) for months now. The UFO appears to have advanced cloaking devices and is moving quite quickly while scanning objects all over the globe. Due to recent advances in technology your team of engineers have managed to track the trajectory of the object and have successful identified it's shape. The origin of the object is unknown and your role is highly classified.

Your mission is to develop a graphical user interface (GUI) that provides a visual representation of the object and it's trajectory. You should report back the trajectory as text to the console as well. Potential trajectories include, North-East, South-East, South-West and North-West. Artists have provided images that you can use, good luck and happy coding!



sample image of potential solution.

Task

Your task is develop the graphical user interface using the Processing environment. The specific implementation details and images, as well as skeleton code are provided, but if you wish to add any further details or use additional/alternative images, you are free to do so.

- The first task will be to get the images to appear on the screen and to give the interface a screen size that works with the background image.
- Once this is complete, try to get the image object to move around the screen.
- As this is a fictional exercise, the starting position can be random, while the starting trajectory can be fixed.
- The interface will be restriction to a bounded 2D space, for this reason the object should bounce off the sides of the containing window and return in the opposite direction.
- The object only needs to move in 4 directions, North-East, South-East, South-West and North-West.
- Five marks have been allocated to any further effects you might want to add, this part is entirely up to you.
- Lecture examples and tutorials for weeks 1 to 4 should assist you.

Submission

Your submission will consist of three or more items and all should be contained in one zip file.

- The first item will be your processing (.pde) implementation file.
- The second and third item will be the background and UFO images.
- Lastly, any other files required by your program need to be included in the zip.

Marking

Implementation - 95%

Components in this section include:

- Solution Correctness:
    - Are all of the basic components implemented?
    - Is there any "buggy" behaviour?
- Quality of Solution:
    - Is your code broken down into functions?
    - Have you generated general-purpose / reusable functions?
    - Does the code run without errors / warnings?
    - Have you avoided the use of hard-coded values?
    - Is your solution efficient? (e.g. functions are not longer than 60 lines each, avoid repetition, etc.)
- Coding style:
    - Is your indentation consistent?
    - Have blank lines been used so that the code is easy to read?
    - Are any lines longer than 80 characters (i.e. off the end of the screen)?
    - Are capitalisation and naming conventions consistent?

- Internal Documentation:
    - Does your header block contain the author's name, the purpose of the program and a description of how to compile and run the solution.
    - Are variables and functions named in a meaningful way?
    - Are any obscure sections fully explained?
    - Do your functions have a header block that explains the purpose, its arguments and return value?

Additional effects – 5%

This part is only worth 5%, so be creative, but you do not need to go overboard. Components in this section could include:

- Sound effects.
- Additional motion or graphical effects.
- Additional functionality.