G(A)SP Balloon Simulation

*Prototype*

Designers: Maddie Mackey, Andrew Wilkie

Language: Python

Category: Automation

Requires: Python 2.7.1; PyGame 1.9.1

The G(A)SP Balloon simulation models the conditions and descent of a steerable stratospheric weather balloon over time, based off the balloon planned to be launched by the mechatronics class in September 2016. The model is written in Python 2.7.1, enabling PyGame, and self-constructed physics, GUI and simulation files.

# Understanding the Graphics

The simulation is from a bird’s eye view as from the moment the weather balloon begins its steerable descent.

The weather balloon and its sprite’s yaw are represented by the image ‘arrow.png’.   
The destination or “home” is represented by the image ‘home.png’.



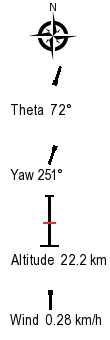
Graphics on the simulation window assist in visualising and representing data.

 Pause: click to pause the simulation.

PAUSE SCREEN (grey):  
 Un-pause: click to un-pause the  
 simulation.

Restart: click to restart the simulation.



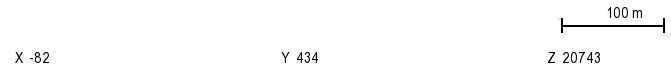


Compass.

Theta: angle directly from home to the weather balloon.

Yaw: actual angle the weather balloon is facing.  
(Differs from theta to correct for wind)  
  
Altitude: height of weather balloon.  
  
  
Wind: angle and speed of wind.

X: distance of weather balloon east/west from home (latitude)   
Y: distance of weather balloon north/south from home (longitude)   
Z: altitude in metres

Scale  


# Simulation

The simulation.py file contains the core of the simulation itself: the data for the balloon and calculations that determine how the parachute must steer against the imported physics values (and values from calculations done in this file) in order to get to its destination. This is done through the PID algorithm in the function calculatePID(). It then updates the graphics to the GUI.

# Physics

Physics is applied into python code from physics.py, containing the equations, as well functions to determine the values used in them – such as ones that are dependent on altitude. It changes these values into units and forms that directly affect the simulation.

# Libraries

All PyGame libraries are included when you download PyGame.

The built-in Python libraries used are:   
- Math – to calculate angles and to enable the use of all mathematical functions.   
- Random – for testing purposes.   
- Time – will allow the simulation to run in real time (not yet implemented).

# **Instructions**

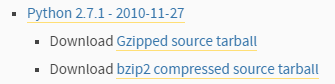
## Save Simulation File

Save “GASP\_Balloon\_Simulation.zip” to the hard-drive that you will be running your simulation on.

## Download and Install Python 2.7.1

Download Python 2.7.1 for your OS as per instructions on the website and save to your hard-drive:

Windows and Mac Python 2.7.1 download: <https://www.python.org/download/releases/2.7.1/>

UNIX Python 2.7.1 download: <https://www.python.org/downloads/source/> 

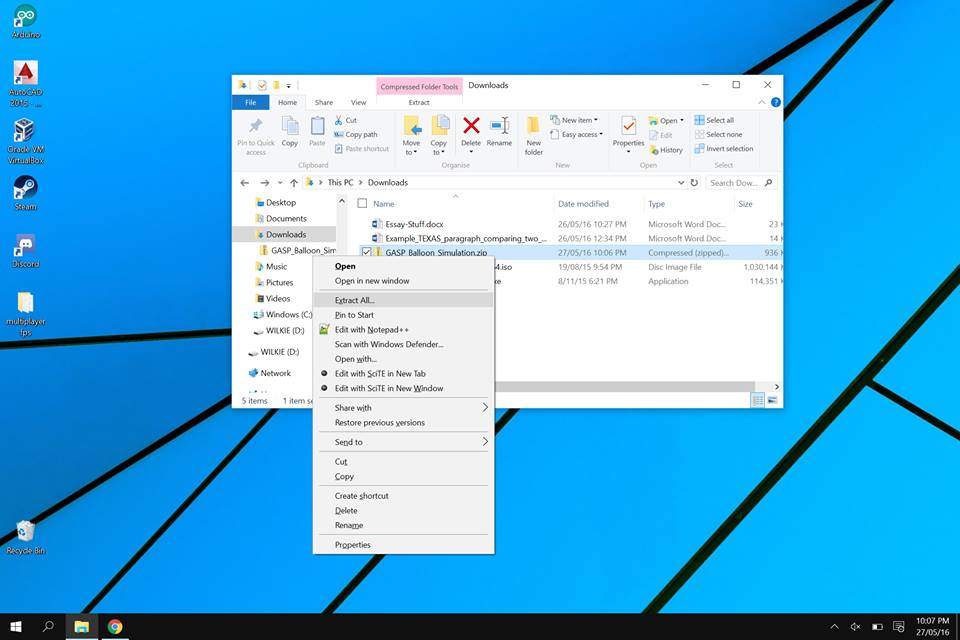
Follow the install instructions and prompts to suit your preferences.

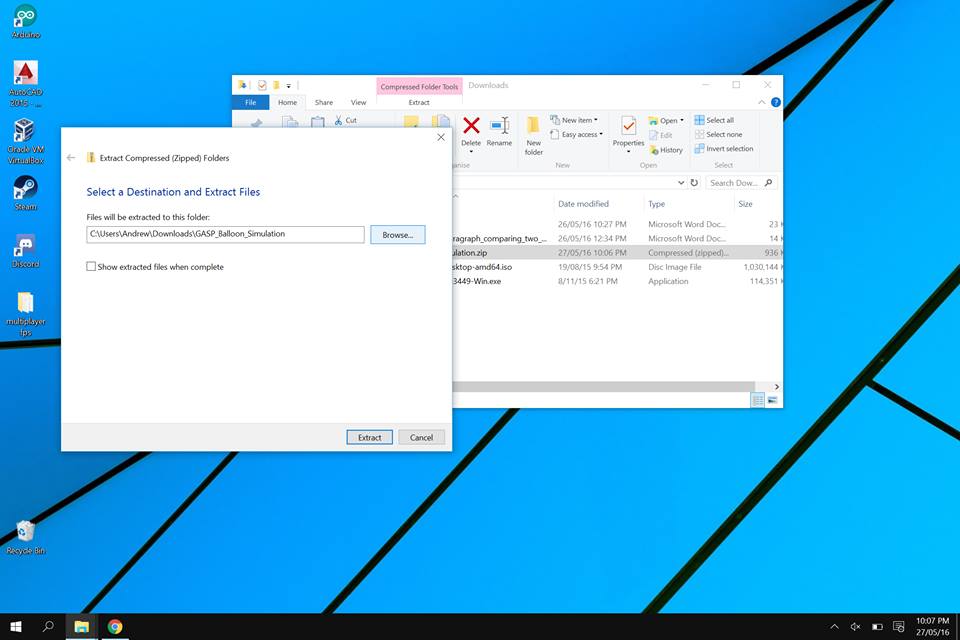
## Download and Install PyGame 1.9.1

Download PyGame 1.9.1 for your OS as per instructions on the website and save to your hard-drive (available at: [http://PyGame.org/download.shtml](http://pygame.org/download.shtml))

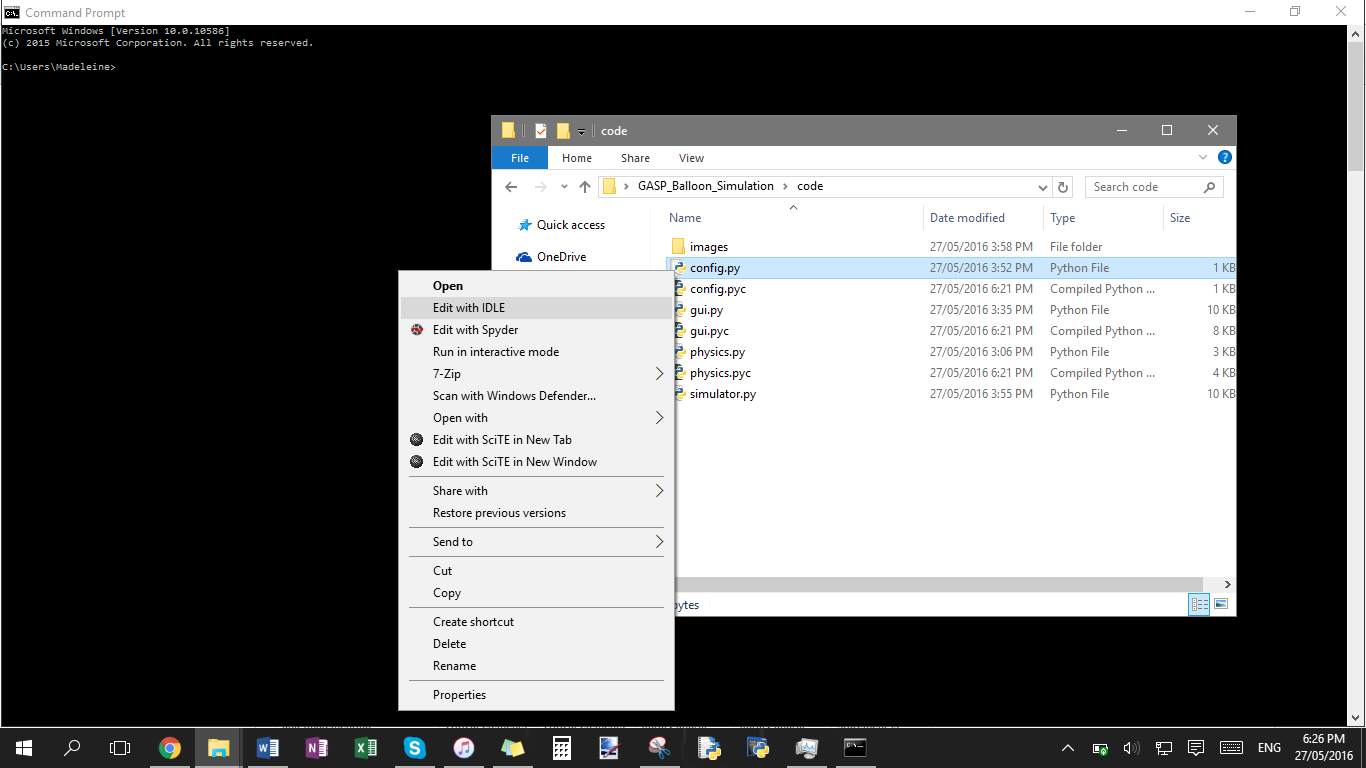
Follow the install instructions and prompts to suit your preferences.

## Unzip the Simulation File

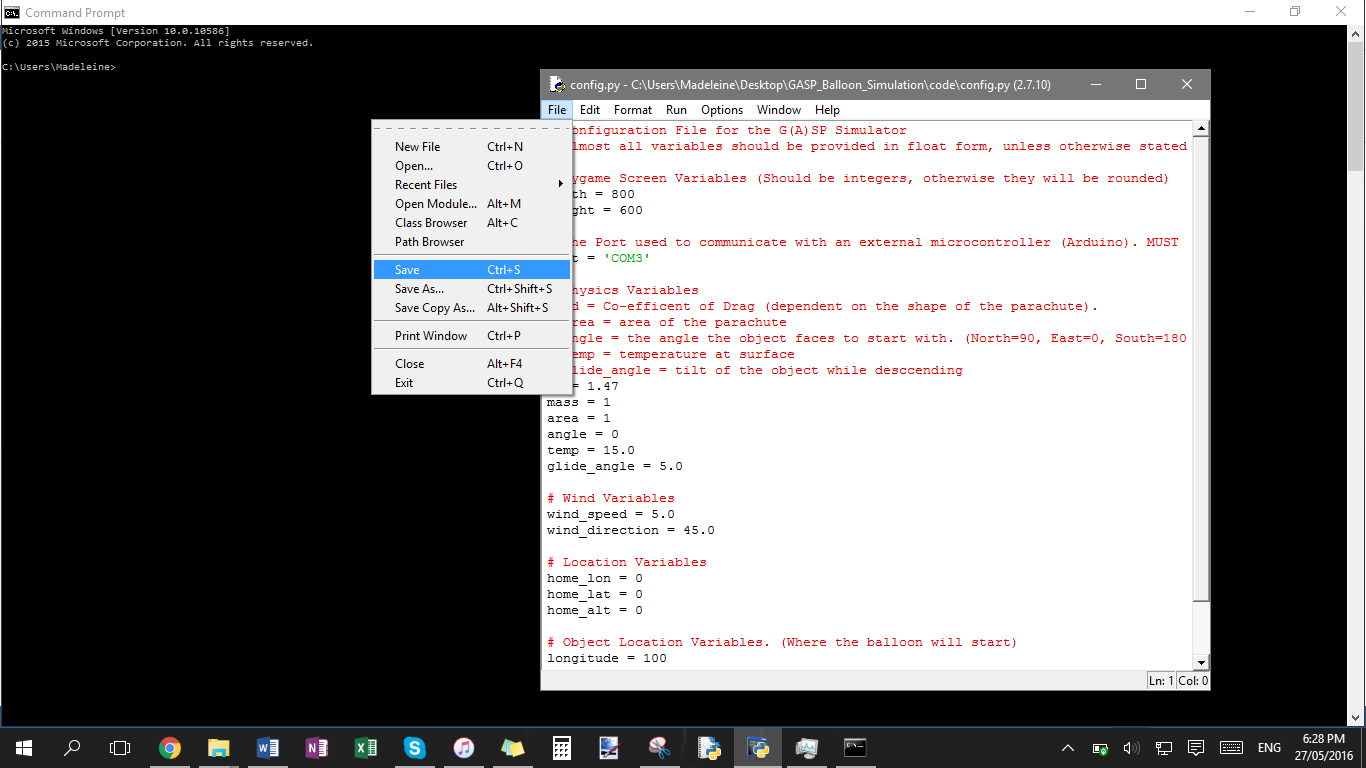
Go to the location on your hard-drive where you saved “GASP\_Balloon\_Simulation.zip” and unzip the folder by right clicking, selecting “Extract All…”.

Then selecting the location you wish to save the simulation file and click “Extract”. 

## Open the Configuration File

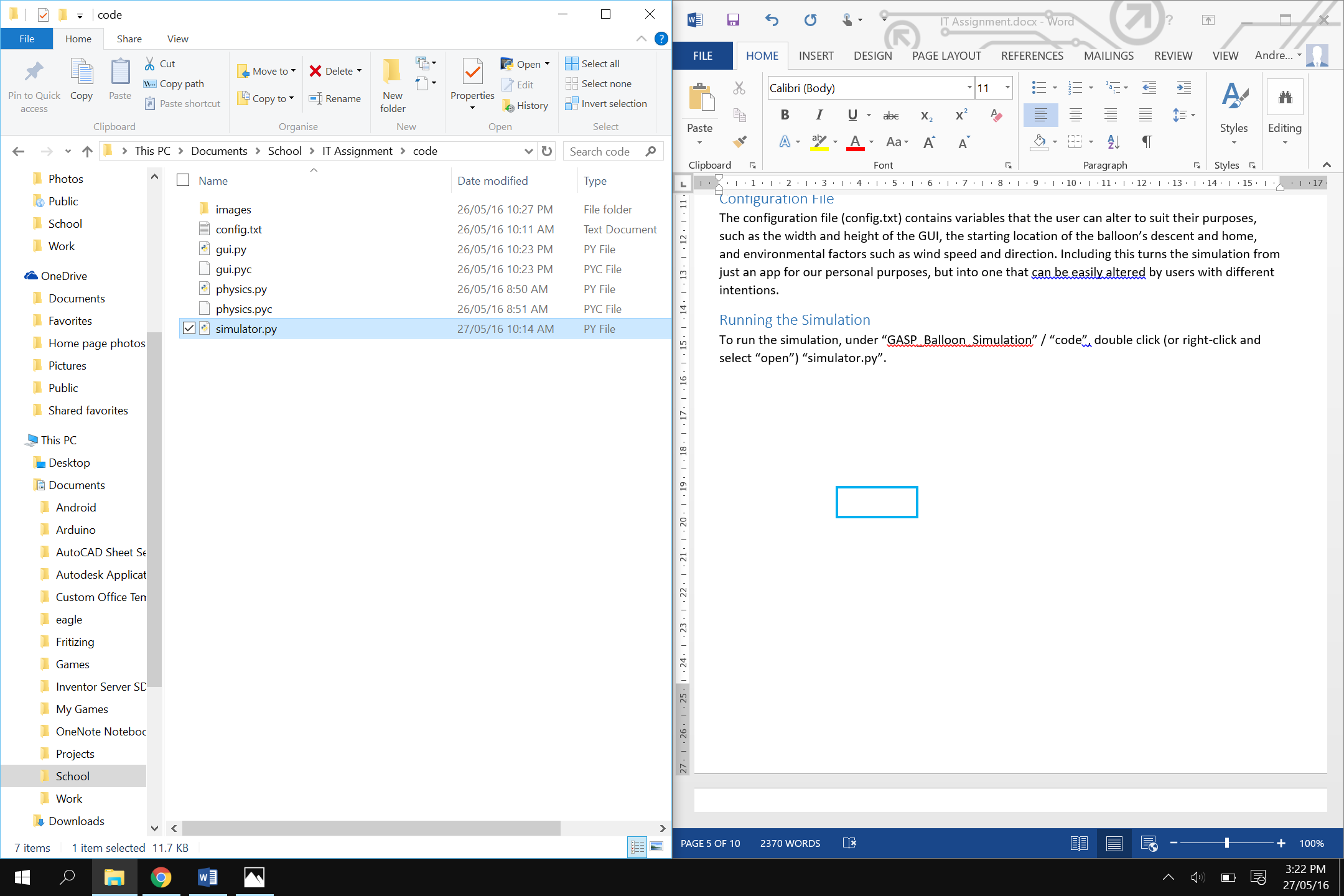
Open the folder “GASP\_Balloon\_Simulation”, “code”, and open “config.py” in python by right-clicking and selecting “Edit with IDLE”. 

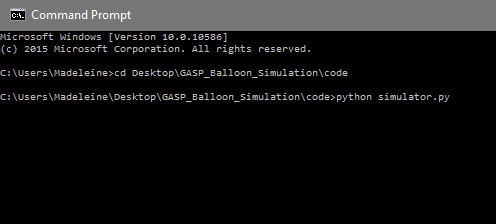
## Edit the Configuration File

The configuration file contains variables that effect the simulation, such as GUI preferences and environmental factors. This enables users to easily interact with and alter the simulation to suit their purposes.   
Save any changes.  


## Running the Simulation

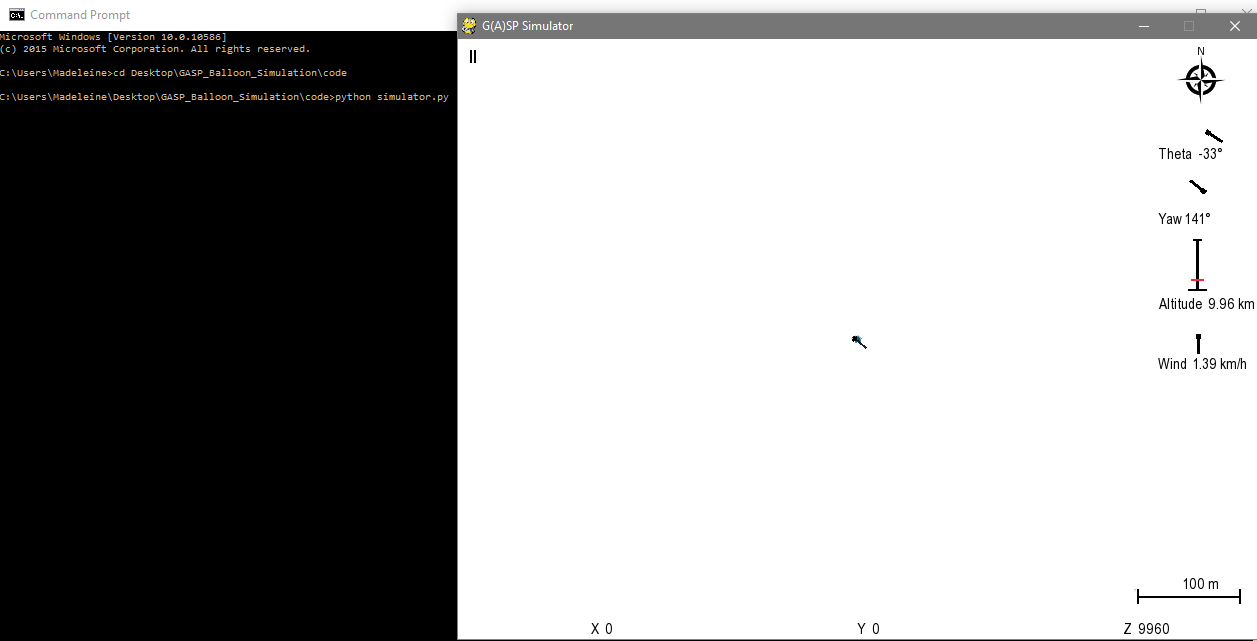
### Windows

To run the simulation, open the folder “GASP\_Balloon\_Simulation”, then open “code”, double click on “simulator.py” (or right-click and select “open”).



Alternatively, through command prompt,   
go to the simulation file location (GASP\_Balloon\_Simulation\code).

Type “python simulator.py” and press enter.



The simulation will run as above and according to the variables in “config.py”.

### Mac OSX and UNIX

Open “Terminal” (or “Shell” for UNIX users).

Navigate to the folder “GASP\_Balloon\_Simulation”, then into “code”.

Run the script using python “python simulator.py”

## Simulation Shortcuts

These shortcuts make running the simulation a little quicker, allowing users to pause to note particular aspects or restart using different variables, or the same.

esc: quit  
p: pause  
r: restart

# Automation

The G(A)SP Balloon Simulation is a model of a feedback system in the physical world. In the real world, the simulation would generate input from sensors, receive it through serial and send into the user interface, to then calculate the output and send it back to actuators (controlling the steerable parachute) altering the next input. This is the remote collection of data, which is processed and analysed in order to physically alter the next iteration – an automated system.

# Equations

## Newton’s Second Law

Where F=force, m=mass, a=acceleration  
This equation was used in the calculation of weight in the “gravity()” function.

Drag

Where Fdrag=force of drag, Cd=drag coefficient (determined by shape of parachute), ρ=air density, v=velocity, A=area  
<https://www.grc.nasa.gov/www/k-12/airplane/drageq.html>

Terminal velocity

(used to find velocity)  
Where Fnet=force of terminal velocity   
<http://hyperphysics.phy-astr.gsu.edu/hbase/airfri2.html>

# Limitations

### Altitude

The simulator can only simulate properly up to 43,000m altitude due to the equations used in the physics file.

### Boundaries

In order to ensure that the simulation is visual on the screen, the weather balloon must remain within the range of negative half of the width of the GUI to positive half of the width on the x-axis, and the same but for height on the y-axis.

### Time

The simulation does not run in real time, but is determined by the speed of the machine it is being run on. Each iteration of the simulation represents 1 second.

# Purpose

The simulation is designed to imitate everything a stratospheric balloon would experience during its descent with a steerable parachute. The simulator allows us to test and develop control algorithms and gain information about what the balloon and payload might go through during the journey, helping us to understand how we can alter the real thing to optimise our launch in September. This makes this simulation a useful application and relevant to the real world.

Due to the nature of the simulation, users can not only simulate weather balloon drops, but also parachuting and other steerable freefalling objects through a dynamic medium – such as air or water. In which wind would be translated to currents and air density would become water density.

# Attribution of Work

Simulation code; configuration incorporation; PID algorithm and tuning; physics file; GUI file: Andrew Wilkie  
Research; theory; physics; equations; PID algorithm and tuning; debugging; simulation code; documentation: Maddie Mackey

# References

Allen, Blake. "Install Python 3 & Python 2.7 for Mac OS X the Right Way." *YouTube*. How To Bioinformatics, 04 Sept. 2013. Web. 25 May 2016. <https://www.youtube.com/watch?v=QhutbLBKBok>.

Hall, Nancy. "The Drag Equation." *NASA Physics*. NASA, 5 May 2015. Web. 27 May 2016. <https://www.grc.nasa.gov/www/k-12/airplane/drageq.html>.

Nave, R. "Fluid Friction." *Hyperphysics*. N.p., 20 May 2016. Web. 2009. <http://hyperphysics.phy-astr.gsu.edu/hbase/airfri2.html>.

"PyGame." *PyGame Downloads*. PyGame 1.9.1, 6 Aug. 2009. Web. 10 May 2016. <http://PyGame.org/download.shtml>.

Rossum, G. Van. "Python Language Reference." *Python.org*. Python Software Foundation, n.d. Web. 10 May 2016. <https://www.python.org/download/releases/2.7.1/>.

# Rubric

Andrew  
Madeleine  
Both

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criteria | Grading Scale | | | | |
| **Explanation of Libraries/Code**  Planning, Designing, Creating and Implementing | **10**  Explains the role the libraries/code plays in making the computer science behind the application usable and accessible to developers, in language that is both clear and succinct (abstraction). | **8.5**  Explains the role the libraries/code plays in making the computer science behind the application usable and accessible to developers, but may use language that is either overly technical, verbose or unclear. | **7**  States the key elements of the libraries/code that makes the computer science usable and accessible to developers, but no explanation of how it works is given. There may be some inaccuracies or errors in what is presented. | **5.5**  Uses general terms to explain how the library/code works for developers, but lacks specific details about the hooks used. Errors have been made in parts of the explanation. | **4**  There is no statement about the role of the libraries/code used to implement the computer science, or the statements made are incorrect. |
| **Use of Libraries**  Planning, Designing, Creating and Implementing | **10**  Identifies the critical elements of the library and/or API used, and explains the importance of the structures of both inputs and/or outputs of the library/API, with clear examples given. | **8.5**  Identifies the critical elements of the library and/or API used. States the inputs and/or outputs used and the reason for them, and demonstrates examples. | **7**  Refers to the library and/or API in general terms and makes mention of the elements accessed. States the inputs and/or outputs used but does not provide a detailed explanation of why they are important. Examples are given. | **5.5**  Refers to the library and/or API but does so only in general terms and without specific reference to the elements accessed. The inputs/outputs used are referred to in general terms without examples. | **4**  Either no statement is made about the library/code elements used, or there is no mention of the inputs/output used. |
| **Group Contribution**  Communication and Interpersonal Skills | **10**  The student contributes to all aspects of the project in a meaningful way, and works well with others in the group. | **8.5**  The student contributes to the project in a meaningful way, but may not involve themselves in all aspects of the task. They generally work well with others in the group. | **7**  The student contributes to some aspects of the project but relies heavily on the work of others. They do not cause conflict within the group but may not work well with others. | **5.5**  The student contributes to very specific aspects of the project with little involvement in other parts. They may create conflict in the group or negatively impact the work of others. | **4**  The student does not contribute to the project, or has a primarily disruptive influence on the group. |
| **Internal Documentation**  Communication and Interpersonal Skills | **10**  Code is documented extensively, with clear use of explanatory comments in appropriate parts of the application source | **8.5**  Code is documented with clear use of comments in appropriate parts of the application source. Comments may be too verbose or lack clarity in some parts. | **7**  Code is documented throughout the application source. Comments may lack clarity or be missing in necessary places. | **5.5**  Code has been commented but the comments are very sparse and add little to the reader's understandability of the source code. | **4**  Either no comments are presented in the code, or the comments included are trivial in nature and do not assist with readability of the source. |
| **User Documentation**  Communication and Interpersonal Skills | **10**  The user documentation for the application is clear and well presented, and includes screenshots and explanations of each step as appropriate. There are clear statements about application installation and requirements. | **8.5**  The user documentation for the application is well presented. Steps are supported with screenshots and explanations, but some may be ambiguous or do little to add to the clarity of the document. There are clear statements about application installation and requirements. | **7**  The user documentation for the application is well presented, but contains spelling and/or noticeable grammatical errors. There may be insufficient screenshots and/or explanations to assist the reader. Statements about application installation and requirements may be lacking in detail. | **5.5**  The user documentation for the application is very basic and contains many spelling and/or grammatical errors. There are very few screenshots and/or explanations to assist the reader, and statements about application installation and requirements may not be present. | **4**  The user documentation is either not presented, or is poorly structured and difficult to follow. Statements about application installation and requirements are missing. |
| **Scalability of Application**  Flexible, Adaptive and Creative Thinking | **10**  The scalability of the application’s potential is described with clear references to the design used, and more complex scenarios using similar concepts are explained with clear examples. | **8.5**  The scalability of the application’s potential is described, and more complex scenarios using similar concepts explained. | **7**  The scalability of the application’s potential is briefly described, with complex scenarios using similar concepts mentioned but not explained in detail. | **5.5**  The scalability of the application’s potential is stated and examples of how it has been applied given, but without the link between the examples presented and the application articulated. | **4**  There is no mention of the scalability of the application, nor its potential. Examples may be given of the application's use in the real world, but these are limited and do not demonstrate understanding of the breadth of the topic. |