

Project requirements

In this project you are required to:

- *Implement a GUI application in Python 3*
 - *The GUI app interfaces the user to an underlying data store.*
- *The project should be completed using the agile principles articulated in Pilone & Miles.*

Make sure to meet with your instructor as regularly as you require to ensure that the project is scoped appropriately – neither too narrowly nor too widely.

Beyond those requirements, you and your team are free to choose your own domains to create a solution project for.

Instructions - Run Program

Prerequisites:

- Python3
- Connection with Tello drone
- Packages (if Anaconda is not installed):
 - threading
 - socket
 - sys
 - time
 - pytest
 - datetime import date
 - sys
 - sqlite3
 - Os
- DB Browser for SQLite

Instructions:

1. Download "Drone_Control.py"
2. Open command line (e.g. Windows, Windows key+R, type "cmd", enter)
3. Change current directory to the location of the "Drone_Control.py" file
 - a. E.g. cd Desktop (if the file is in your Desktop folder)
4. If running Python, please enter "python Drone_Control.py" to run the program
 - a. If running ipython, please enter "ipython" to begin using ipython and enter "run Drone_Control.py" to run the program
5. Follow instructions from the command line:

- a. The below will not be available if you are not connected to the Tello drone:
 - i. Enter your name
 - ii. Select the mode that you want to use
 1. a for automatic mode
 2. m for manual mode
 - iii. Enter command mode by typing "command"
 - iv. Enter the Help command by typing "help command"
 - v. Enter instructions (e.g. takeoff)
 - vi. Once complete with flight, enter "end" to exit the program
6. After entering "end" and inputting flight notes, the flight record will be saved in the SQLite3 database.
7. You can use a GUI program like "DB Browser for SQLite" to access the flight records.

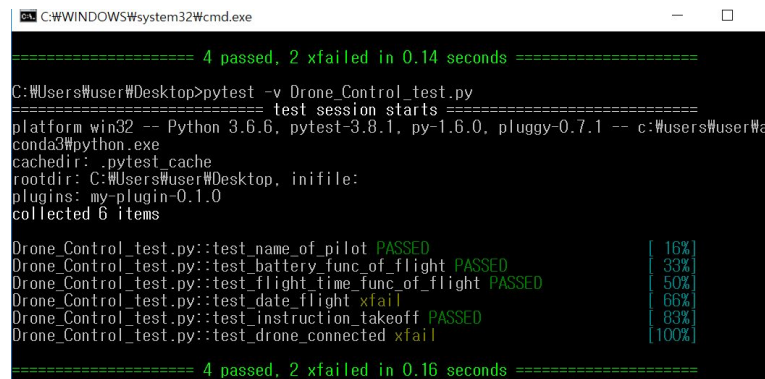
Instructions - Run Tests

Prerequisites:

- Python3
- Packages:
 - Install Pytest

Instructions:

1. Open command line (e.g. Windows, Windows key+R, type "cmd", enter)
2. Change current directory to the location of the "Drone_Control_test.py" file
 - a. Note: Please ensure that Drone_Control.py and Drone_Control_test.py are in the same directory
3. Type "pytest -v Drone_Control_test.py"
4. View the results of the five tests
 - a. You will see four "PASSED" and two "XFAILED" results



```

C:\WINDOWS\system32\cmd.exe

===== 4 passed, 2 xfailed in 0.14 seconds =====

C:\Users\User\Desktop>pytest -v Drone_Control_test.py
test session starts
platform win32 -- Python 3.6.6, pytest-3.8.1, py-1.6.0, pluggy-0.7.1 -- c:\Users\User\AppData\Local\Continuum\anaconda3\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\User\Desktop, inifile:
plugins: my-plugin-0.1.0
collected 6 items

Drone_Control_test.py::test_name_of_pilot PASSED [ 16%]
Drone_Control_test.py::test_battery_func_of_flight PASSED [ 33%]
Drone_Control_test.py::test_flight_time_func_of_flight PASSED [ 50%]
Drone_Control_test.py::test_date_flight xfail [ 66%]
Drone_Control_test.py::test_instruction_takeoff PASSED [ 83%]
Drone_Control_test.py::test_drone_connected xfail [100%]

===== 4 passed, 2 xfailed in 0.16 seconds =====
  
```

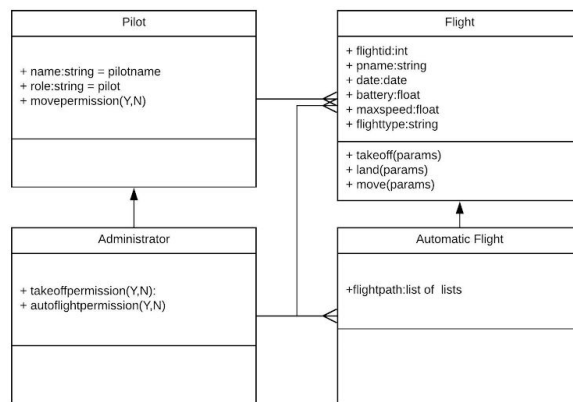
Concept

Control the Tello drone with a GUI to take off, fly, and land.

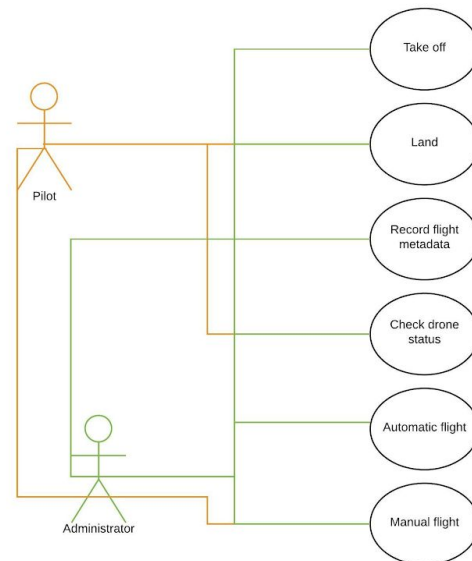
Use Cases and Class Relationships

Python Pilots
IST 303
20181011

Class diagrams



Use cases



Requirements - User stories and Tasks

1. Title: Take off
 - a. Description: As a user, I should be able to launch the drone from the ground into the air, so that I can begin a flight route.
 - b. Completion time: One week
 - c. Priority: 10
 - d. Tasks:
 - i. Create a command to connect with the drone (7 hours)
 - ii. Create a Flight class (12 hours)
 - iii. Create a Flight object in the datastore (12 hours)
2. Title: Land drone

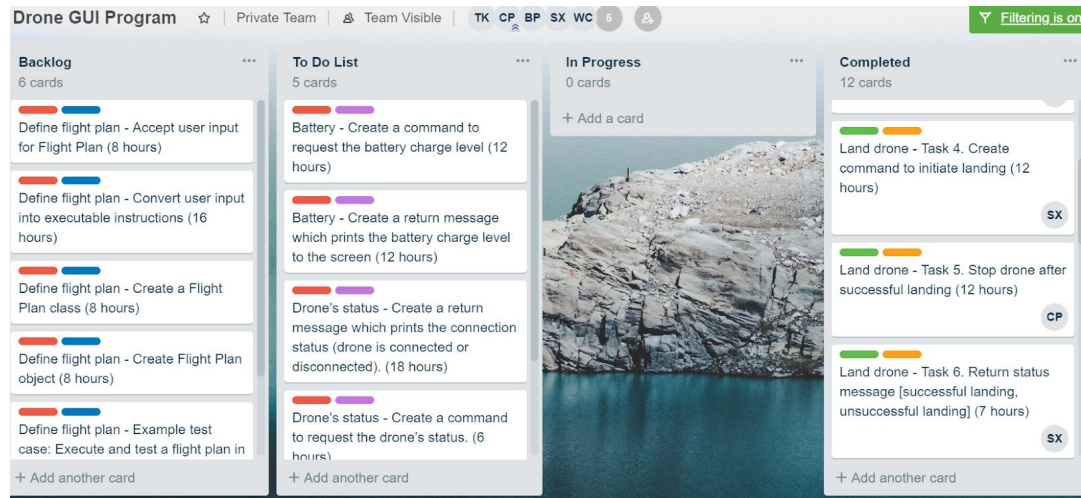
- a. Description: As a user, I should be able to land the drone safely on the ground so that I can complete the flight.
 - b. Completion time: One week
 - c. Priority: 10
 - d. Tasks:
 - i. Program the code for landing (31 hours)
 - 1. Create a command to initiate landing (12 hours)
 - 2. Stop drone after successful landing (12 hours)
 - ii. Return status message [successful landing, unsuccessful landing] (7 hours)
3. Title: Choose flight mode
- a. Description: As a user, I should be able to choose the flight mode (manual control or flight plan) so that I can fly the drone in two different ways.
 - b. Completion time: One week
 - c. Priority: 20
 - d. Tasks:
 - i. Create a command to receive user input for flight mode (manual or automatic) (12 hours)
4. Title: Fly drone in manual mode
- a. Description: As a user, I should be able to control the drone's movement in real time so that I can improvise the flight path.
 - b. Completion time: Two weeks
 - c. Priority: 10
 - d. Tasks:
 - i. Create commands to receive user input for flight direction, distance, and speed. (20 hours)
 - ii. Create commands to send user input to drone for execution (20 hours)
5. Title: Display the remaining charge of the battery.
- a. Description: As a user, I should be able to see the remaining battery charge in the GUI.
 - b. Completion time: One week
 - c. Priority: 30
 - d. Tasks:
 - i. Create a command to request the battery charge level (12 hours)
 - ii. Create a return message which prints the battery charge level to the screen (12 hours)
6. Title: Display the drone's status
- a. Description: As a user, I should be able to see the connection status of the drone. For example, if the drone is either connected or disconnected from the computer.
 - b. Completion time: One week
 - c. Priority: 30
 - d. Tasks:
 - i. Create a command to request the drone's status. (6 hours)

- ii. Create a return message which prints the connection status (drone is connected or disconnected). (18 hours)
- 7. Title: Define flight plan
 - a. Description: As a user, I should be able to design a flight plan in the GUI and have the drone execute the flight plan so that I can fly the drone automatically.
 - b. Completion time: 2 weeks
 - c. Priority: 20
 - d. Tasks:
 - i. Create a Flight Plan class (8 hours)
 - ii. Accept user input for Flight Plan (8 hours)
 - iii. Create Flight Plan object (8 hours)
 - iv. Execute Flight Plan object (16 hours)
 - v. Convert user input into executable instructions (16 hours)
 - vi. Example test case: Execute and test a flight plan in a square (8 hours)
- 8. Title: Record flight metadata
 - a. Description: As a user, I should be able to write flight notes after a flight so that I can have a record for flights for reference (e.g. debugging)
 - b. Completion time: 1 week
 - c. Priority: 30
 - d. Tasks:
 - i. Create a command to accept user input for flight notes (11 hours)
 - ii. Create a command to accept user input for temperature during flight (11 hours)
 - iii. Create a command to accept user input for pilot name after flight (11 hours)
 - iv. Save flight notes, temperature, and pilot name as a part of the Flight object (17 hours)
- 9. Archives (User stories on hold or eliminated)
 - a. Title: Fly in a circle
 - i. Description: As a user, I should be able to fly the drone in a circle.
 - ii. Completion time: Two weeks
 - iii. Priority: 30
 - iv. Notes: This user story is accounted for in the "Define flight plan" user story.
 - b. Title: Take Photo
 - i. Description: As a user, I should be able to take a photo, in order to see what the drone saw during flight.
 - ii. Completion time: 3 weeks
 - iii. Priority: 30
 - iv. Notes: The drone manufacturer does not support taking photos through the interface and the customer agreed to remove this from scope.
 - c. Title: Register user

- i. Description: As a user, I should be able to create an account to be recognized in the system as a unique user.
 - ii. Completion time: 2 weeks
 - iii. Priority: 10
 - iv. Tasks:
 - 1. Create an Account class (3 days)
 - 2. Create user interface to enter data to create an account (5 days)
 - 3. Create an Account object with the user-entered data (3 days)
- d. Title: Login/out
 - i. Description: As a user, I should be able to log in to be able to access the system.
 - ii. Completion time: 2 weeks
 - iii. Priority: 10
 - iv. Tasks:
 - 1. Allow user to start a session by entering a valid Username and Password (1 week)
 - 2. Check user-entered data against the Account data store (3 days)
 - 3. Create a return message for valid and invalid user credentials (3 days)
 - 4. Allow user to end the session (3 days)
- e. Title: Forgot password
 - i. Description: As a user, I should be able to reset my password in case I forget it, in order to access the system.
 - ii. Completion time: 2 weeks
 - iii. Priority: 20
 - iv. Tasks:
 - 1. Create a command to send users message to confirm identity (2 days)
 - 2. Create a command for users to reset their password. (4 days)
 - 3. Test the code (2 days)

Allocation

Please check our Trello website (<https://trello.com/b/BYz9KJnR/drone-gui-program>)

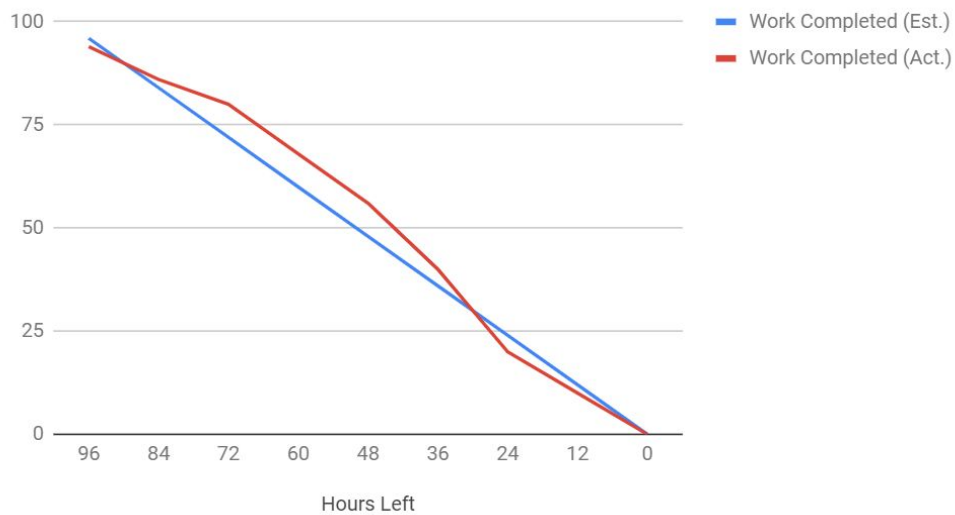


Burn-Down Chart

- Velocity:
 - Numerator: 8 hours/person/week (with 4 people, we estimate 32 hours of work per week)
 - Denominator: 3 days * 8 hours/day = 24 hours/person/week (Assumption: 3 possible working days per week, we estimate 3 days/week as the total number of days/week)
 - Estimated length of each iteration:
 - Iteration 1.1 (2 weeks) = 62 hours (Oct 7 - 21)
 - Iteration 1.2 (2.5 weeks) = 90 hours (Oct 22 - Nov 8)
 - Iteration 2.1 (2 weeks) = 60 hours (Nov 9 - Nov 22)
 - Iteration 2.2 (2 weeks) = 64 hours (Nov 23 - Dec 6)

- Velocity for the iteration 1.2 : (16 hours/person) / (48 hours/person) = $\frac{1}{3}$

Iteration 1.2 Burndown Chart



Stakeholders

- Farmers - check crops
 - As a farmer, I want to plan a flight route, to check my crops.
- Landscape architects - help design
 - As a landscape, I want to take photos of the site, to help design the landscape.
- Mining planner - check progress of mining
 - As a mining planner, what I want is the geographical characteristics of particular areas like longitude and latitude.
- People in the travel industry or real estate - plan aerial shots of a location
 - As a travel blogger, I want to post aerial photographs of my travel locations to my website and social media platforms
- Toy store - let users use a simple UI to control the drone
 - As a player, I want fly drone in a circle because I think it's cool.
- Sell software to people that don't have a suitable phone to control it

Data elements (flight log)

- Flight ID
- Flight time
- Pilot ID
- Date
- Location
- Flight time
- Temperature

- ~~Distance~~
- ~~Photo taken during flight~~
- ~~Complete flight? [True/False]~~
- ~~Reason for the failure, if applicable~~
- ~~# of charges~~
- ~~Location of charge~~

GitHub

GitHub repository: https://github.com/thkim91/IST_303-GroupProject--Group2.git

Team members

- Taehoon Kim
 - Phone #: 909 543 8810
 - Email address: th.kim9112@gmail.com
- Charidy Paige
 - Phone #: 205 332 4191
 - Email address: mecpaige@gmail.com
- Bill Pepper
 - Phone #: 818 793 3384
 - Email address: willamepepper@gmail.com
- Siyu Xiang
 - Phone #: 909 506 8559
 - Email address: siyu7866@gmail.com

Team meetings

1. 9/26/201
 - a. Attendees: Taehoon, Charidy, Bill, & Siyu
 - b. Notes:
 - i. Discussed project concept and user stories
 - ii. Collaboratively created new user stores (e.g. flight plan)
 - iii. Scheduling time to meet with Prof. Chipidza on project scope
2. 09/27/2018
 - a. Attendees: Taehoon, Charidy, Bill, Siyu, & Prof. Chipidza
 - b. Notes:
 - i. Identify stakeholders
 - ii. Identify different kinds of users
 - iii. New user stories:
 1. Change the password, should be able to access.
 - iv. Determine reports
 - v. Consider the feasibility of the project

- vi. You can change it to normal airline company if drone does not work
- 3. 10/07/2018
 - a. Attendees: Taehoon, Bill, & Siyu
 - b. Notes:
 - i. Added user stories
 - ii. Divide user stories into tasks
 - iii. Allocated the tasks
 - iv. Velocity Calculations & Burndown charts
 - v. Tried to send command to Drone
- 4. 10/11/2018
 - a. Attendees: Taehoon, Charidy (remote), Bill, Siyu, & Prof. Chipidza
 - b. Notes:
 - i. Discuss progress, velocity, datastore, grades for ReadMe
 - ii. Tools
 - 1. Database - SQLite
 - 2. GUI - Tkinter
- 5. 10/14/2018
 - a. Attendees: Taehoon, Charidy (remote), Bill, Siyu
 - b. Notes:
 - i. Code review and bug fixing
 - ii. Draft instructions for code and running tests (Pytest)
 - iii. Update burndown chart



c.

6. 10/30/2018

a. Attendees: Taehoon, Charidy, Bill

b. Notes:

- i. Git updates
- ii. Progress updates (database, program, GUI)
- iii. Presentation draft
- iv. Scheduling

V.



7. 11/3/2018

a. Attendees: Charidy, Taehoon, Bill, Siyu

b. Notes:

- i. Code modification
- ii. Powerpoint slide presentation
- iii. Demo(s)
- iv. Meet with Professor



v.

References

- <https://www.rzyzerobotics.com/tello>
- <https://dl-cdn.rzyzerobotics.com/downloads/tello/0228/Tello+SDK+Readme.pdf>
- https://www.tutorialspoint.com/python/python_gui_programming.htm

Group Meetings:

Thurs 11/8 Milestone 1 Presentation and Iteration 1.2 Release

Sat 11/10

Sun 11/11

Sat 11/17

Sun 11/18

Thurs 11/22 - Iteration 2.1 Release

Sat 11/24 Thanksgiving Break

Sun 11/25 Thanksgiving Break

Thurs 12/29 Charidy out of Town on business

Sat 12/1 Charidy out of town on business

Sun 12/2 Charidy out of town on business

Thurs 12/6 Iteration 2.2 Release

Sat 12/8

Sun 12/9

Thurs 12/13 Milestone 2 Presentation

*****Sundays will need to be after 2PM*****