



AAE2004 Introduction to Aviation Systems

AAE

Design of Path Planning Algorithm for Aircraft Operation

First Week

Dr Li-Ta Hsu and Dr Kam Hung NG
Assisted by
Miss Hiu Yi HO (Queenie), Miss Yan Tung LEUNG (Nikki)

Lecturer's Information

- Instructor: Dr Li-Ta HSU
- Office: QR828
- Phone: 3400-8061
- Email: lt.hsu@polyu.edu.hk
- Office Hour: by appointment
- Expertise: GPS navigation, Autonomous driving, Pedestrian localization using Smartphone, Sensor Integration

Li-Ta HSU

1985.08 – Born in a fish farmer family in Tainan, Taiwan

2003.06 – Graduated from Kang Ming Senior High School, Taiwan

2007.06 – Bachelor of NCKU Department of Aeronautics and
Astronautics (DAA), Taiwan

2010.09 – Ph.D. Candidate of NCKU DAA, Taiwan

2012.02 – Visiting Researcher
in University College London, UK

2012.06 – Part-time Consultant for Spirent, UK

2013.07 – Visiting Researcher
in Tokyo Marine University, Japan

2013.12 – Ph.D. of NCKU DAA, Taiwan

2014.04 – Postdoctoral Researcher in the
University of Tokyo , Japan

2017.05 – Assistant Professor
in AAE of PolyU, Hong Kong

2021.07 – Associate Professor
in AAE of PolyU, Hong Kong



Ground Rules

For students

- Try to speak as much English as possible.
- Participate the class activates assigned.

For teaching staffs

- Reply your email with 3 working day.
- Open to any question regards to the subject

For us!

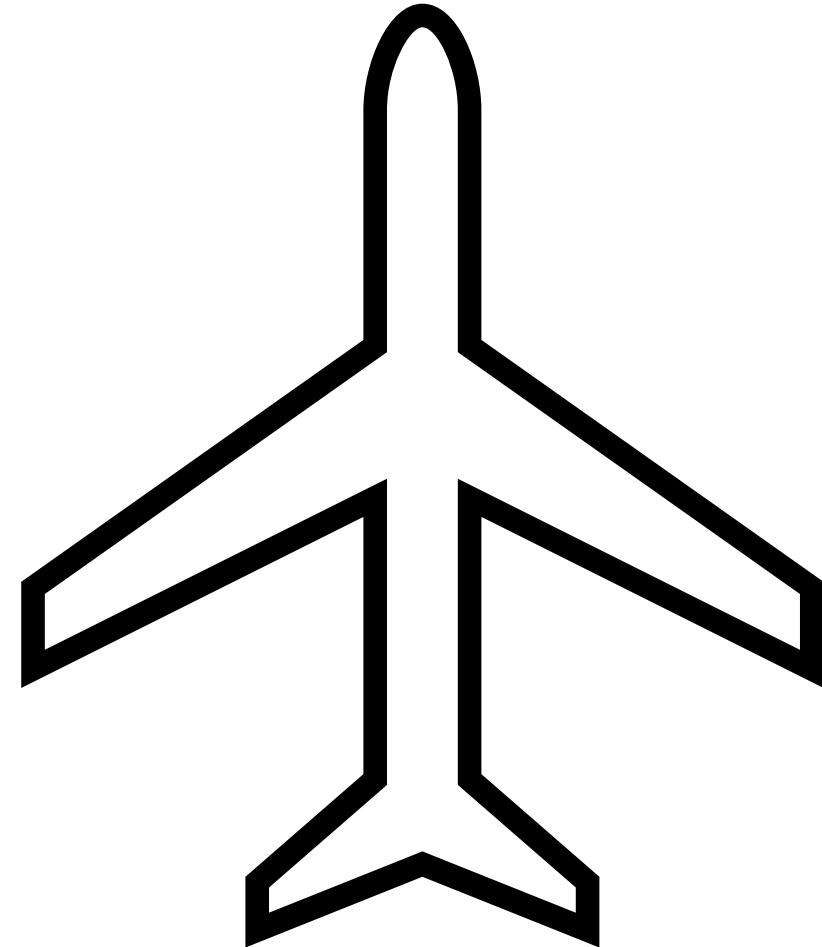
- Keep an open mind—enter the classroom dialogue with the expectation of learning something new. Look forward to learning about—and being challenged by—ideas, questions, and points of view that are different than your own.
- Arrive on time to the class and finish the class on time

Necessary Information

- Course Repository link: https://github.com/IPNL-POLYU/PolyU_AAE2004_Github_Project
- TA Information & Contact:
 - Group 1-5: Queenie Ho (hiu-yi.ho@connect.polyu.hk)
 - Group 6-10: Nikkie Leung (yan-tung.leung@connect.polyu.hk)

Week 1 Content

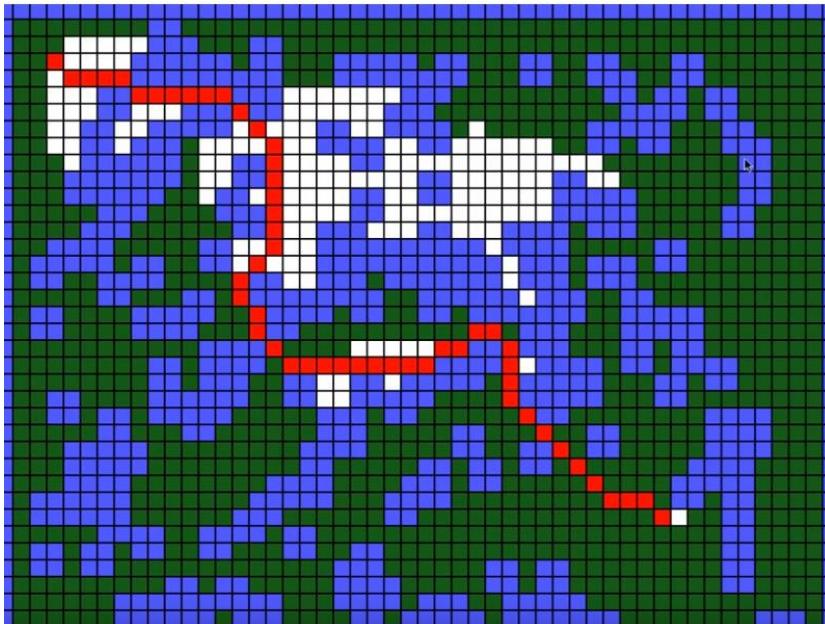
1. Introduction to Path Planning
2. Introduction to GitHub
(Background)
3. Introduction to GitHub Operations
4. Software Installation and setup
Guide
5. Path Planning Code Guide



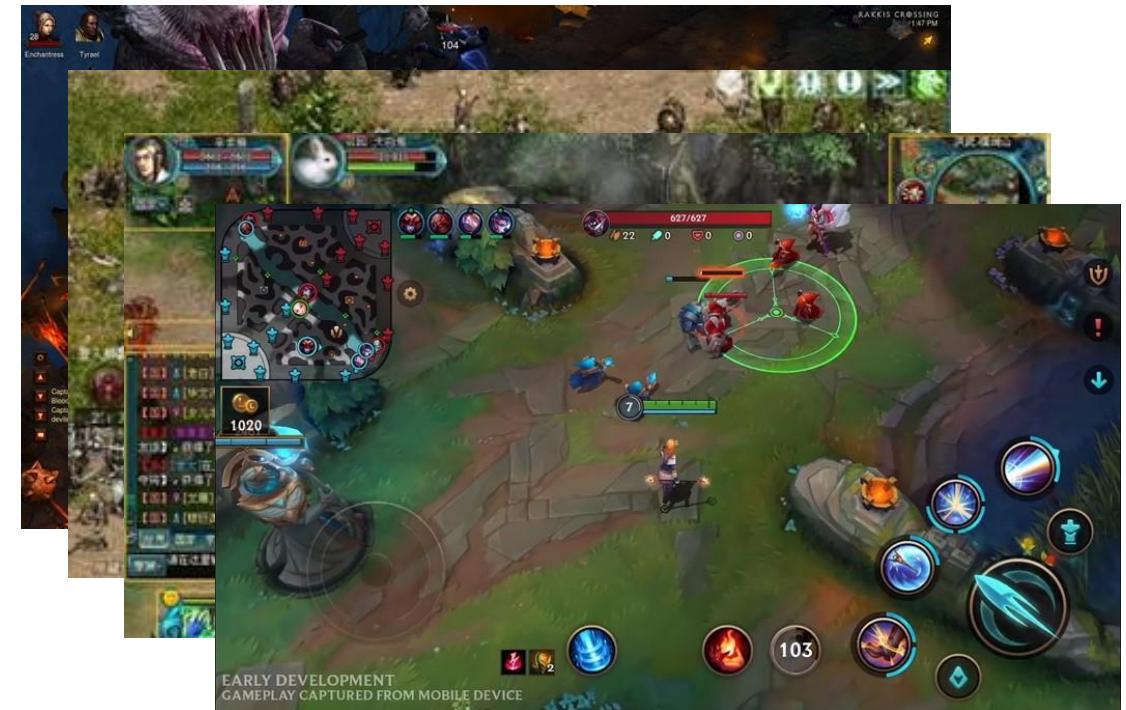
Introduction to Path Planning

What is Path Planning?

- **Path planning** (also known as the **navigation problem**) is computational problem to find a sequence of valid configurations that moves the object from the source to destination. The term is used in **aviation, robotics and computer games**.

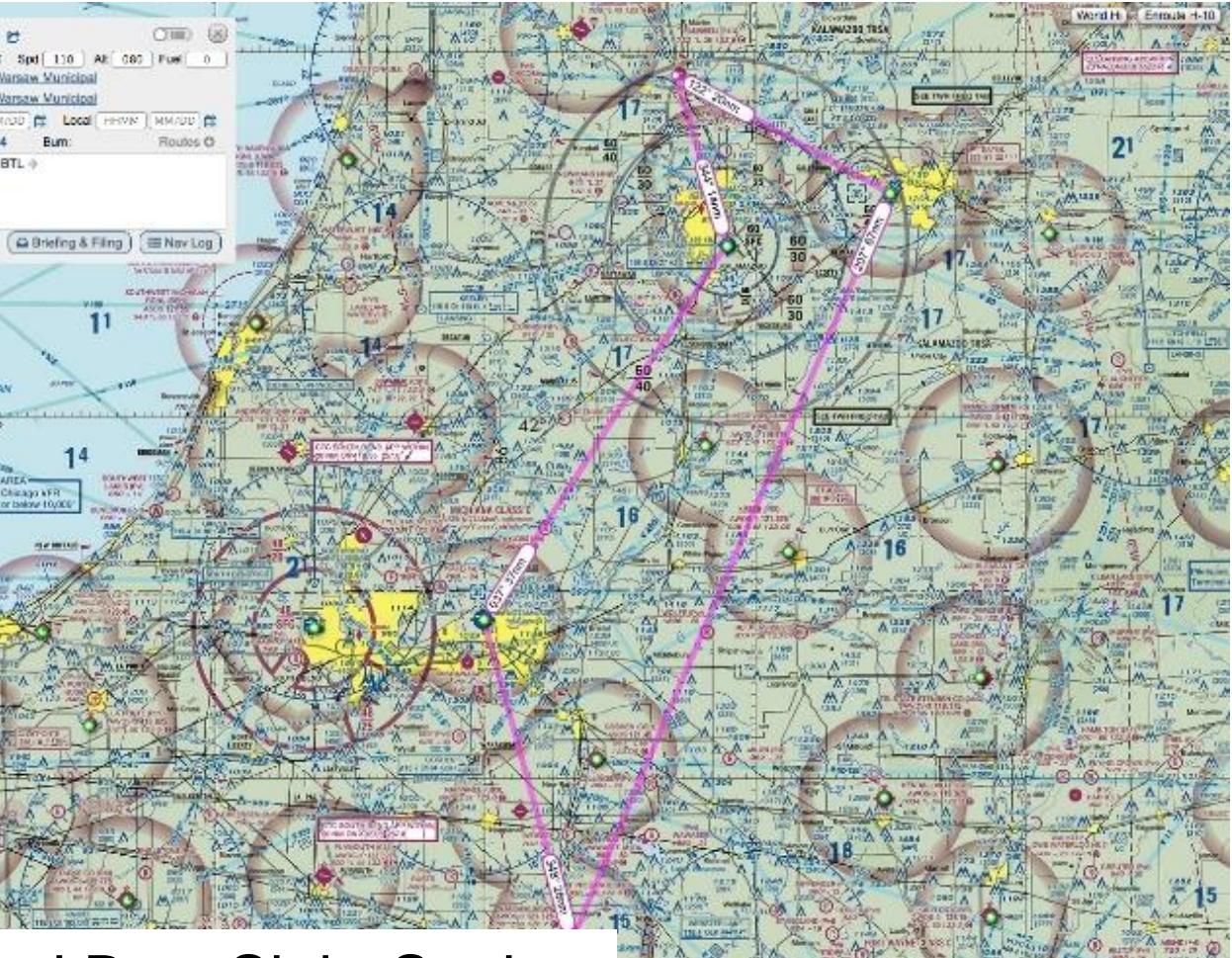
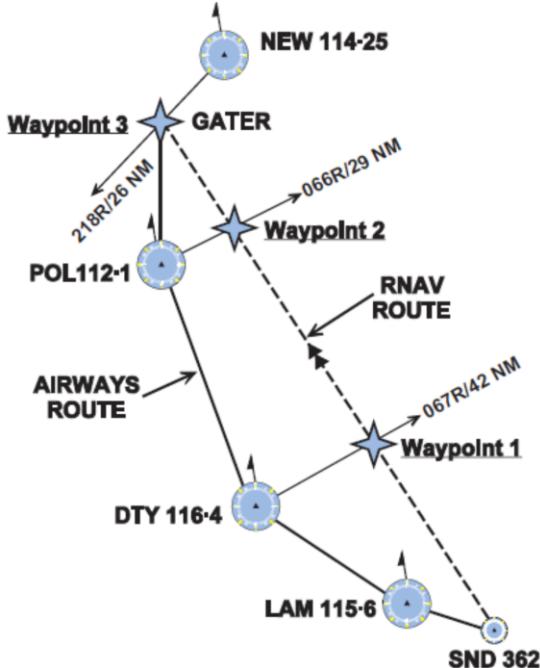


How to go from A to B considering factors!



How is Path Planning important to Aviation Engineering?

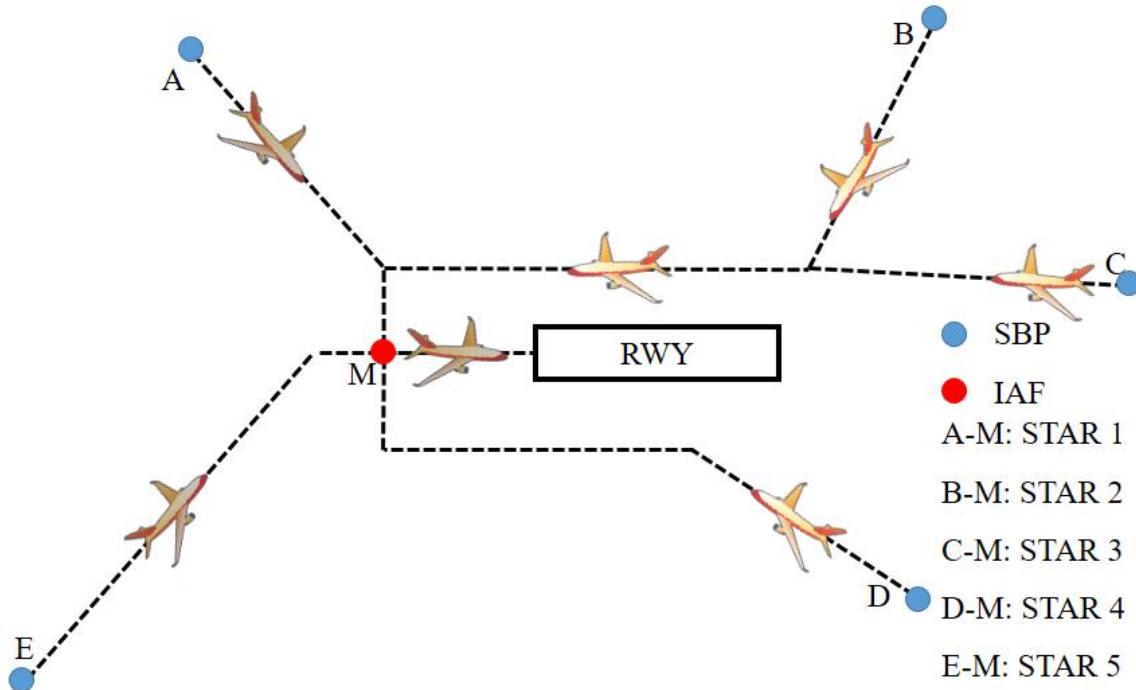
- Private pilots do the path plan before the flight to make sure the navigation aid is available



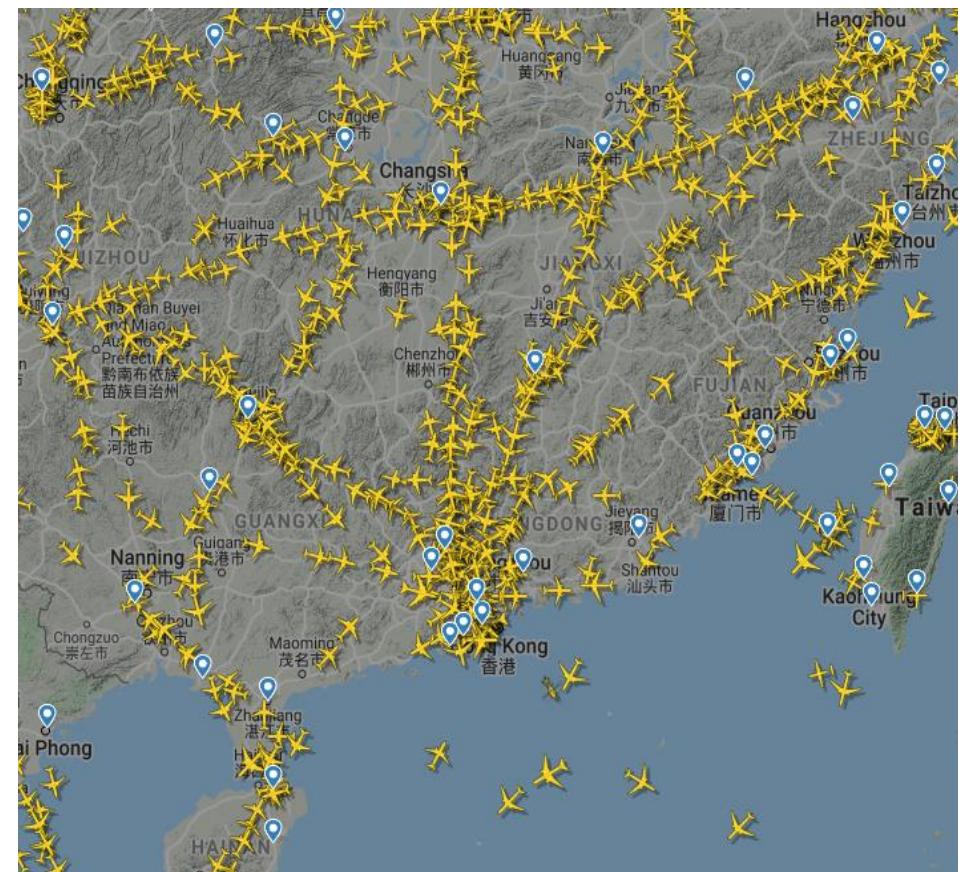
Objective: Safe and Best Sight Seeing

How is Path Planning important to Aviation Engineering?

- For ATC near airports, collaborative path planning is required to make the best use of the crowded airspace



Objective: Safe and least delay

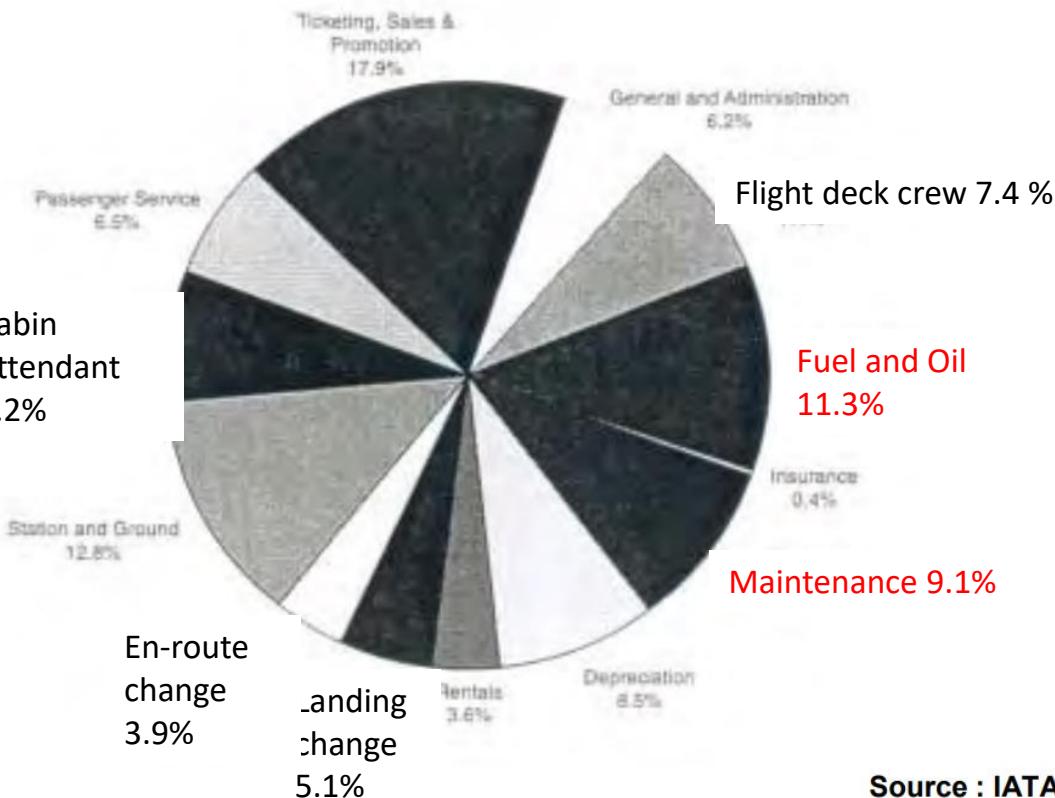


How is Path Planning important to Aviation Engineering?

- Commercial pilot follow the path that plan based on different cost index designed by airlines.

Objective: Safe and Minimum Cost

Figure 2. Distribution of operating costs



2.1 Trip cost

Without having to resort to complicated mathematics we can readily appreciate that the total cost of a specific trip is the sum of fixed and variable costs :

$$C = C_F \times \Delta F + C_T \times \Delta T + C_c$$

with C_F = cost of fuel per kg
 C_T = time-related cost per minute of flight
 C_c = fixed costs independent of time
 ΔF = trip fuel
 ΔT = trip time

In order to minimize C or the total trip cost we therefore need to minimize the variable cost :

$$C_F \times \Delta F + C_T \times \Delta T$$

Source : IATA

<https://ansperformance.eu/library/airbus-cost-index.pdf>

Cost-Index Published by Aircraft Manufacturer

Flight Operations Support & Line Assistance

getting to grips with the cost index

Issue II - May 1998

Customer Services

AIRBUS



3.1 A300/A310 Family

Considering, with good approximation, that the following range of time-related costs cover the maintenance cost difference between A300 and A310 as well as the cabin crew contingent (plus or minus two) difference, the following cost brackets result :

6 < Hourly maintenance cost < 12 (US\$/min)

+ 7 < Crew cost < 14 (US\$/min)

13 < Time-related cost < 26 (US\$/min)

NB : Crew composition = 2 cockpit crews + 8 (\pm 2) cabin crews.

In turn, the following cost index tables reflect these cost ranges for the A300 and for the A310.

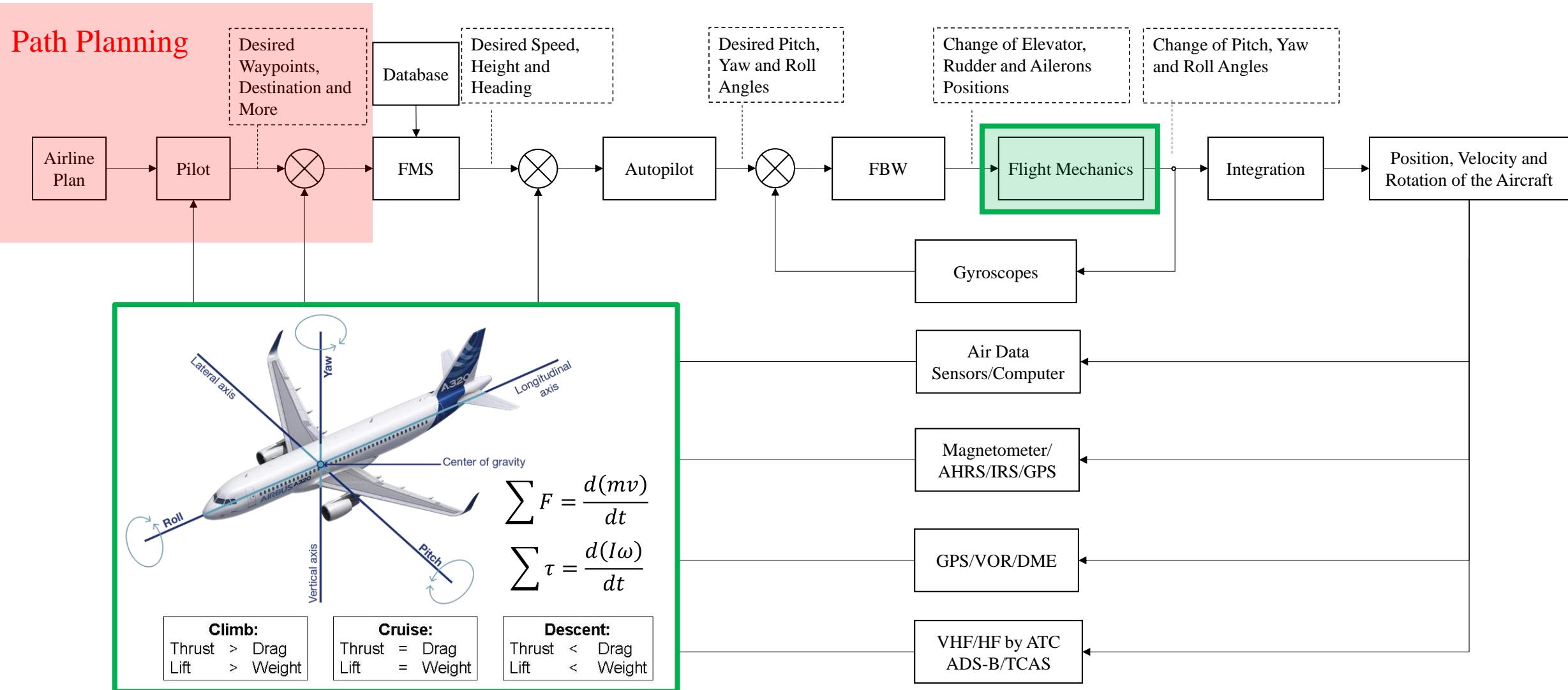
Table 1. A300/A310 cost index

(kg/min)
(Honeywell FMS)

TIME COST (US\$/min)\ FUEL COST (US\$/USG)	LOW	MEDIUM	HIGH
LOW < 0.7	65	85	100
MEDIUM 0.7 < < 0.9	50	65	80
HIGH > 0.9	40	55	65

<https://ansperformance.eu/library/airbus-cost-index.pdf>

Aircraft Operation in Flight Control System



How is the Freshman Project related to the AE programme study?

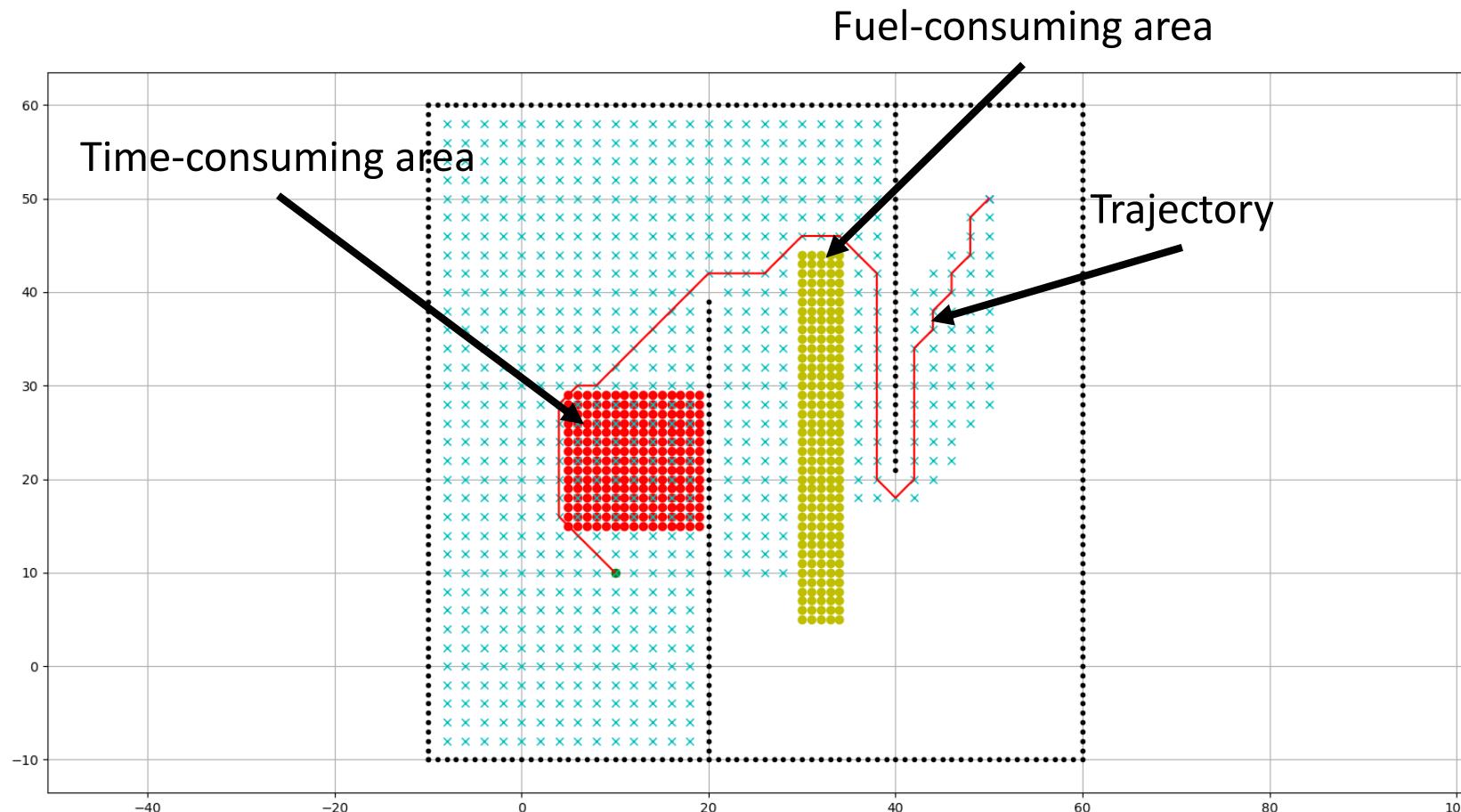
- Mathematics & Physics
- Computer Science
- Aeronautical and Aviation
- The plan should be planned considering the physical limitation (dynamics) of the aircraft

Year 1 (33 + 4 training credits)					
Semester 1 (15 + 2 training credits)		Semester 2 (18 +2 training credits)			
AAE2001	Introduction to Aircraft and Aviation Systems	AMA1120	Basic Mathematics II		
AMA1110	Basic Mathematics I	AP10006	Physics II		
AP10005	Physics I	APSS1L01	Tomorrow's Leaders		
ENG1003	Freshman Seminar for Engineering	ENG2003	Information Technology		
LCR I (English)		LCR II (English)			
		CAR I ^			
Healthy Lifestyle (non-credit bearing) ^					
IC2105 Engineering Communication and Fundamentals (4 training credits) <i>or</i> IC2133 Aircraft Manufacturing and Maintenance Fundamentals (4 training credits)					
Year 2 (30 + 3 training credits)					
Semester 1 (15 credits)		Semester 2 (15 + 3 training credits)			
AMA2111	Mathematics I	AMA2112	Mathematics II		
ENG2001	Fundamentals of Materials Science and Engineering / Biology / Chemistry	EE2902S	Fundamentals of Electrical and Electronic Engineering		
ENG2002	Computer Programming	ME33001	Mechanics of Materials		
ME23001	Engineering Mechanics	LCR III (Chinese)			
CAR II ^		CAR III ^			
IC381	Appreciation of Aircraft Manufacturing Processes (3 training credits)				
Year 3 (32 + 3 training credits)					
Semester 1 (17 + 1.5 training credits)		Semester 2 (15 + 1.5 training credits)			
AAE3001	Fundamentals of Aerodynamics	AAE3003	Aircraft Propulsion Systems		
AAE3002	Aircraft Structures and Materials	AAE4006	Flight Mechanics and Control Systems		
AAE3004	Dynamical Systems and Control	AAE4301	Avionics Systems		
ELC3531	Professional Communication In English For Engineering Students (2 credits)	AF3625	Engineering Economics		
CAR IV ^		ISE3009	Aviation Safety and Reliability		
Service Learning ^					
IC388 Aircraft Manufacturing and Maintenance practice (3 training credits)					

Path Planning

- Optimization Problem:
- To optimize a path that fulfilling all the constrains and by a set of certain criteria.
- Goal of this project, ***to select the best aircraft models with an optimized route that minimized the cost of the aircraft operation under given scenario.***
- ***Design the cost of the aircraft operation***
- ***Design an aircraft model (virtually) with different cost coefficients to fly safe and cheapest.***
- ***Design the path planning algorithm considering 3D, 2D + time, scenarios.***

Expected Outcome. Every Group have different scenarios



Model of Aircraft to select

Aircraft Model	C_F	ΔF	C_T	ΔT	C_c	ΔF_a	ΔT_a
PolyU-A380	1	1	2	5	10	0.2	0.2
PolyU-A381	1	1.5	3	5	10	0.3	0.4
PolyU-A382	1	2.0	4	5	10	0.4	0.5
PolyU-A383	1	2.5	5	5	10	0.5	0.1
AAE-Group 1	?	?	?	?	?	?	?
AAE-Group 2	?	?	?	?	?	?	?
AAE-Group 3	?	?	?	?	?	?	?
:	:	:	:	:	:	:	:

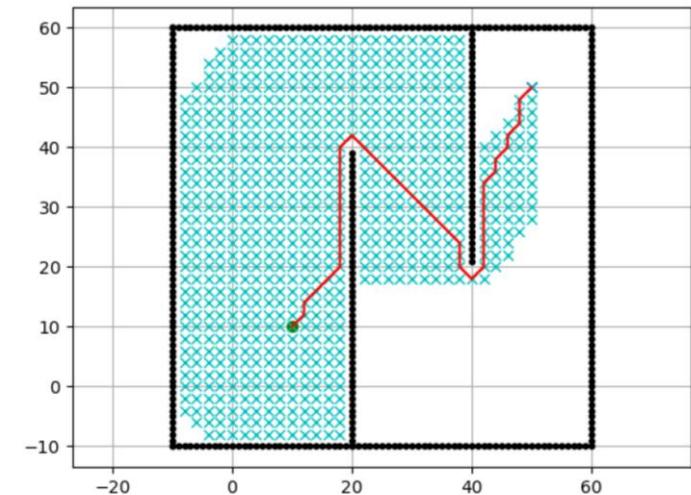
What you are expected to learn?

Academic level of algorithm designs

- Design of a path planning algorithm and aircraft model cost function
 - 2D path planning for simplicity

Make use of the **open-resource** to work on coding-project **remotely**.

- Programming and coding
 - Python
- Online coding collaboration
 - GitHub



In this project, students will be acted as

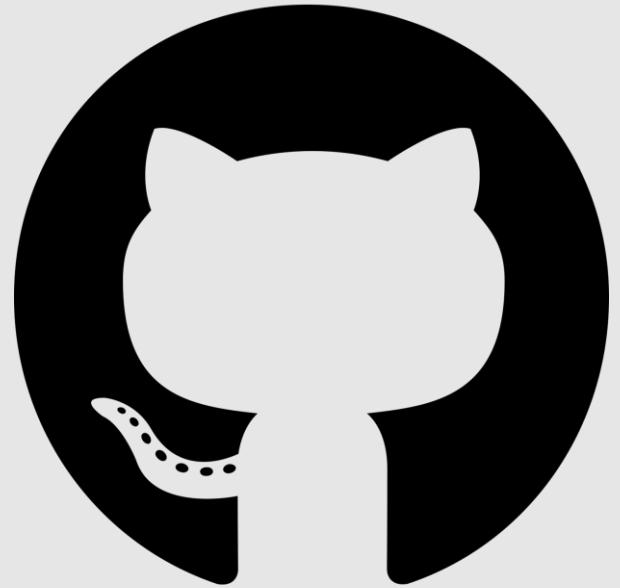
1. Group leader or members to collaborate on an online project,
2. R&D engineers to design and develop path-planning algorithm,
3. Project manager to present the designed code and prepare report.

Assessments

- (30%) Demonstration and Presentation
- (40%) Report & reflective essay – one report per group, with individual reflective essay
- (20%) Log sheet – one per student after the first week
- (10%) Performance/participation in in-class activities (Confidential peer evaluation)

Introduction to GitHub (Background)

What is GitHub?



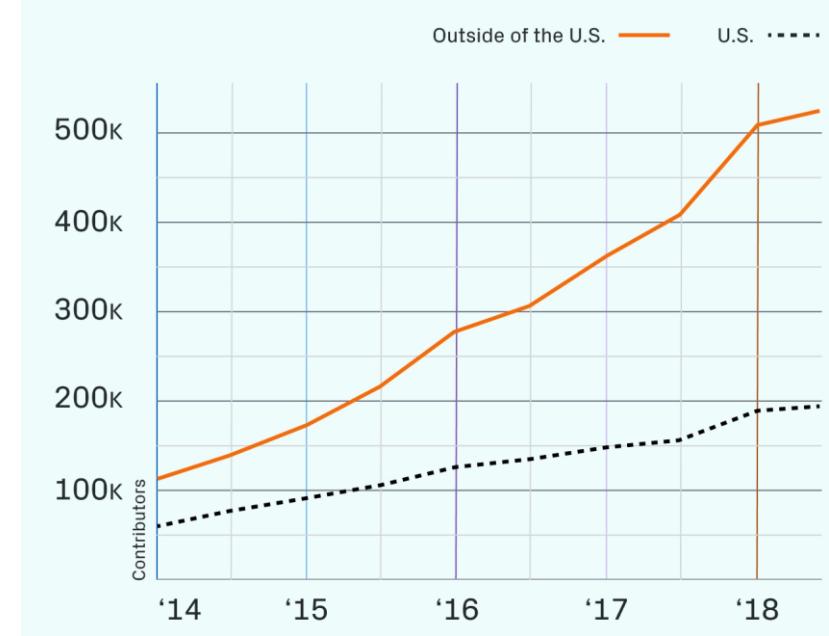
What is Github?



- A social network and platform for software developers
 - Over 65 million users
 - A place to Share, Communicate, Collaborate with others, especially programmers
-

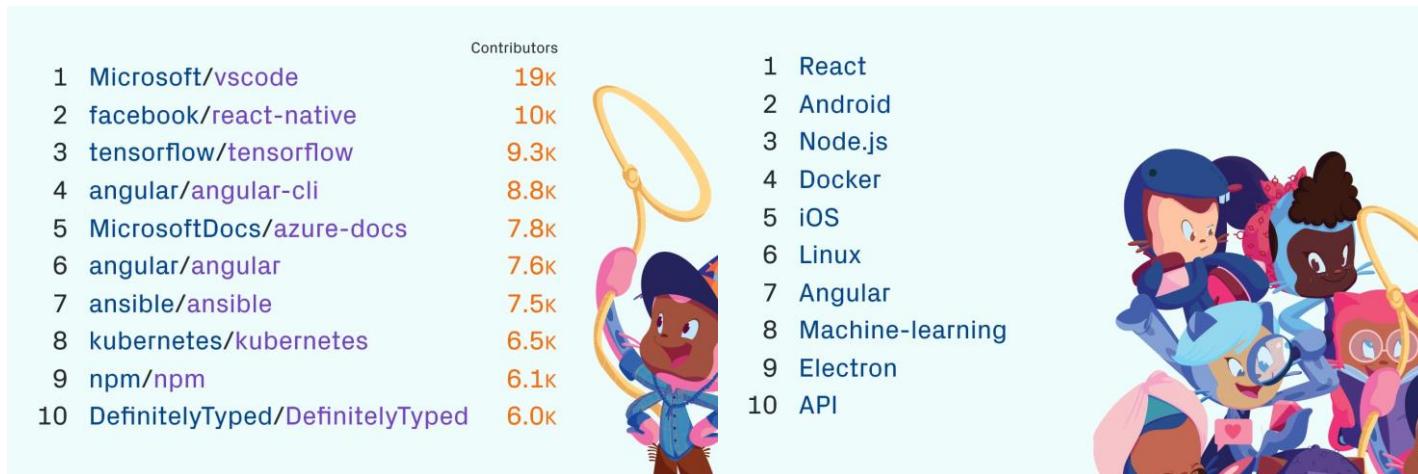
What's on GitHub

- Over 100 million live Projects
- New projects increasing over the years
- Showing the trend of using GitHub for Software development



Variety of GitHub

- Countless topics and projects available for the public
- Encompassing most popular topics nowadays



BIG GitHub Pages

Tesla, Inc.
Open source projects and samples from Microsoft

Microsoft
Open source projects and samples from Microsoft

Graphomer
This is the official implementation for "Do Transformers Really Perform Bad for Graph Representation?"

calculator
Windows Calculator: A simple yet powerful calculator that ships with Windows

v8-jsi
React Native V8 JSI adapter

ts-gyb
Generating native code interfaces from TypeScript

fixed-containers
C++ Fixed Containers

mongo-go-driver
Forked from mongo/mongo-go-driver
The Go driver for MongoDB

coreboot
Coreboot sources

linux
Linux sources

ansible puller

Google
Google ❤️ Open Source

<https://opensource.google/> [@GoogleOSS](#) opensource@google.com [Verified](#)

Repositories 2.1k **Packages** **People** 1.2k **Projects**

closure-compiler-npm
Package for managing and documenting closure-compiler for use via npm

it-cert-automation-practice
Google IT Automation with Python Professional Certificate - Practice files

CFU-Playground
Want a faster ML processor? Do it yourself! -- A framework for playing with custom opcodes to accelerate TensorFlow Lite for Microcontrollers (TFLM).

pytype
A static type analyzer for Python code

trax
Trax — Deep Learning with Clear Code and Speed

pigweed

BIG GitHub Pages

Boeing

Overview Repositories 5 Packages People Projects

Popular repositories

modular_navigation C++ ⭐ 6 📈 6	modular_cartographer C++ ⭐ 5 📈 5
cartographer C++ ⭐ 5 📈 3	math6d Python ⭐ 2 📈 1
image_tran C++	

Airbus Group

We design, manufacture and deliver industry-leading commercial aircraft, helicopters, military transports, satellites and launch vehicles
Toulouse <https://www.airbus.com> Verified

Overview Repositories 2 Packages People Projects

Popular repositories

scikit-decide AI framework for Reinforcement Learning, Automated Planning and Scheduling Python ⭐ 14 📈 10	ED247_LIBRARY Example of ED-247 standard implementation C++ ⭐ 13 📈 7
Repositories Find a repository... Type Language Sort	
ED247_LIBRARY Example of ED-247 standard implementation C++ ⭐ 13 📈 7 ⚡ 3 🔍 1	
scikit-decide AI framework for Reinforcement Learning, Automated Planning and Scheduling Python ⭐ 14 📈 MIT 📈 10 ⚡ 2 🔍 0	

[View all repositories](#)

NASA

Read about NASA's Open Data initiative here: <https://www.nasa.gov/open/> & Members Find Instructions here: <http://nasa.github.io/>
United States of America <https://github.com/nasa/nasa.github.io...> nasa-data@lists.arc.nasa.gov

Repositories 359 Packages People 42 Projects 1

Pinned repositories

[nasa.github.io](#)
<https://github.com/nasa/nasa.github.io/blob/master/docs/INSTRUCTIONS.md>

HTML ⭐ 238 📈 52

Find a repository... Type Language Sort

fprime

F - A flight software and embedded systems framework

raspberry-pi components real-time framework embedded cpp
nasa

C++ Apache-2.0 📈 981 ⚡ 8,370 ⚡ 68 (1 issue needs help) 🔍 7 Updated 34 minutes ago

cumulus-orca

Python 📈 7 ⚡ 7 ⚡ 0 🔍 1 Updated 1 hour ago

ow_autonomy

C++ 📈 4 ⚡ 13 ⚡ 0 🔍 0 Updated 2 hours ago

LHASA

R 📈 7 ⚡ 8 ⚡ 0 🔍 0 Updated 2 hours ago

earthdata-search

Earthdata Search is a web application developed by NASA EOSDIS to enable data discovery, search, comparison, visualization, and access across EOSDIS' Earth Science data holdings.

data-discovery hacktoberfest eosdis earthdata-search
JavaScript 📈 202 ⚡ 618 ⚡ 14 🔍 1 Updated 3 hours ago

Top languages

Python C C++ JavaScript
Jupyter Notebook

Most used topics

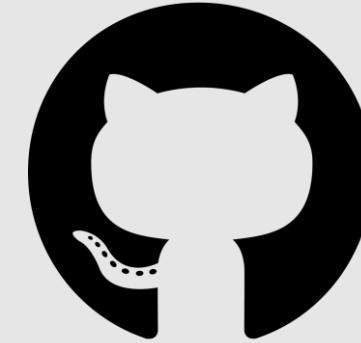
nasa cumulus nasa-cumulus
eosdis satellite

People 42

Developer Program Member

Report abuse

What's on GitHub? How is it related to AAE?



Flight control softwares for UAV

Ardupilot, PX4 and more

The image displays three GitHub repository pages side-by-side:

- ArduPilot / ardupilot**: A repository for ArduPilot flight control software. It features a banner with various UAV images, a star count of 6k, and several open pull requests. One prominent issue is "Move test_onboard_logging_generation outside autotest.py".
 - Tags: AllVehicles, BuildSystem, CI, good first issue
 - Issues: Fix Copter test which tests all frames in autotest
 - Issues: Plane: maximum altitude for FBWB and cruise mode
- PX4 / PX4-Autopilot**: A repository for PX4 Autopilot Software. It has a star count of 4.7k and a large list of tags related to UAV components and protocols.
 - Tags: uav, drone, ros, px4, pixhawk, uas, dronecode, autopilot, mavlink, autonomous, drones, dds, hacktoberfest, ugv, mavros, multicopter, qgroundcontrol, fixed-wing, fast-rtps, avoidance
- mavlink / qgroundcontrol**: A repository for a cross-platform ground control station. It has a star count of 1.9k and a brief description: "Cross-platform ground control station for drones (Android, iOS, Mac OS, Linux, Windows)".
 - Tags: qt, uav, drone, px4, pixhawk, uas, mavlink, ardupilot

About These Softwares

Ardupilot:

- Open source software suite
- Quadcopters, VTOL and more
- Cross platform
- Over 10 years of development and improvement



PX4:

- Also open source
- Works with QGC and MAVLink (Also available in GitHub)

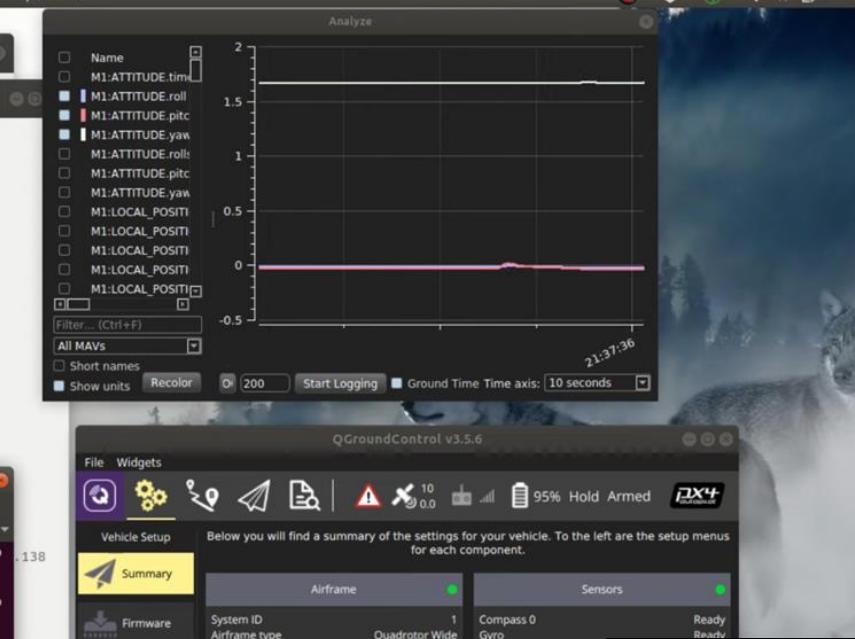
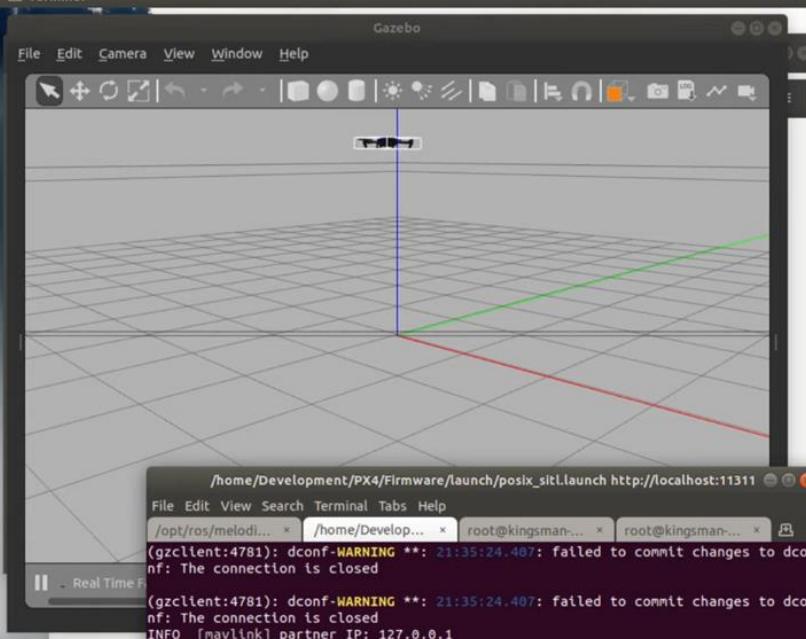


Activities

Terminal

Sun Feb 9, 21:37:36

95%



```
/home/Development/PX4/Firmware/launch posix_sitl.launch http://localhost:11311
File Edit View Search Terminal Tabs Help
/opt/ros/melodic... /home/Develop... root@kingsman... root@kingsman...
(gzclient:4781): dconf-WARNING **: 21:35:24.407: failed to commit changes to dconf: The connection is closed
(gzclient:4781): dconf-WARNING **: 21:35:24.407: failed to commit changes to dconf: The connection is closed
INFO [navlink] partner IP: 127.0.0.1
INFO [navlink] partner IP: 127.0.0.1
INFO [ecl/EKF] 468000: EKF aligned, (baro height, IMU buf: 22, OBS buf: 14)
INFO [ecl/EKF] 468000: reset position to last known position
INFO [ecl/EKF] 468000: reset velocity to zero
INFO [ecl/EKF] 644000: GPS checks passed (WGS-84 origin set)
INFO [vehicle.angular_velocity] updating filter, sample rate: 1000.000 Hz -> 250.000 Hz
INFO [vehicle.acceleration] updating filter, sample rate: 1000.000 Hz -> 250.000 Hz
INFO [ecl/EKF] 5188000: reset position to GPS
INFO [ecl/EKF] 5188000: reset velocity to GPS
INFO [ecl/EKF] 5188000: commencing GPS fusion
pxh>
pxh>
pxh> commander takeoff
pxh> INFO [commander] Takeoff detected
```

QGroundControl v3.5.6

File Widgets

Vehicle Setup

Summary

Below you will find a summary of the settings for your vehicle. To the left are the setup menus for each component.

Airframe		Sensors	
System ID	1	Compass 0	Ready
Airframe type	Quadrrotor Wide	Gyro	Ready
Vehicle	3DR Iris Quadrrotor	Accelerometer	
Firmware Version	1.11.dev		

INS

Radio		Flight Modes	
Roll	Setup required	Mode switch	
Pitch	Setup required	Flight Mode 1	
Yaw	Setup required	Flight Mode 2	
Throttle	Setup required	Flight Mode 3	
Aux1	Disabled	Flight Mode 4	
Aux2	Disabled	Flight Mode 5	

Firmware

Airframe

Sensors

Radio

Flight Modes

Power

Safety

Aviation Services Engineering

Logistics and Facility Management
and more

airport

Here are 148 public repositories matching this topic...

Language: All ▾

Sort: Best match ▾

Ysurac / FlightAirMap

Star 388

Code Issues Pull requests

Open source project displaying live aircrafts, ships or trackers on 2D/3D map. Browse through the data based on a particular aircraft, airline, airport, tracker or vessel to search through the database or see extensive statistics. Can use ADS-B in SBS1 format (dump1090, Radarcap...), VRS, VA (VATSIM, IVAO whazzup.txt, phpvms,...), ACARS (acarsdec, acarsdeco2), APRS, AIS as datasource.

tracker cesium crash metar airport airline flight ship vatsim aircraft iavo acars
glidernet phpvms notam modes ads-b sbs vessel 3d-map

Updated on Nov 25, 2020 TSQL

felix-dumit / FSDAirportFlipLabel

Star 83

Code Issues Pull requests

UILabel like old Airport flipping labels

ios label ios-animation airport

Updated on Mar 9, 2018 Objective-C

Ivysauro / CNRT

Star 74

Code Issues Pull requests Discussions

中国轨道交通数据库 (非技术类) - 另一角度看地铁/ Data base of China Rail Transit (Non-tech) - Another view of Rail Transit

bus metro payment railway china airport subway rail-transit

Updated 3 days ago SCSS

gravity-EDDS / EDDS-freeware-releases

Star 57

Code Issues Pull requests

Logistics and Facility Management

- Data analysis resources
- System modelling and simulations
- Logistical models

Machine learning for data analysis

- Faster analysis
- Potential extra self-learning for students
- Extremely beneficial to their careers

The screenshot shows a GitHub repository page for 'LogisticsPipes'. At the top, it displays '4 branches' and '11 tags'. Below the repository name, there's a list of commits from 'Michał-MK' with details like date, file changes, and descriptions. To the right, there are sections for 'Releases' (17), 'Packages' (No packages published), 'Contributors' (60), and 'Languages' (Java 84.4%, Kotlin 15.6%).

The screenshot shows a GitHub search results page for 'Machine Learning'. It includes a sidebar with repository statistics (339K repositories, 8M code, 271K commits, etc.) and a language section (Jupyter Notebook 132,549, Python 72,791, etc.). The main area shows search results for 'Machine Learning' with details like repository name, description, language, and update date.

Aeronautical Engineering

Material, aircraft designs, CFD and more

computational-fluid-dynamics

Here are 198 public repositories matching this topic...

Language: All ▾

Sort: Best match ▾

doyubkim / fluid-engine-dev

Star 1.1k

Code Issues Pull requests

Fluid simulation engine for computer graphics applications

c-plus-plus visual-studio sdk animation computer-graphics physics-engine computational-physics fluid-simulation-engine computational-fluid-dynamics

Updated on Apr 25 C++

CubbyFlow / CubbyFlow

Star 175

Code Issues Pull requests

Voxel-based fluid simulation engine for computer games

cplusplus cpp computer-graphics physics-engine computational-physics cpp17 fluid-simulation-engine computational-fluid-dynamics

Updated 6 days ago C++

AvtechScientific / ASL

Star 153

Code Issues Pull requests

Advanced Simulation Library - hardware accelerated multiphysics simulation platform.

crystallography gpgpu scientific-computing high-performance-computing design-space-exploration computational-fluid-dynamics virtual-sensing image-guided-surgery computer-aided-engineering

Updated on Dec 21, 2018 C++

loliverhennigh / Steady-State-Flow-With-Neural-Nets

Star 119

Code Issues Pull requests

A Tensorflow re-implementation of the paper Convolutional Neural Networks for Steady Flow Approximation

aircraft-design

Here are 21 public repositories matching this topic...

Language: All ▾

Sort: Best match ▾

JSBSim-Team / jsbsim

Star 394

Code Issues Pull requests Discussions

Open XML validation

14

bcoconni commented on Jan 12, 2019

JSBSim provides schemas for XML validation (JSBSim.xsd for flight models, JSBSimScript.xsd for script files and JSBSimSystem.xsd for system files) but they have not been updated for a while so they might reject perfectly valid XML files.

XML files can be tested with xmllint

> xmllint --noout --schema JSBSim.xsd file.xml [Read more](#)

bug help-wanted good-first-issue

Aero Sandbox

by Peter Sharpe



peterdsharpere / AeroSandbox

Star 236

Code Issues Pull requests

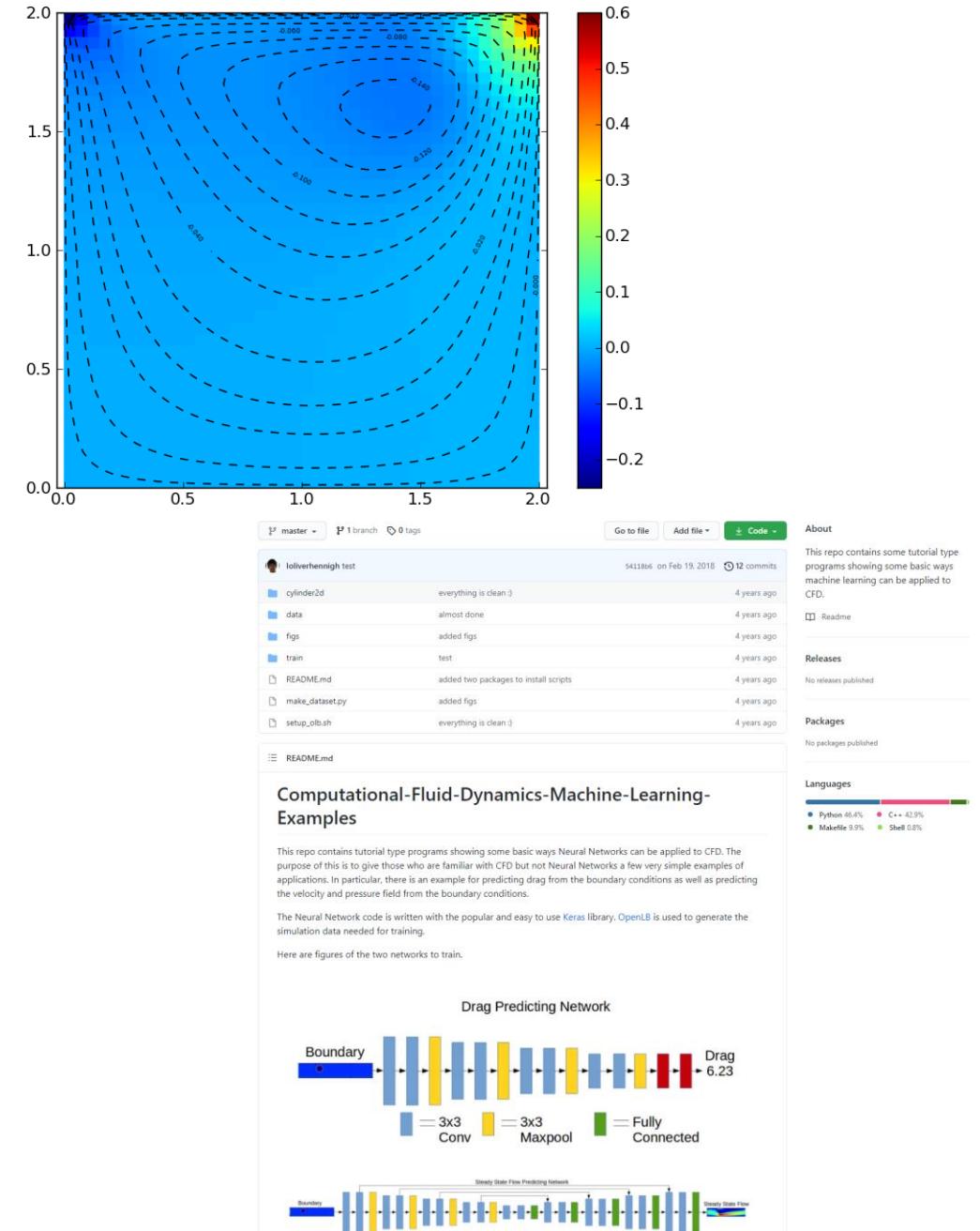
Aircraft design optimization made fast through modern automatic differentiation. Plug-and-play analysis tools for aerodynamics, propulsion, structures, trajectory design, and much, much more.

python analysis optimization aerospace automatic-differentiation airplane cfd aircraft aerodynamics vlm xfoil aircraft-design indra mdao aerodynamic-analysis 3d-panel

Updated 4 days ago Jupyter Notebook

CFD Python

- Full course for CFD Python
- Course Materials (Lectures and video)
- Software resources



Aircraft Maintenance

Propulsion, NDT and more

NASA Jet Propulsion Laboratory
A world leader in the robotic exploration of space
Pasadena, California, US <http://www.jpl.nasa.gov> github@jpl.nasa.gov

Overview Repositories 26 Packages 6 People 6 Projects

Pinned

open-source-rover
A build-it-yourself, 6-wheel rover based on the rovers on Mars!
Gnuplot 6.8k 1.1k

osr-rover-code
Code that runs on the Open Source Rover
Python 267 96

COVID-19-respirators
JPL designed 3D and tested printed respirators to help with the COVID-19 pandemic response.
G-code 104 18

osr-android-app
Android application used to control the Open Source Rover
Java 59 30

SPOC
A website showcasing SPOC (Soil Property and Object Classification), a deep learning-based terrain classifier for Mars rovers
HTML 6 1

spoc_lite
A light-weight, experimental terrain classifier for Mars rovers
C++ 7 4

Repositories

Find a repository... Type Language Sort

itslive
A NASA MEaSUREs project to provide automated, low latency, global glacier flow and elevation change datasets
Jupyter Notebook 4 MIT 0 3 0

LiveViewLegacy
Real-time tools for Imaging Spectroscopy Data
C++ 18 7 0 0 0

open-source-rover
A build-it-yourself, 6-wheel rover based on the rovers on Mars!
Gnuplot 6,785 Apache-2.0 1,104 34 (1 issue needs help) 8

itslive-projects
Jupyter Notebook 0 MIT 0 0 0

sstmp
Solar System Treks Mosaic Pipeline
Python 4 Apache-2.0 0 21 0

jsd
Just SOEM Drivers

Example on NDT

- Resources on the whole conference
 - 20th World Conference on Non-Destructive Testing
- Documentation and software resources
- Applying LSTM to NDT

master 3 branches 0 tags Go to file Add file Code

vewald Merge branch 'master' of https://github.com/xaviergoby/LSTMforSHM 0f00291 on Oct 21, 2020 67 commits

File	Description	Time Ago
configs_and_settings	recent work	10 months ago
literature	non-code commit	13 months ago
results	Merge branch 'master' of https://github.com/xaviergoby/LSTMforSHM	10 months ago
src	recent work	10 months ago
.gitignore	recent work	10 months ago
LICENSE	Initial commit	2 years ago
README.md	Revert "Results Update"	13 months ago
__init__.py	1st commit of og Vicent lstm & my data_loading script w/ data and labels	2 years ago
call_mainscript.py	Update results	10 months ago
main.py	recent work	10 months ago
main_v2.py	Merge branch 'master' of https://github.com/xaviergoby/LSTMforSHM	10 months ago
main_v3_xav.py	recent work	10 months ago
p2atR.jpg	non-code commit	13 months ago
settings.py	recent work	10 months ago

Readme MIT License

Releases No releases published

Packages No packages published

Contributors 2 xaviergoby Alexander Xavier O'Rour... vewald

Environments 1 github-pages Active

Languages Python 100.0%

About Application of LSTM network for Structural Health Monitoring & Non-Destructive Testing

xaviergoby.github.io/convlstm-compu...

computer-vision timeseries tensorflow keras cnn waves lstm supervised-learning classification shm ultrasonic-sensor ndt structural-engineering structural-analysis convlstm structural-health-monitoring lambwaves pzt aircraft-inspection non-destructive-testing

Sequential Modelling in Data-Driven Approach for Structural Health Monitoring by Recurrent Convolutional Neural Networks

Conference: 20th World Conference on Non-Destructive Testing
Location & Date: South Korea, Seoul - June 2020
Co-authors: Ewald V., Goby X., Groves R.M. & Benedictus R.
Labarotory: TU Delft Aerospace NDT Lab

Usage Instruction

In order to make use of this project all you need mainly be concerned with is the main.py Python script. In it you shall (hopefully) find yourself a more than sufficient amount of documentation in order to understand and be able to make use of it!

Dev Progress Log-Journal, Data Characteristics & Background Information

Notes:

Navigation and Positioning

RTKLIB, VINS, ORB_SLAM3, ROS and more

The image displays three GitHub repository pages side-by-side:

- UZ-SLAMLab / ORB_SLAM3**: This page shows the repository's code history. The master branch has 49 commits from richard-elvira. The commits are dated from April 2021 to March 2020. The repository includes sections for About, Releases, Packages, and Contributors.
- tomojitakasu / RTKLIB**: This page shows the repository's code history. The master branch has 69 commits from tomojitakasu. The commits are dated from January 2018 to January 2013. The repository includes sections for About, Releases, Packages, and Contributors.
- HKUST-Aerial-Robotics / VINS-Mono**: This page shows the repository's code history. The master branch has 76 commits from shaozu. The commits are dated from March 2019 to March 2017. The repository includes sections for About, Releases, Packages, and Contributors.

Aviation

Powered By GitHub

aviation

Here are 368 public repositories matching this topic...

Language: All Sort: Best match



Generalized Aviation™

[generalized-intelligence / GAAS](#) Star 1.6k

Code Issues Pull requests

GAAS is an open-source program designed for fully autonomous VTOL(a.k.a flying cars) and drones. GAAS stands for Generalized Autonomy Aviation System.

aviation uav drone flight-controller flight lidar autonomous drones autonomous-quadcopter autonomous-driving autonomous-vehicles vtol flying-car evtol hd-map e-vtol

Updated on Oct 25, 2021 C++

[cyoung / stratus](#) Star 842

Code Issues Pull requests

Aviation weather and traffic receiver based on RTL-SDR.

aviation weather traffic rtl-sdr stratus

Updated 6 days ago C

[szpaider / RTLSDR-Airband](#) Star 469

Variety of resources

All free on GitHub

aircraft

Here are 287 public repositories matching this topic...

Language: All ▾ Sort: Best match ▾

wiedehopf / [tar1090](#) ★ Star 582

Provides an improved webinterface for use with ADS-B decoders readsb / dump1090-fa

sdr rtl-sdr webinterface aircraft ads-b adsb rtl-sdr 1090 readsb 1090mhz

Updated 3 days ago JavaScript

JSBSim-Team / [jsbsim](#) ★ Star 504

Open XML validation 16

bcoconni commented on Jan 12, 2019

JSBSim provides schemas for XML validation (`JSBSim.xsd` for flight models, `JSBSimScript.xsd` for script files and `JSBSimSystem.xsd` for system files) but they have not been updated for a while so they might reject perfectly valid XML files.

XML files can be tested with `xmllint`

```
> xmllint --noout --schema JSBSim.xsd file.xml
```

Read more

bug help wanted good first issue

robin-shaun / [XTDrone](#) ★ Star 418

Code Issues Pull requests

UAV Simulation Platform based on PX4, ROS and Gazebo

px4 ros px4 gazebo aircraft self-driving



What Will You will Learn and Experience?

- Pull Request (Basic Features)
 - Allowing leaders to **double check** modifications done by teammates
- Code Synchronization (Version Control on Web, VS, MATLAB etc)
 - Ensuring everyone is working on the **latest** code version
- Self-learning using GitHub resources
- Look for interesting issues on GitHub

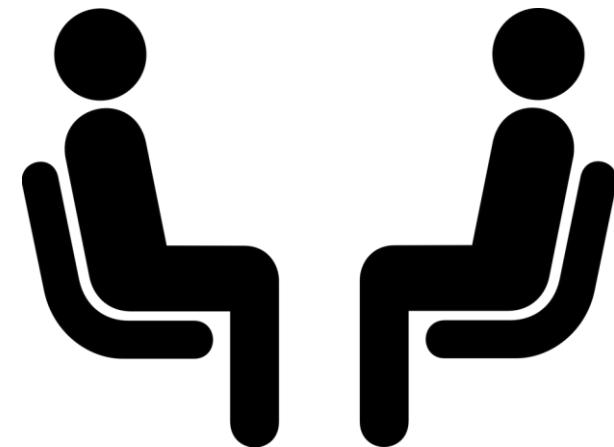


For Your Career

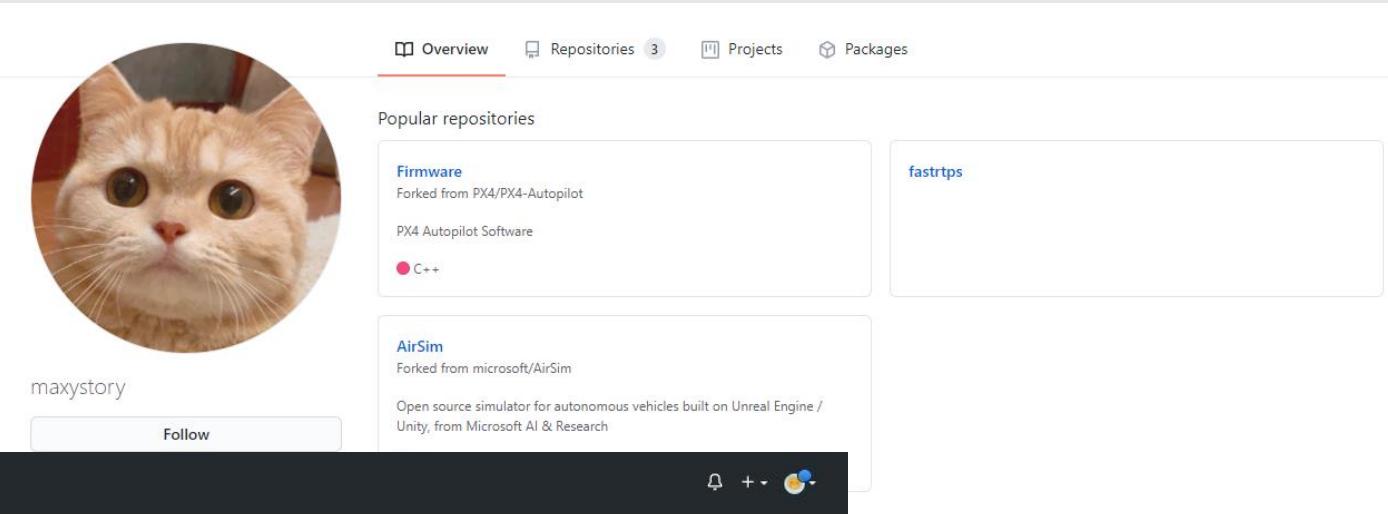


GitHub Facilitated Job Hunting

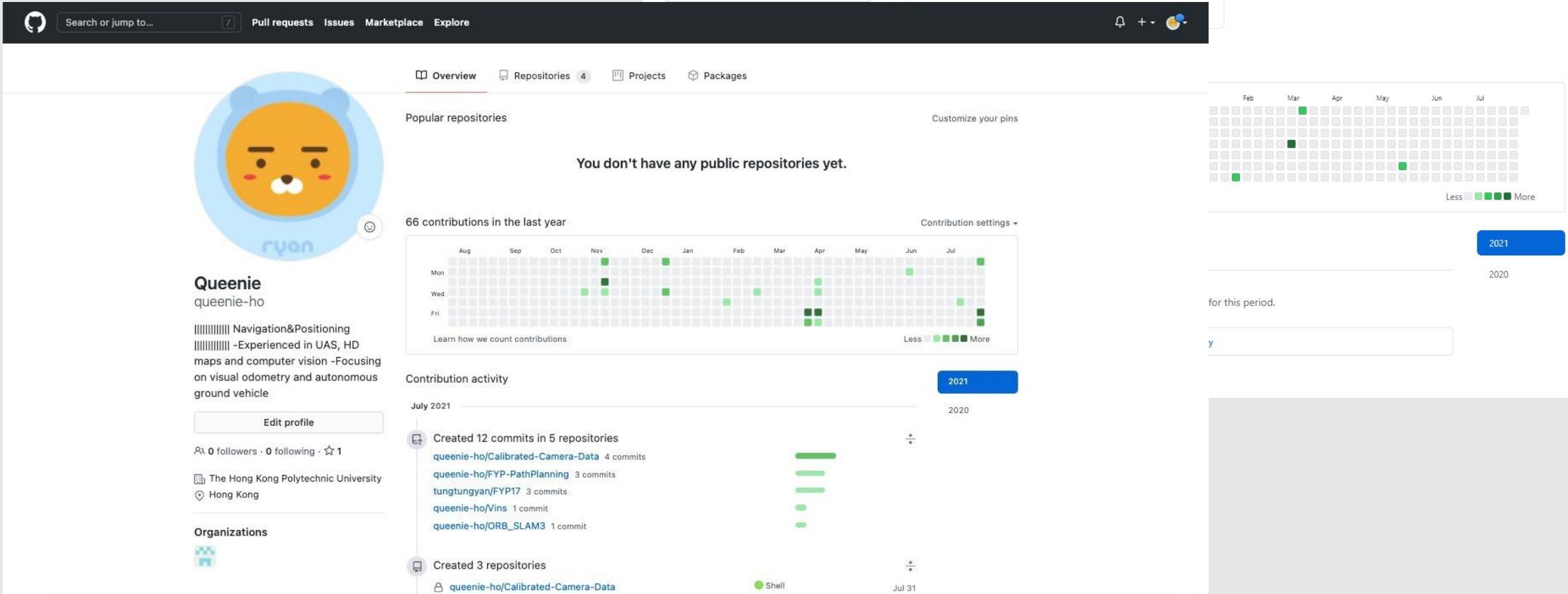
- A online digital profile for students
 - Indicating interests
 - Show past work
 - Roadmap of students' development career
- Let people know that you are looking for a job
- During job hunting
 - Better first impression
 - Better understanding
 - Appears more enthusiastic



Develop Personal Profiles Starting from AAE UG



The screenshot shows a GitHub profile for the user 'maxystory'. The profile picture is a close-up of an orange cat's face. Below the picture, the username 'maxystory' is displayed, along with a 'Follow' button. The top navigation bar includes links for Overview, Repositories (3), Projects, and Packages. The 'Popular repositories' section lists 'Firmware' (forked from PX4/PX4-Autopilot) and 'AirSim' (forked from microsoft/AirSim). The 'Firmware' repository is described as 'PX4 Autopilot Software' and is tagged with 'C++'. The 'AirSim' repository is described as an 'Open source simulator for autonomous vehicles built on Unreal Engine / Unity, from Microsoft AI & Research'.



The screenshot shows a GitHub profile for the user 'queenie-ho'. The profile picture is a cartoon character with a yellow face and blue ears. Below the picture, the username 'queenie-ho' is displayed, along with a 'Follow' button. The top navigation bar includes links for Overview, Repositories (4), Projects, and Packages. The 'Popular repositories' section displays a message: 'You don't have any public repositories yet.' The 'Contribution settings' section shows a heatmap of contributions over the last year, indicating activity in August, September, November, December, January, February, March, April, May, June, and July. The 'Contribution activity' section shows a timeline from July 2021 to Jul 31, 2021, listing contributions to repositories like 'queenie-ho/Calibrated-Camera-Data', 'queenie-ho/FYP-PathPlanning', 'tungtungyan/FYP17', 'queenie-ho/Vins', and 'queenie-ho/ORB_SLAM3'. The 'Edit profile' button is visible on the left side of the profile page.



B.X.W
baaixw

Follow

Visual SLAM GNSS

At 8 followers · 15 following · ⭐ 42

Intelligent Positioning and Navigation L...
Hong Kong

Achievements



Block or Report

Overview Repositories 33 Projects Packages

Popular repositories

remoteSensing2020

Improved VINS based on the adaptive covariance and adaptive M-estimator

C++ ⭐ 4 2

ios_logger

Forked from Vanvarilos_ios_logger

Application for camera and sensor data logging (IOS)

Objective-C++ ⭐ 1

vins-application

Forked from engcang/vins-application

VINS-Mono and Fusion application of different sets of cameras and imu on different board including desktop and jetson xavier

C++ ⭐ 1

catkin

Original vins-fusion for validation some data. The related path has been revised to ourself path.

C++ ⭐ 1 1

CV_GNSS

Forked from weisongwen/CV_GNSS

CV Aided GNSS

C++

tutorials

128 contributions in 2020



Activity overview

Contributed to weisongwen/researchTools.

2021
2020
2019
2018

Follow



Darren Wong

DarrenWong

Follow

move fast

At 18 followers · 16 following · ⭐ 95

Hong Kong

darrenwongf@gmail.com

Achievements



Organizations



Block or Report

Overview Repositories 24 Projects Packages

Pinned

e3372-web-management

Getting HUAWEI E3372 info with official API, such as device info, data switch and send sms etc

CSS ⭐ 8 2

protobuf-over-nanomsg-example

Protobuf over nanomsg (C++ as server, and nodejs as client)

C++

weisongwen/UrbanNavDataset

UrbanNav: an Open-Sourcing Localization Data Collected in Asian Urban Canyons, Including Tokyo and Hong Kong

139 32

98 contributions in the last year



IPNL @IPNL-POLYU @HKUST-Aerial-Robotics

Activity overview

Contributed to IPNL-POLYU/UrbanNavDataset,
IPNL-POLYU/ipnl-sensor-kit,
DarrenWong/benchmark_lo
and 5 other repositories

Code review

100% Commit Issues

2021

2020

2019

2018

2017

2016

2015

2014

2013

Up to Phd Studies



Seph Soliman
scarlac

[Follow](#)

Tattoodo, Bitbucket, Konstellation. Software entrepreneur, developer and true full stack developer.

64 followers · 7 following · 71 contributions

[Tesla](#)
San Francisco
<https://www.seph.dk>

Achievements



Organizations


Block or Report

Pinned

- js-stopwatch**
JavaScript Stopwatch class. Output can be controlled using a simple callback.
JavaScript ⭐ 24 ⚡ 7
- drag-check-js**
Library for checking multiple checkboxes by click-and-dragging over them. Paint your selection!
JavaScript ⭐ 21 ⚡ 9
- ClamshellOpen**
App to allow running your laptop in clamshell mode with an open lid for Lion (10.7) and Mountain Lion (10.8)
Objective-C ⭐ 5
- chargenow**
See DriveNow EVs in your area that needs charging. First react project.
JavaScript ⭐ 1
- d3d-strategy**
Exam assignment to write a C# program. I chose to write a Direct 3D Strategy game. Graphics for this game was borrowed from C&C: Red Alert. For legal reasons, they are not included in the repository.
C# ⭐ 1
- lолpause**
MacOS utility app to pause League of Legends GUI Flash client while game is active to lower CPU consumption
Objective-C ⭐ 1

44 contributions in 2021





Jonathan Hall
flimzy

You can also find me on GitLab: <https://gitlab.com/flimzy>

[Follow](#)

139 followers · 32 following · 45 contributions

Amsterdam, NL
<https://jhall.io/>
[@DevOpsHabits](#)

Highlights

* Arctic Code Vault Contributor

Organizations

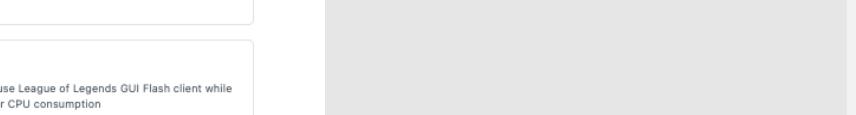




Popular repositories

- anki**
Go library to read Anki *.apk files
Go ⭐ 18 ⚡ 3
- go-pouchdb**
GopherJS bindings for PouchDB ⚠ NOTICE ⚡ this package has been superseded by https://github.com/go-kivik/kivik
Go ⭐ 13 ⚡ 1
- onload**
Onload handler for GopherJS without the bloat of jQuery
Go ⭐ 10 ⚡ 1
- go-sql.js**
GopherJS bindings for SQL.js
Go ⭐ 9
- minimal-pairs**
Tool for finding minimal pairs given a corpus of words
HTML ⭐ 5 ⚡ 1
- jsblob**
GopherJS bindings for JavaScript Blob objects
Go ⭐ 3

977 contributions in the last year





Jonathan Hall
flimzy

You can also find me on GitLab: <https://gitlab.com/flimzy>

[Follow](#)

139 followers · 32 following · 45 contributions

Amsterdam, NL
<https://jhall.io/>
[@DevOpsHabits](#)

Highlights

* Arctic Code Vault Contributor

Organizations





Popular repositories

- anki**
Go library to read Anki *.apk files
Go ⭐ 18 ⚡ 3
- go-pouchdb**
GopherJS bindings for PouchDB ⚠ NOTICE ⚡ this package has been superseded by https://github.com/go-kivik/kivik
Go ⭐ 13 ⚡ 1
- onload**
Onload handler for GopherJS without the bloat of jQuery
Go ⭐ 10 ⚡ 1
- go-sql.js**
GopherJS bindings for SQL.js
Go ⭐ 9
- minimal-pairs**
Tool for finding minimal pairs given a corpus of words
HTML ⭐ 5 ⚡ 1
- jsblob**
GopherJS bindings for JavaScript Blob objects
Go ⭐ 3

977 contributions in the last year





Jonathan Hall
flimzy

You can also find me on GitLab: <https://gitlab.com/flimzy>

[Follow](#)

139 followers · 32 following · 45 contributions

Amsterdam, NL
<https://jhall.io/>
[@DevOpsHabits](#)

Highlights

* Arctic Code Vault Contributor

Organizations






Popular repositories

- anki**
Go library to read Anki *.apk files
Go ⭐ 18 ⚡ 3
- go-pouchdb**
GopherJS bindings for PouchDB ⚠ NOTICE ⚡ this package has been superseded by https://github.com/go-kivik/kivik
Go ⭐ 13 ⚡ 1
- onload**
Onload handler for GopherJS without the bloat of jQuery
Go ⭐ 10 ⚡ 1
- go-sql.js**
GopherJS bindings for SQL.js
Go ⭐ 9
- minimal-pairs**
Tool for finding minimal pairs given a corpus of words
HTML ⭐ 5 ⚡ 1
- jsblob**
GopherJS bindings for JavaScript Blob objects
Go ⭐ 3

977 contributions in the last year



Making GitHub Part of your Work

Now you have an empty GitHub profile. How do you make it shine?

The following tips are roughly organized according to effort. Practically anyone can implement at least some of them. The later suggestions will be more discretionary, depending on your interests and time.

Star interesting projects

Whenever you run across a GitHub project that piques your interest, "star" it. Your starred projects appear on your public profile, and if nothing else, they provide recruiters and hiring managers an indication of what sorts of projects you find interesting.

Follow interesting people

GitHub also allows you to follow interesting people, and these people will appear on your public profile, as well. In addition to signaling to the world whom you find interesting, when people you follow make contributions to their projects, you will



Introduction to GitHub Operations

Mutual editing on a document with collaborators remotely?

 acceptance	26/3/2021 11:58 AM	File folder	
 final submission	2/8/2021 11:38 AM	File folder	
 ieee_taes_novatel_heatmap	29/7/2020 2:44 PM	File folder	
 My EndNote Library.Data	20/7/2020 9:17 PM	File folder	
 My EndNote Library.enl.unzipped	14/7/2020 9:26 AM	File folder	
 revision 1	28/12/2020 5:31 PM	File folder	
 Artical file (single column).docx	24/8/2020 3:55 PM	Microsoft Word D...	4,036 KB
 Article Processing Charges.pdf	19/4/2021 9:33 AM	Adobe Acrobat D...	118 KB
 cover letter.docx	28/7/2020 12:26 PM	Microsoft Word D...	17 KB
 figure.pptx	22/6/2020 11:11 AM	Microsoft PowerP...	7,122 KB
 ieee_taes_3dma_rtk (20200612 Ivan).docx	22/6/2020 10:12 AM	Microsoft Word D...	4,103 KB
 ieee_taes_3dma_rtk (20200622 GH).docx	22/6/2020 2:52 PM	Microsoft Word D...	4,112 KB
 ieee_taes_3dma_rtk (20200709 WS).docx	9/7/2020 3:38 PM	Microsoft Word D...	4,294 KB
 ieee_taes_3dma_rtk (20200714 LT).docx	15/7/2020 5:47 PM	Microsoft Word D...	4,306 KB
 ieee_taes_3dma_rtk (20200722 lucy).docx	23/7/2020 4:49 PM	Microsoft Word D...	7,319 KB
 ieee_taes_novatel_heatmap.rar	29/7/2020 2:42 PM	WinRAR archive	13,851 KB
 Manuscript_two_columns.docx	13/10/2020 1:45 PM	Microsoft Word D...	4,030 KB
 My EndNote Library.enl	8/1/2021 10:42 AM	EndNote Library	141 KB
 References.docx	14/7/2020 9:16 AM	Microsoft Word D...	18 KB

Download files from various channels, such as mail, whatsapp, etc?

Can we edit the documents online ? Google, Microsoft has the solutions!

Can we edit the “code” online? **Github**

Archive your coding online (in the cloud)

Pull

Pull: *To update local branch with remote, update all remote tracking branches*

Video: **VSC-Git Basic Operations**

Cloud repository (project)

Clone : *To download a repository to your local machine*

Clone/
Fetch

Local repository (project)

Fetch : *To pull a branch to your local machine*

Push

Push: *To upload the commit made on a local branch to GitHub*

Check out to: *Switch to a specific branch*

Open file and
Check out to

Code and Debugger

Commit

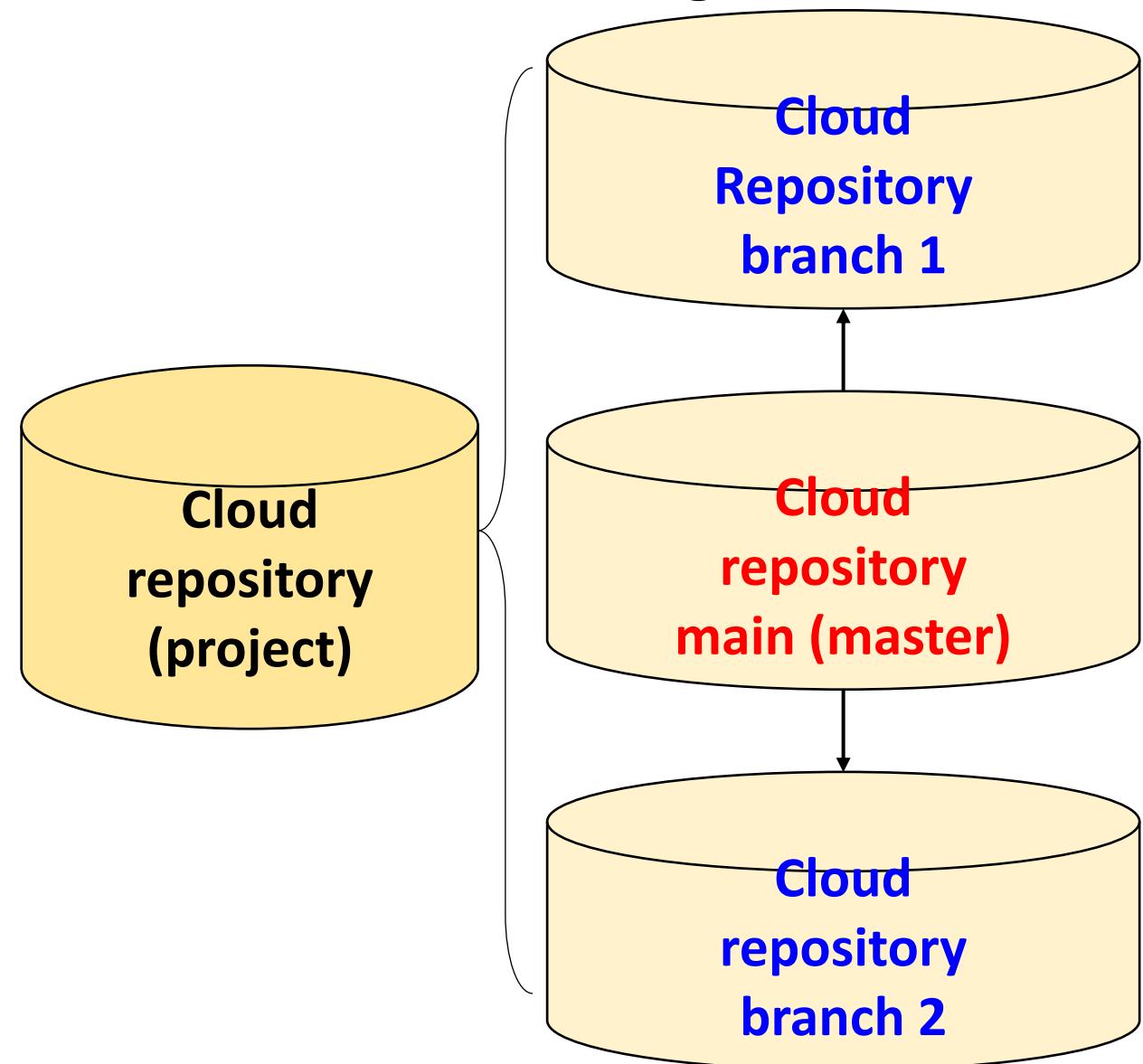
New Version

Add (Stage)

Commit: *To create a snapshot of the repository*

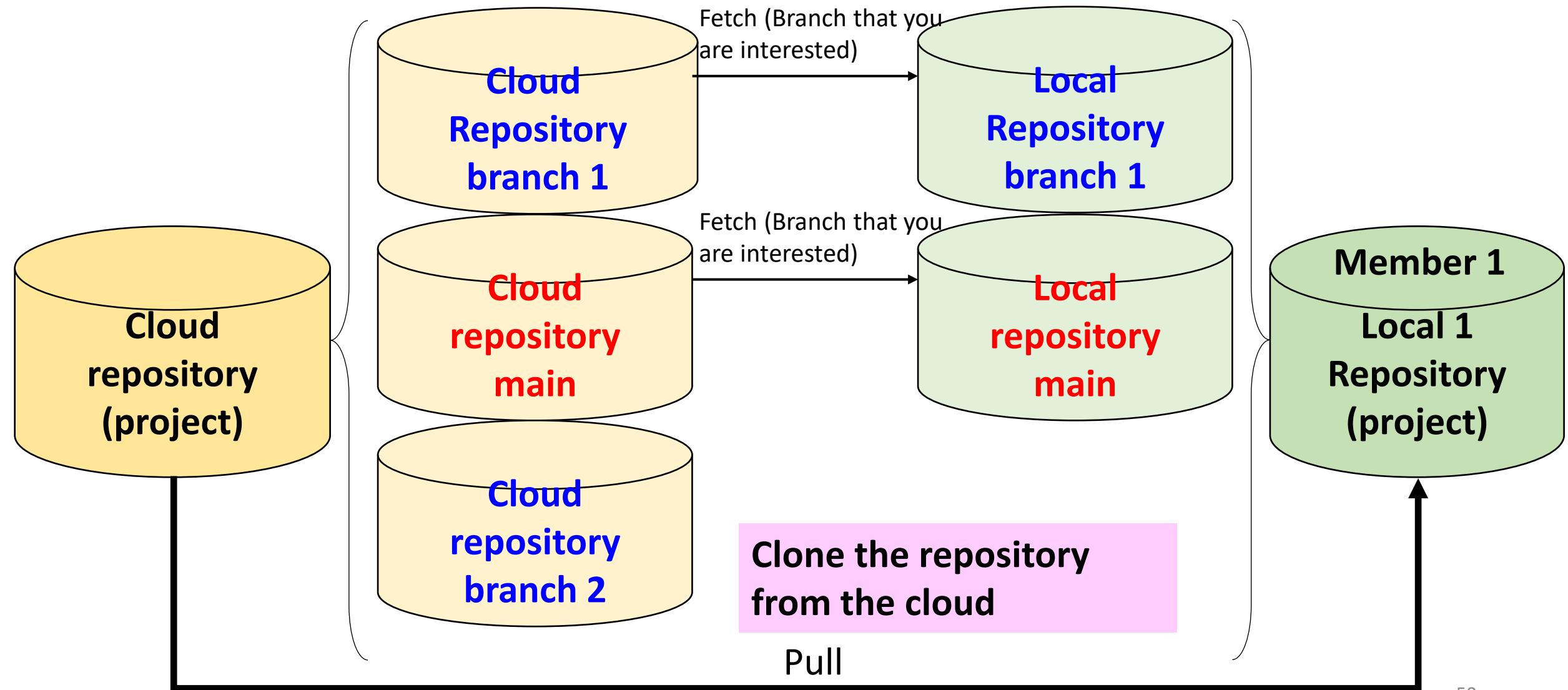
Add (Stage): *To add files or segments in the next commit*

Collaborative coding online – Generate Branch (Member)

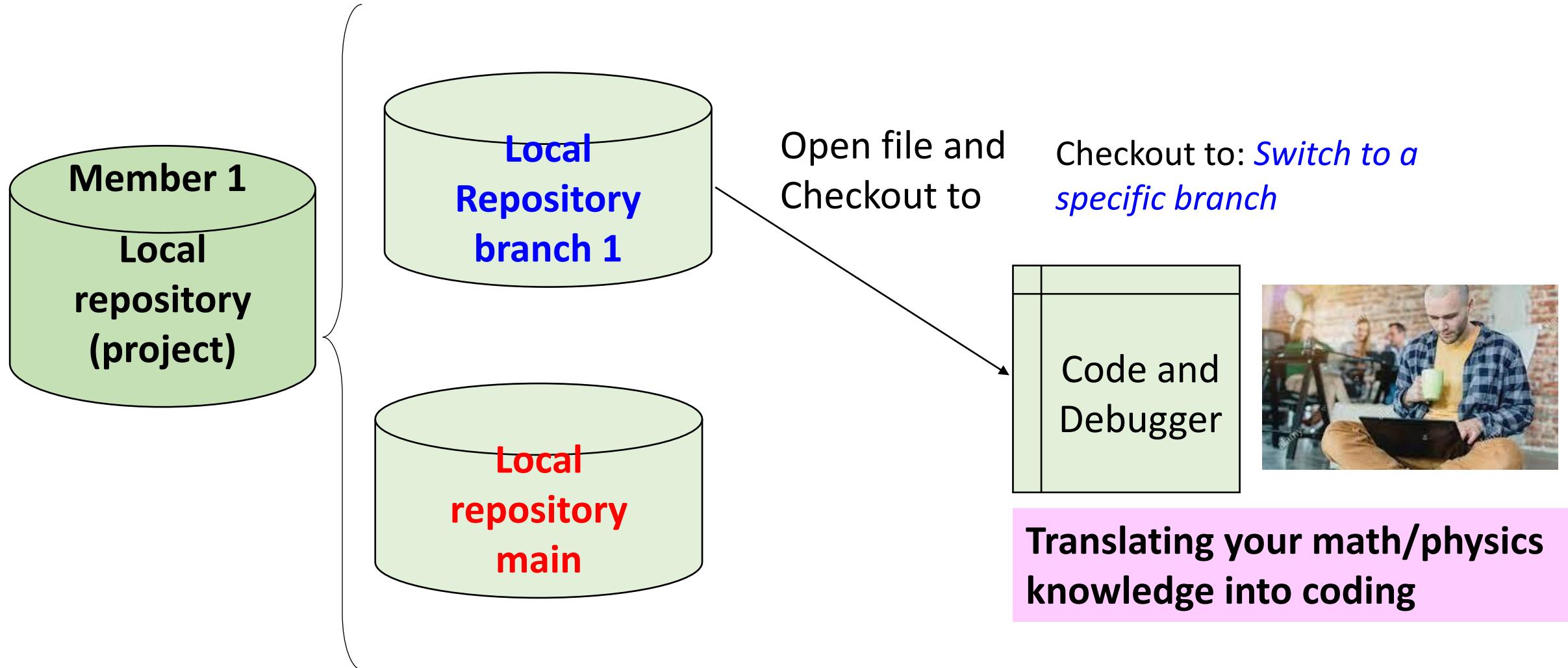


To generate a branch for different member that participant the project, creating a contained area of your repository to develop features, fix bugs and more.

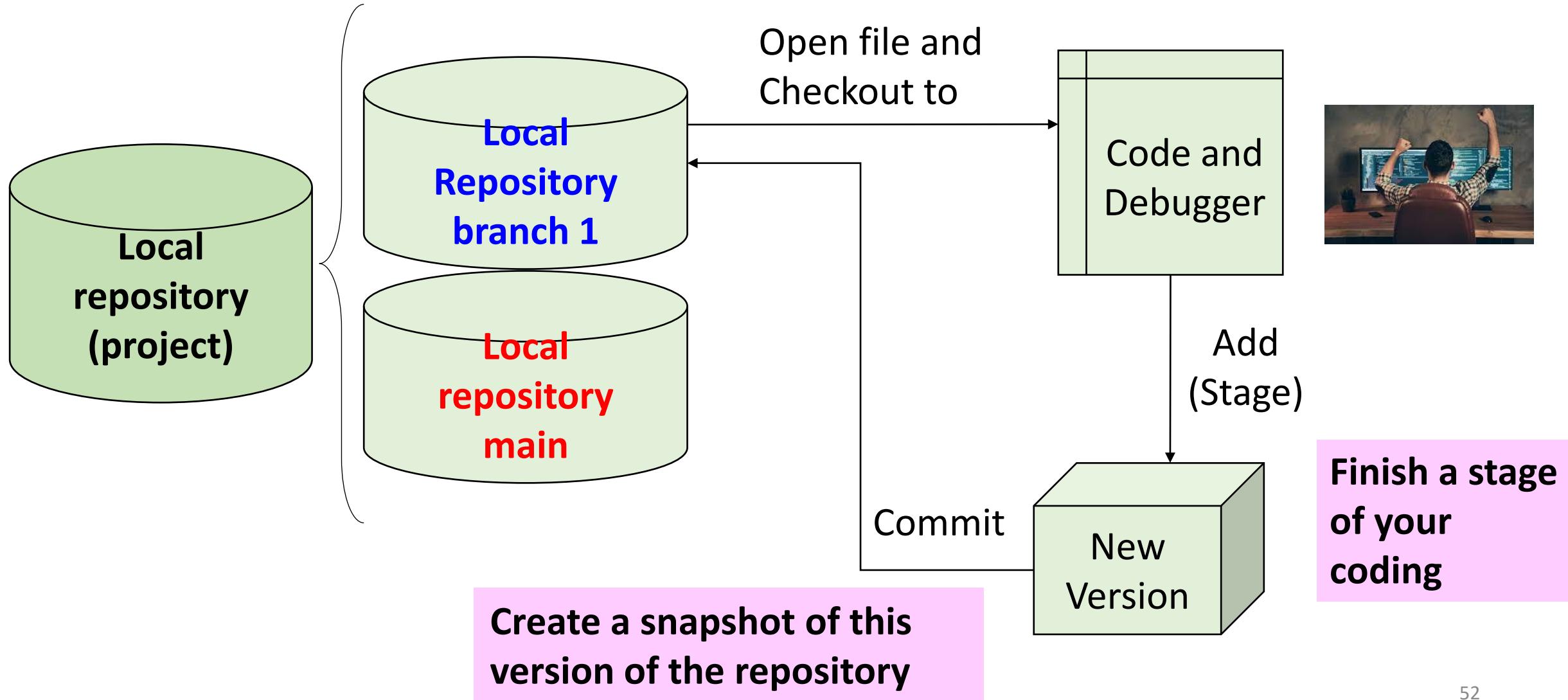
Collaborative coding online – Fetch/Pull Branch from Cloud



Collaborative coding online – Edit the code in the local computer



Collaborative coding online – Edit the code in the local computer

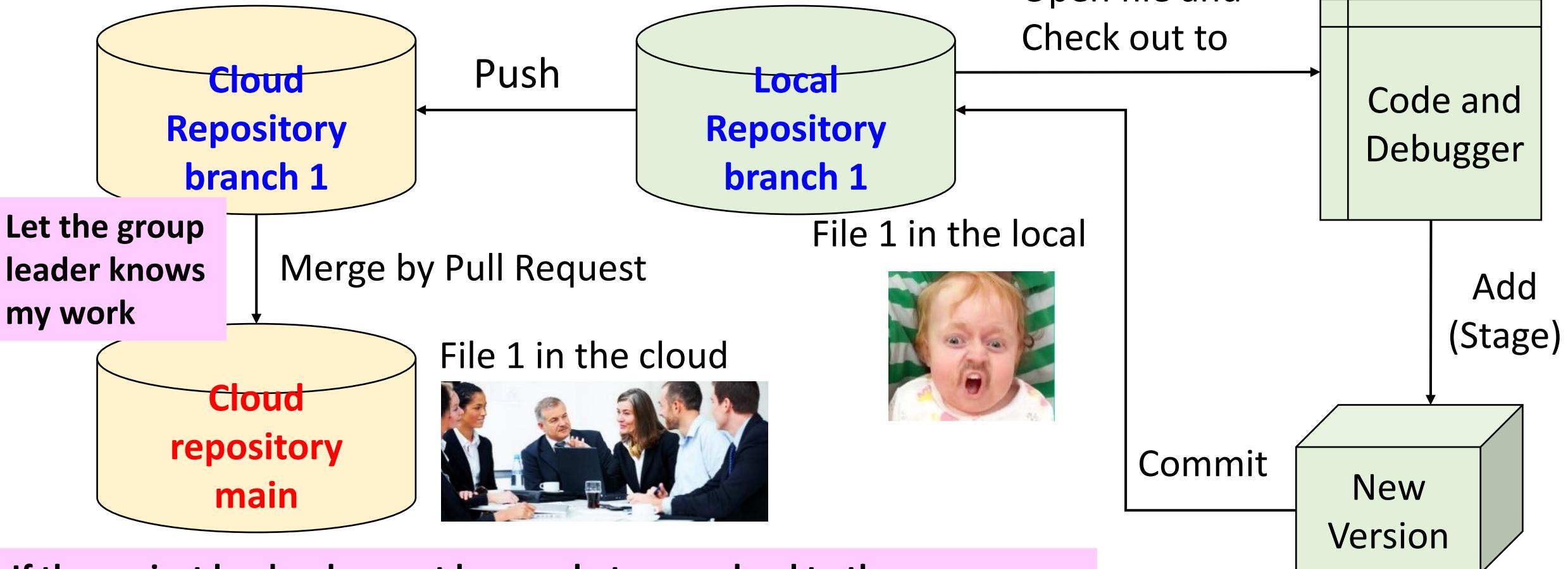


Concept of collaborative coding online

Web based

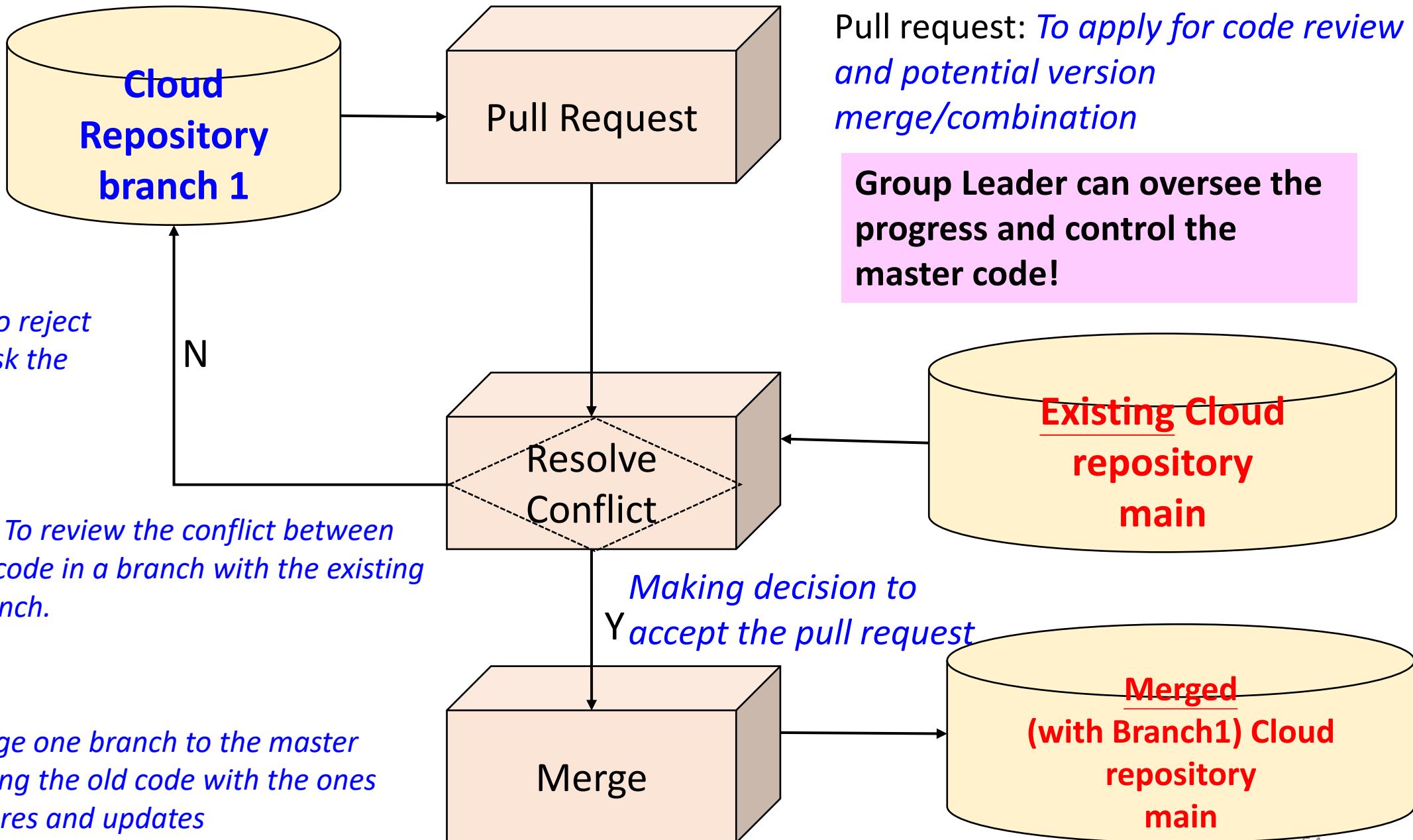
Video: [VSC-Git Branches](#)

Ready to update to the cloud branch!

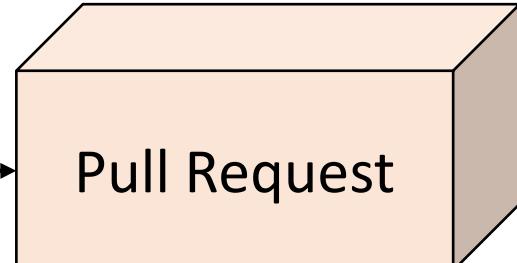
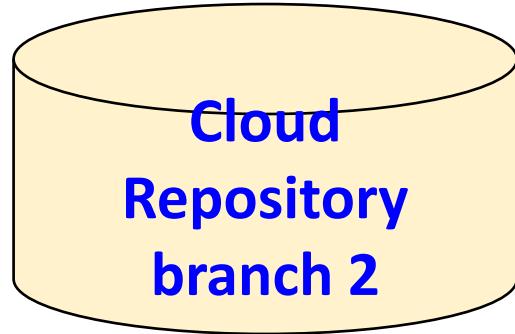


If the project leader does not know what you upload to the source code, you will create a lot of confusions!

Web based



Web based

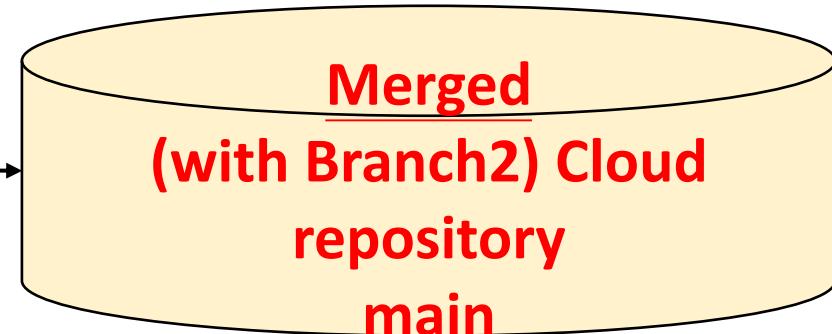
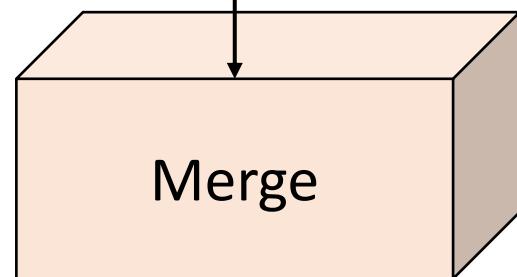
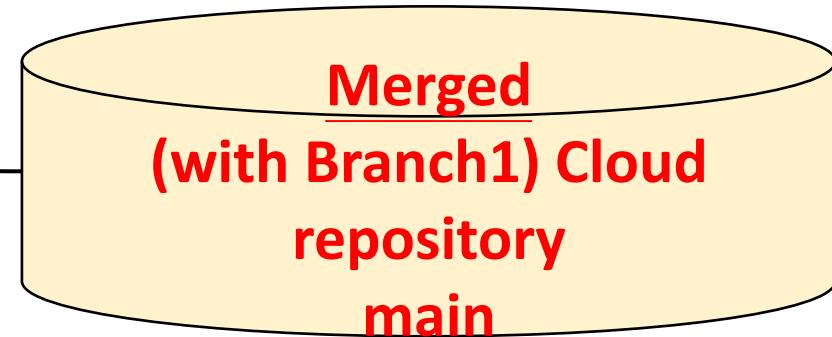
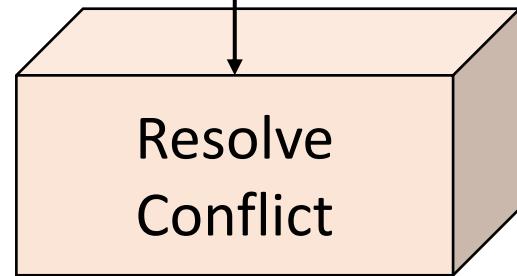


Group Leader can oversee the progress and control the master code!

Resolve Conflict: *To review the conflict between the new version code in a branch with the existing code in main branch.*

[More conflict!](#)

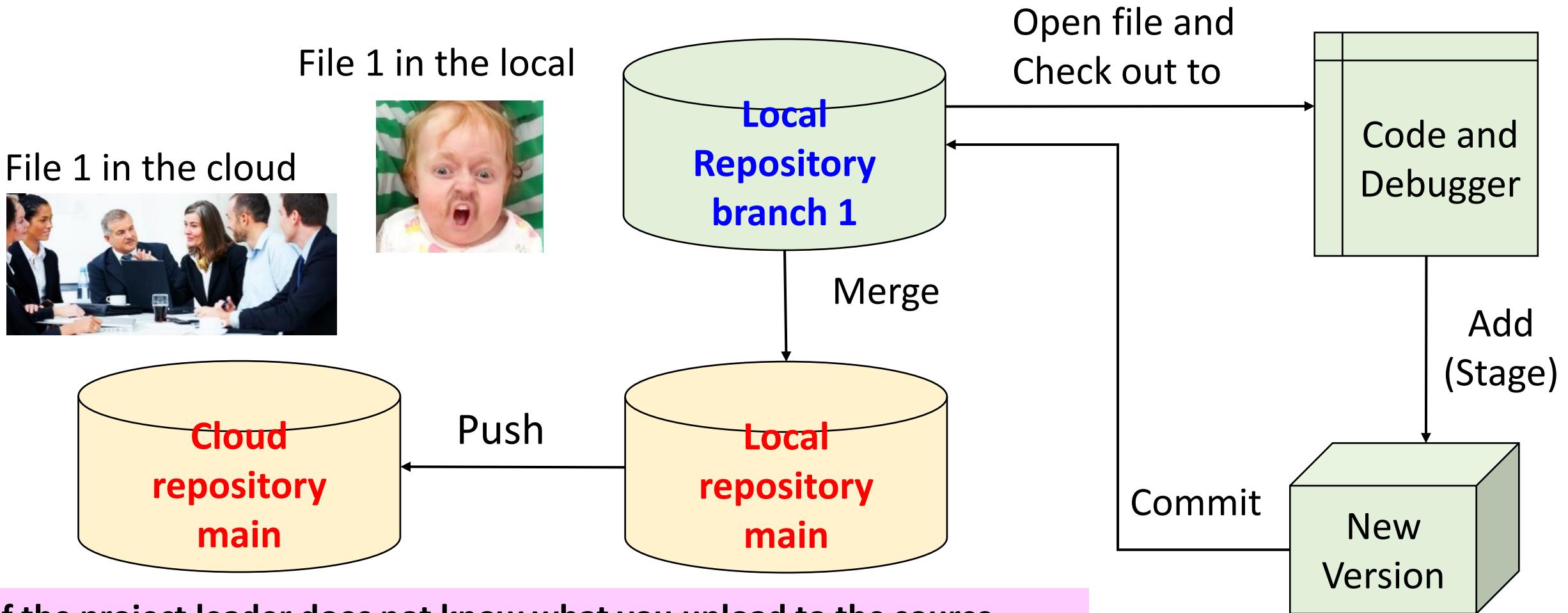
After reviewing, making decision to change or reject the pull request.



Concept of collaborative coding online

VS code based

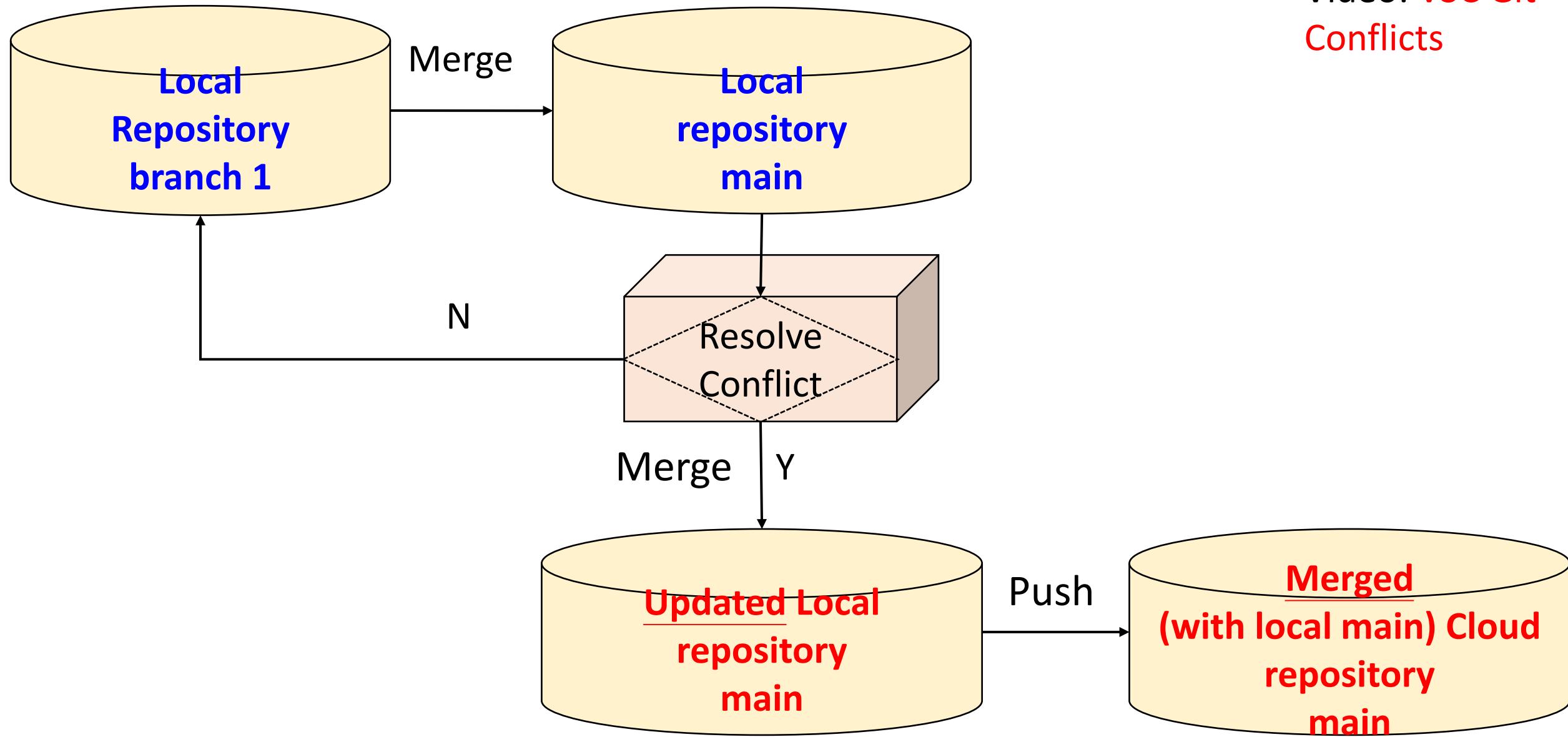
Video: [VSC Git Conflicts](#)



If the project leader does not know what you upload to the source code, you will create a lot of confusions!

VS code based

Video: [VSC Git Conflicts](#)



Software Installation and setup Guide

Install Python in Windows 10

Step 1: Download Python 3.6.4

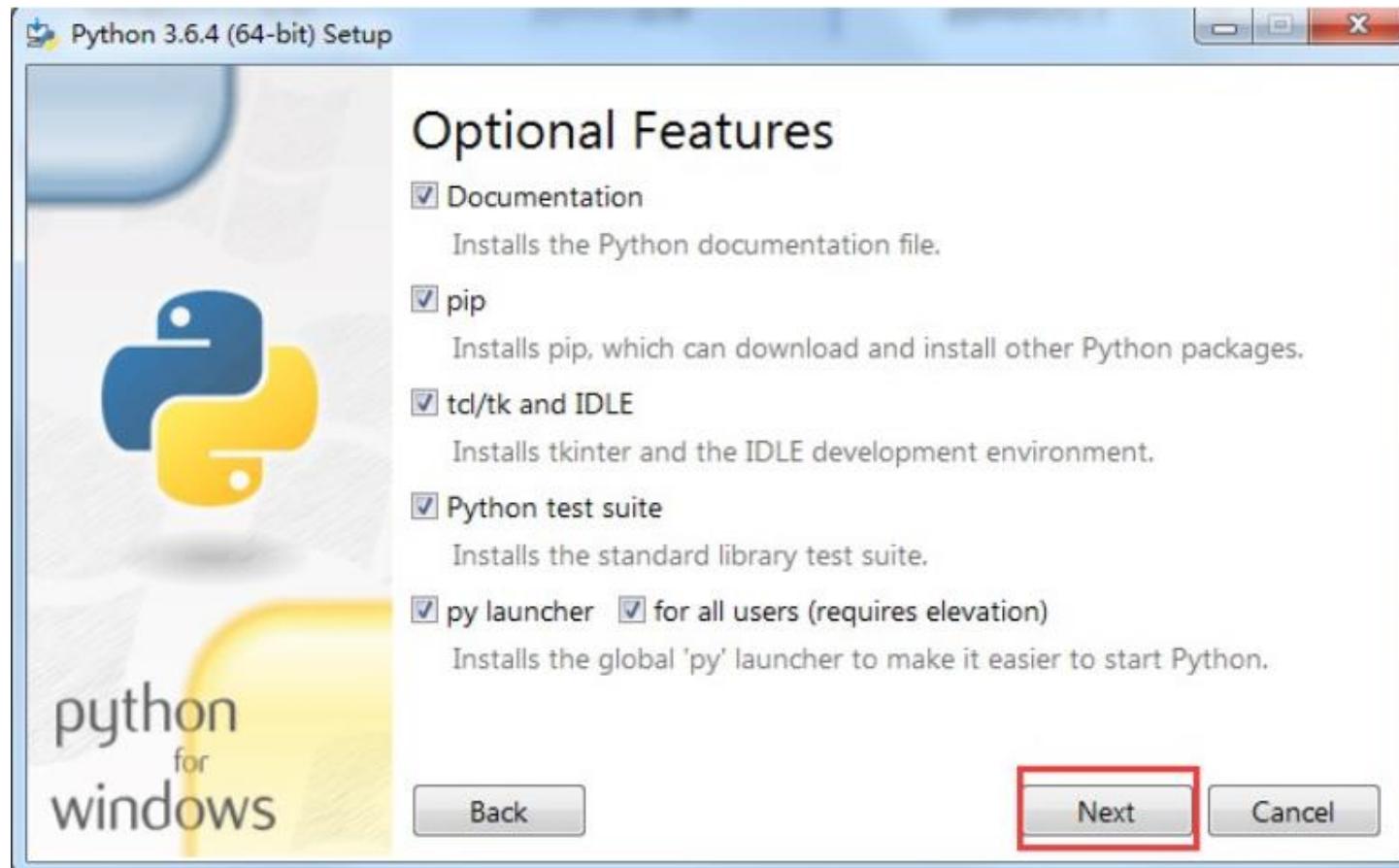
<https://www.python.org/ftp/python/3.6.4/python-3.6.4-amd64.exe>

Step 2: Install Python 3.6.4 in Windows 10



Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

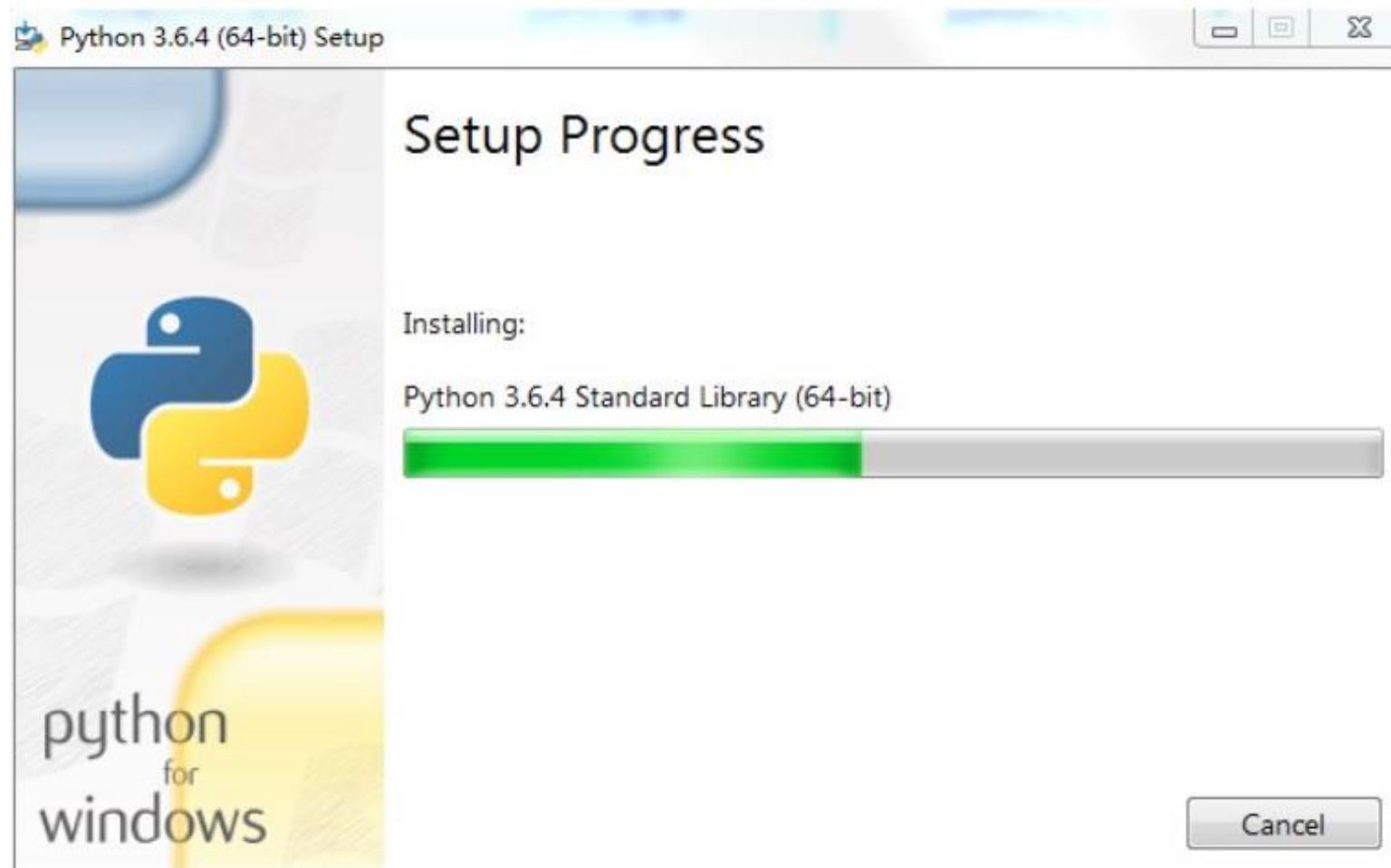
Install Python in Windows 10



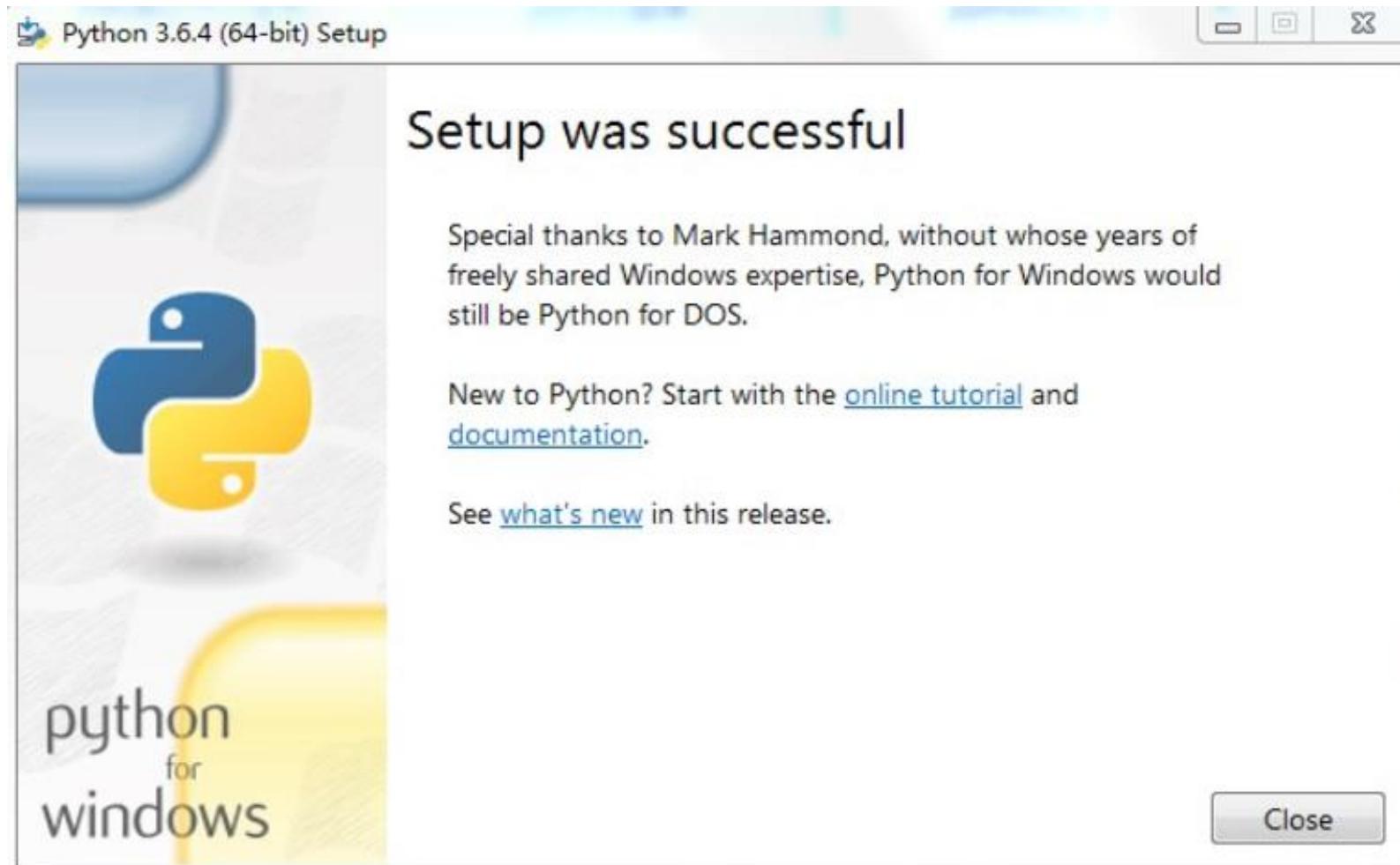
Install Python in Windows 10



Install Python in Windows 10



Install Python in Windows 10

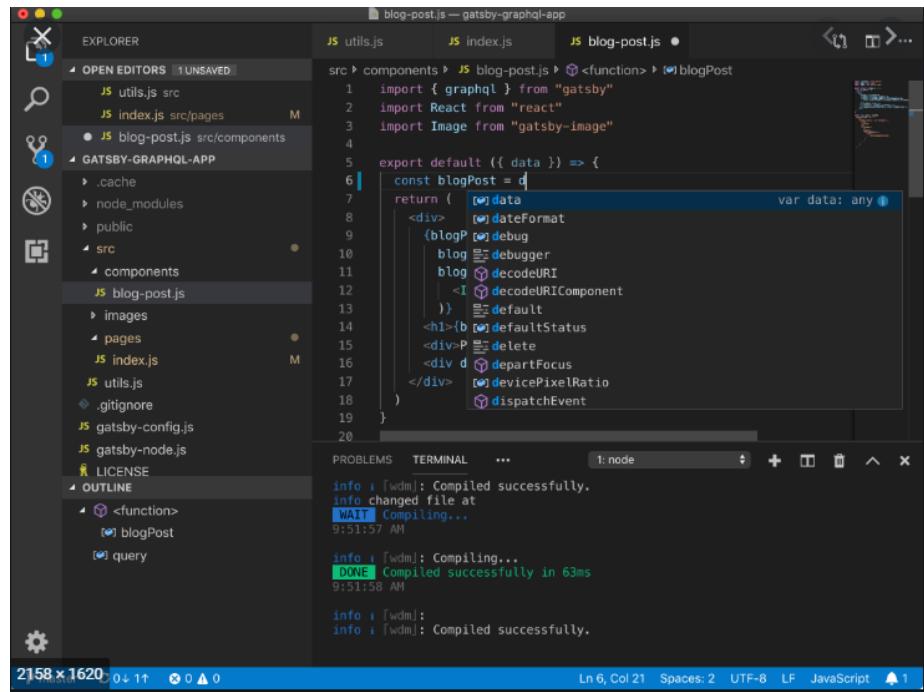


Install VS code in Windows 10

Step 1: Download the latest VS code

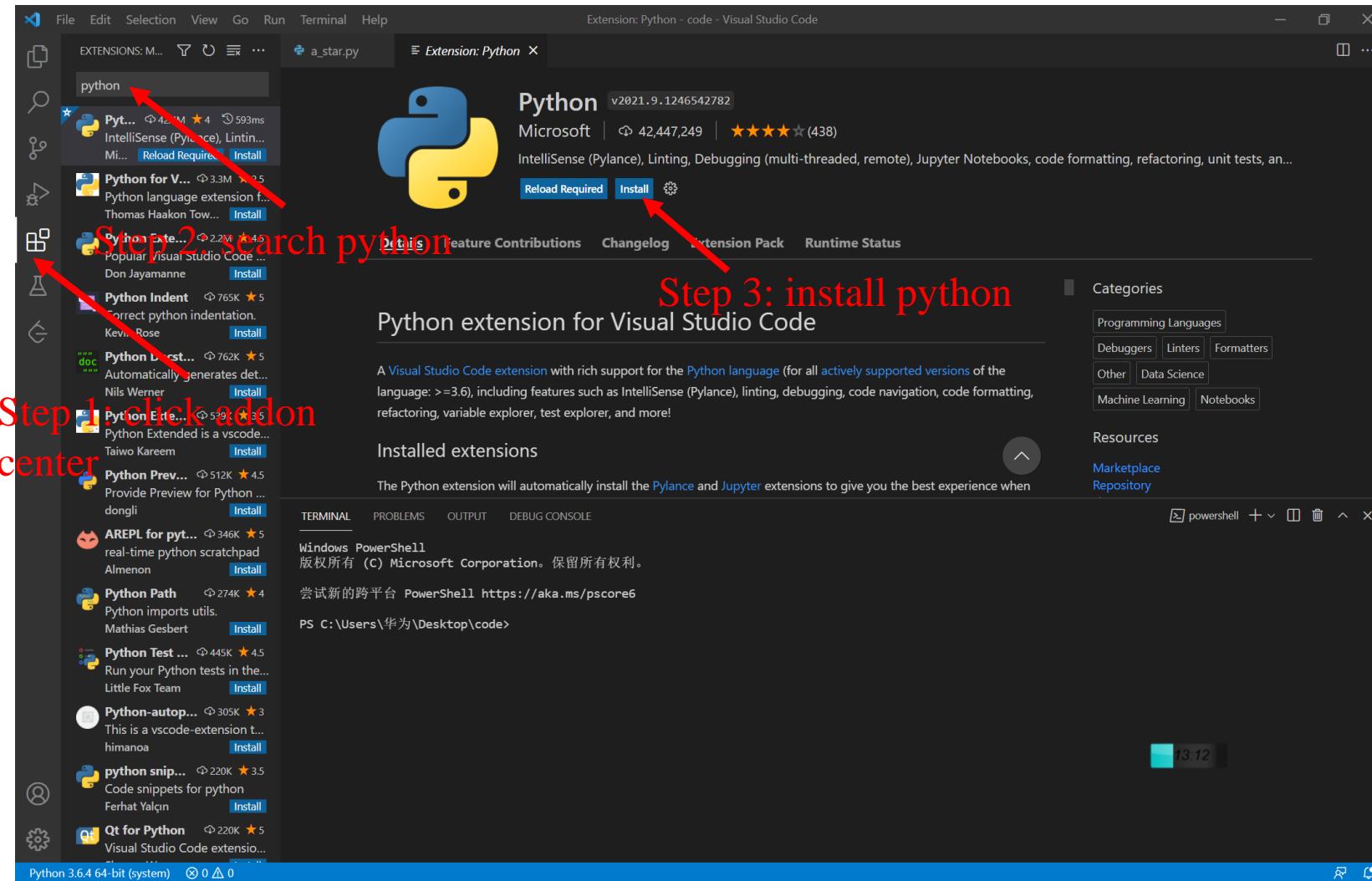
<https://code.visualstudio.com/download>

Step 2: Install latest VS code in Windows 10

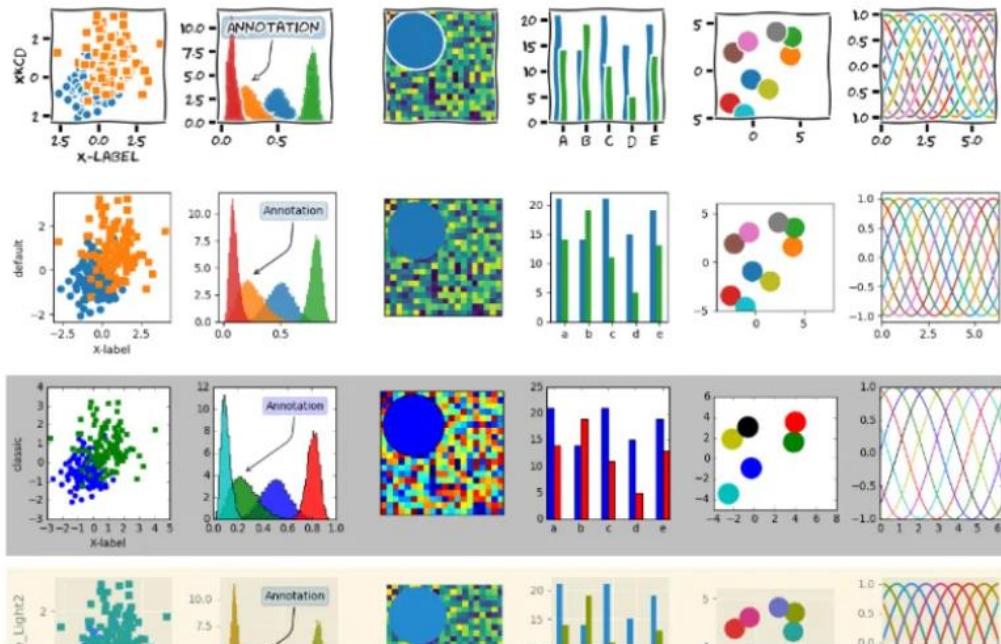


Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Install Python addon in VS code in Windows 10



Install matplotlib



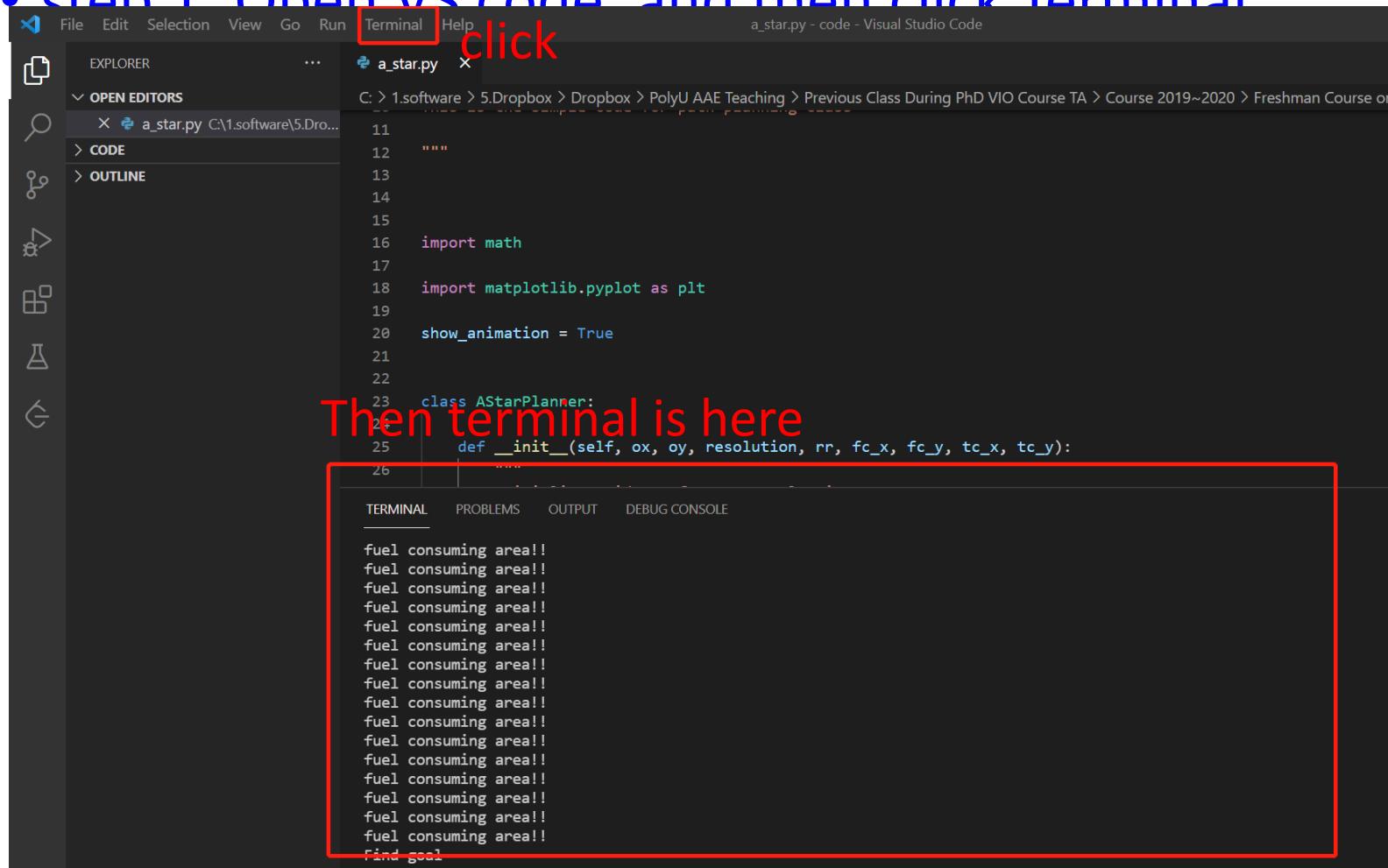
Source:

<https://www.gnuband.org/2017/12/29/gallery-of-xkcd-and-other-python-matplotlib-styles/>

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

Install matplotlib

- step 1: Open VS code, and then click Terminal



Install matplotlib

- step 2: Print following command into terminal>

The screenshot shows a Visual Studio Code interface. On the left, the Explorer sidebar shows a file named 'a_star.py'. The code editor displays the following Python script:

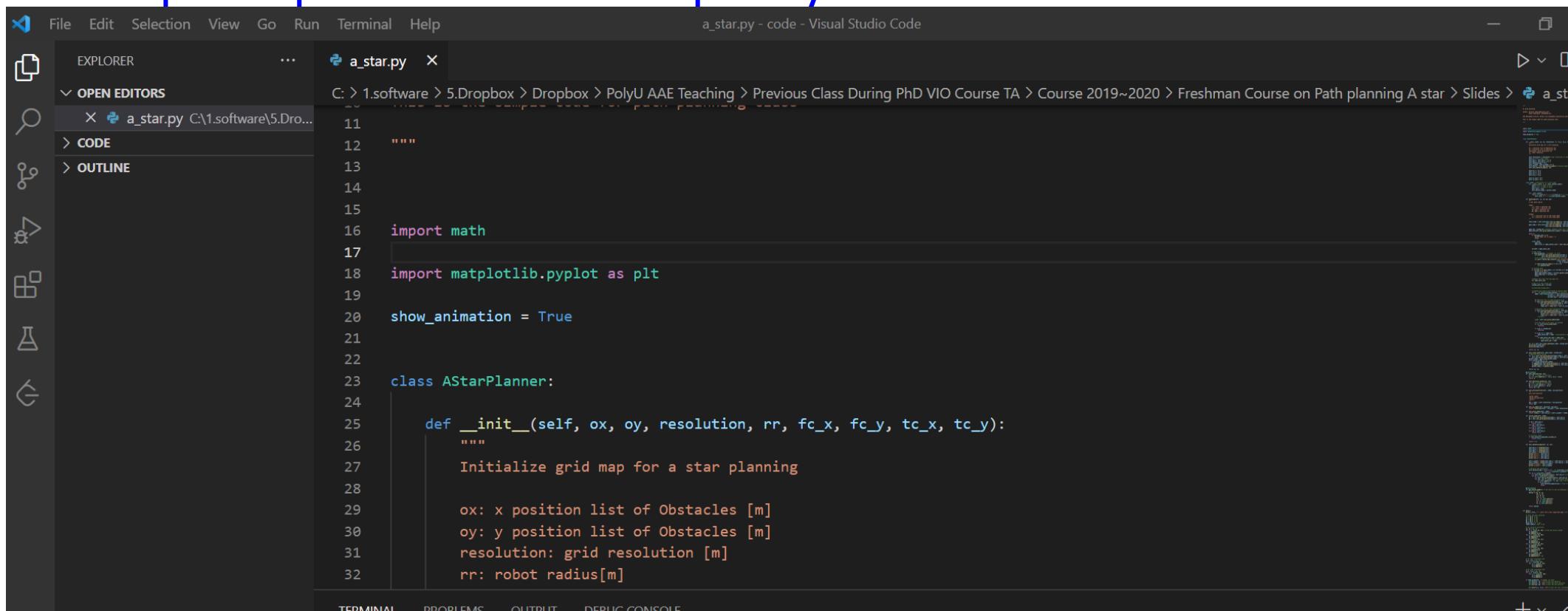
```
11
12 """
13
14
15
16 import math
17
18 import matplotlib.pyplot as plt
19
20 show_animation = True
21
22
23 class AStarPlanner:
24
25     def __init__(self, ox, oy, resolution, rr, fc_x, fc_y, tc_x, tc_y):
26         """
27             Initialize grid map for a star planning
28
29             ox: x position list of Obstacles [m]
30             oy: y position list of Obstacles [m]
31             resolution: grid resolution [m]
32             rr: robot radius[m]
```

Below the code editor is a terminal window showing a Windows PowerShell prompt. The text "pip install matplotlib==3.0.3" is displayed in red at the bottom of the terminal window.

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
尝试新的跨平台 PowerShell <https://aka.ms/pscore6>
PS C:\Users\华为\Desktop\code> pip install matplotlib==3.0.3

Test matplotlib

- step 3: Open the code sample by VS code



```
a_star.py - code - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER OPEN EDITORS CODE OUTLINE
C: > 1.software > 5.Dropbox > Dropbox > PolyU AAE Teaching > Previous Class During PhD VIO Course TA > Course 2019~2020 > Freshman Course on Path planning A star > Slides > a_star.py
11
12
13
14
15
16 import math
17
18 import matplotlib.pyplot as plt
19
20 show_animation = True
21
22
23 class AStarPlanner:
24
25     def __init__(self, ox, oy, resolution, rr, fc_x, fc_y, tc_x, tc_y):
26         """
27             Initialize grid map for a star planning
28
29             ox: x position list of Obstacles [m]
30             oy: y position list of Obstacles [m]
31             resolution: grid resolution [m]
32             rr: robot radius[m]
```

Test matplotlib

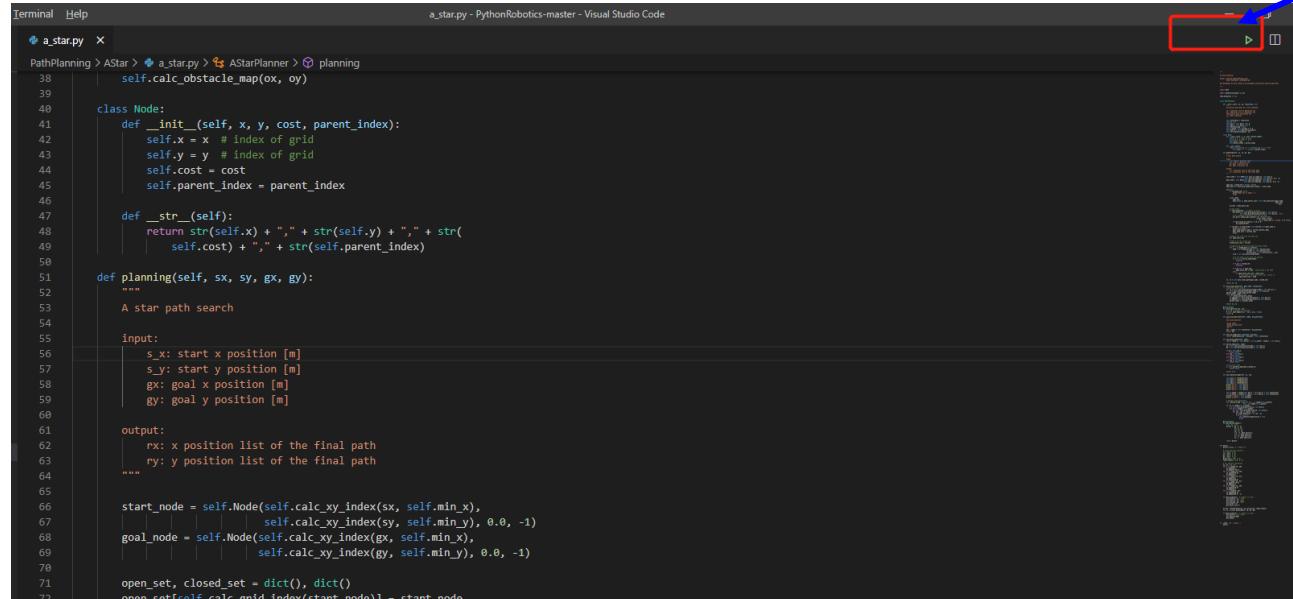
step 4: Run the demo

The screenshot shows a Visual Studio Code interface with the following elements:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows 'a_star.py' in the open editors list.
- Code Editor:** Displays the Python code for 'a_star.py'. The code imports math and matplotlib.pyplot, defines a class AStarPlanner, and includes a terminal output section.
- Terminal:** Shows repeated output 'fuel consuming area!!' followed by 'Find goal'.
- Figure Window:** Titled 'Figure 1', it displays a 2D grid map with axes ranging from -10 to 60. The map contains several obstacles: a red rectangular wall at approximately (15, 20) to (25, 30), a vertical green rectangle at (35, 15) to (45, 25), and two vertical black lines at x=0 and x=60. A red line traces a path starting from a green circle at (15, 10), moving right to (20, 10), then up to (20, 40), then right to (40, 40), then down to (40, 20), then right to (50, 20), and finally up to (50, 50).
- Output Panel:** Shows three entries: powershell, Python, and powershell.
- Bottom Bar:** Includes icons for home, back, forward, search, and refresh, along with a status bar showing '05.48'.

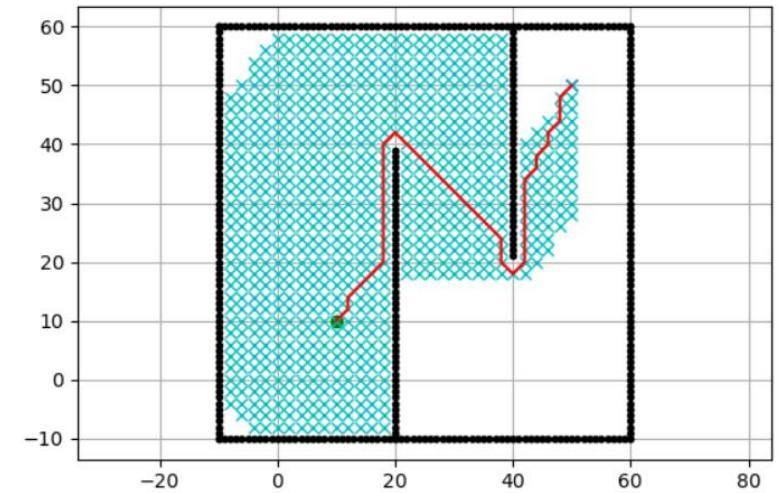
Run A Star in VS code

Run the code



```
a_star.py - PythonRobotics-master - Visual Studio Code
Terminal Help
a_star.py > AStar > a_star.py > AStarPlanner > planning
38     self.calc_obstacle_map(ox, oy)
39
40 class Node:
41     def __init__(self, x, y, cost, parent_index):
42         self.x = x # index of grid
43         self.y = y # index of grid
44         self.cost = cost
45         self.parent_index = parent_index
46
47     def __str__(self):
48         return str(self.x) + "," + str(self.y) + "," + str(
49             self.cost) + "," + str(self.parent_index)
50
51 def planning(self, sx, sy, gx, gy):
52     """
53     A star path search
54
55     input:
56         sx: start x position [m]
57         sy: start y position [m]
58         gx: goal x position [m]
59         gy: goal y position [m]
56
57     output:
58         rx: x position list of the final path
59         ry: y position list of the final path
56
57     """
58
59     start_node = self.Node(self.calc_xy_index(sx, self.min_x),
60                           self.calc_xy_index(sy, self.min_y), 0.0, -1)
61     goal_node = self.Node(self.calc_xy_index(gx, self.min_x),
62                           self.calc_xy_index(gy, self.min_y), 0.0, -1)
63
64     open_set, closed_set = dict(), dict()
65     open_set[self.calc_grid_index(start_node)] = start_node
66
67     while True:
68         cur_node = None
69         for node in open_set:
70             if cur_node == None or open_set[node].f < open_set[cur_node].f:
71                 cur_node = node
72
73         if cur_node == None:
74             print("A* failed to find a path")
75             break
76
77         if cur_node.x == gx and cur_node.y == gy:
78             print("A* found a path")
79             rx, ry = self.reconstruct_path(cur_node)
80             break
81
82         for i, j in self.get_neighboor_index(cur_node.x, cur_node.y):
83             if i < 0 or i > self.grid_size[0] - 1 or j < 0 or j > self.grid_size[1] - 1:
84                 continue
85             neighbor = self.Node(self.calc_xy_index(i, self.min_x),
86                                 self.calc_xy_index(j, self.min_y), 0.0, cur_node.index)
87             if neighbor in closed_set:
88                 continue
89             tentative_g_cost = cur_node.g + self.solid_time * self.grid_size[0] * abs(i - cur_node.x) + abs(j - cur_node.y)
90             if i == gx and j == gy:
91                 rx, ry = self.reconstruct_path(neighbor)
92                 break
93             if neighbor in open_set and tentative_g_cost >= open_set[neighbor].g:
94                 continue
95             open_set[neighbor.index] = neighbor
96
97         closed_set[cur_node.index] = cur_node
98
99     return rx, ry
100
101 def reconstruct_path(self, last_node):
102     rx, ry = [], []
103     while last_node.parent_index != -1:
104         rx.append(last_node.x)
105         ry.append(last_node.y)
106         last_node = open_set[last_node.parent_index]
107     rx.append(last_node.x)
108     ry.append(last_node.y)
109     return rx, ry
110
111 def get_neighboor_index(self, x, y):
112     neighbors = []
113     for i, j in [(0, 1), (0, -1), (1, 0), (-1, 0), (1, 1), (1, -1), (-1, 1), (-1, -1)]:
114         if i + x < 0 or i + x > self.grid_size[0] - 1 or j + y < 0 or j + y > self.grid_size[1] - 1:
115             continue
116         neighbors.append((i + x, j + y))
117     return neighbors
118
119 def calc_xy_index(self, x, min_x):
120     return (x - min_x) * self.grid_size[1] + min_y
121
122 def calc_grid_index(self, node):
123     return self.calc_xy_index(node.x, self.min_x) + self.grid_size[1] * self.grid_size[0] * node.y + self.grid_size[0] * node.y + node.x
```

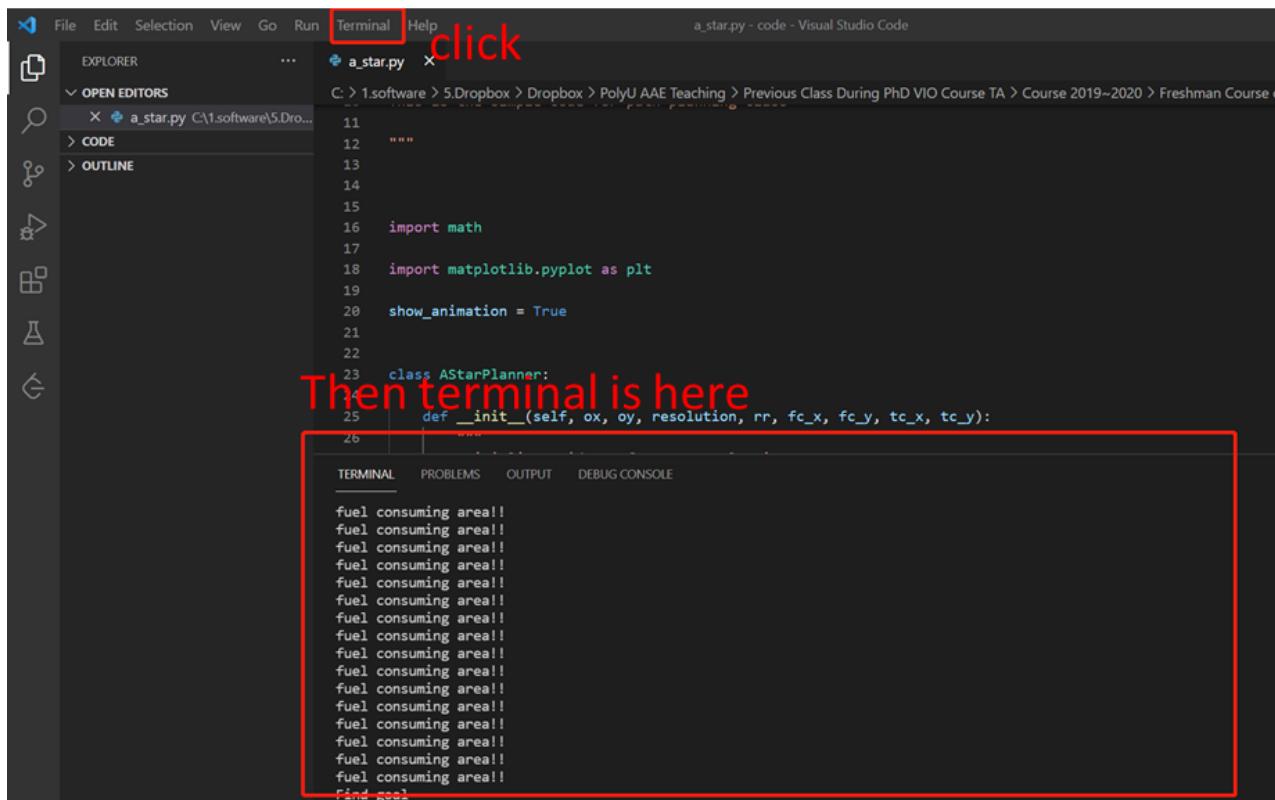
Figure 1



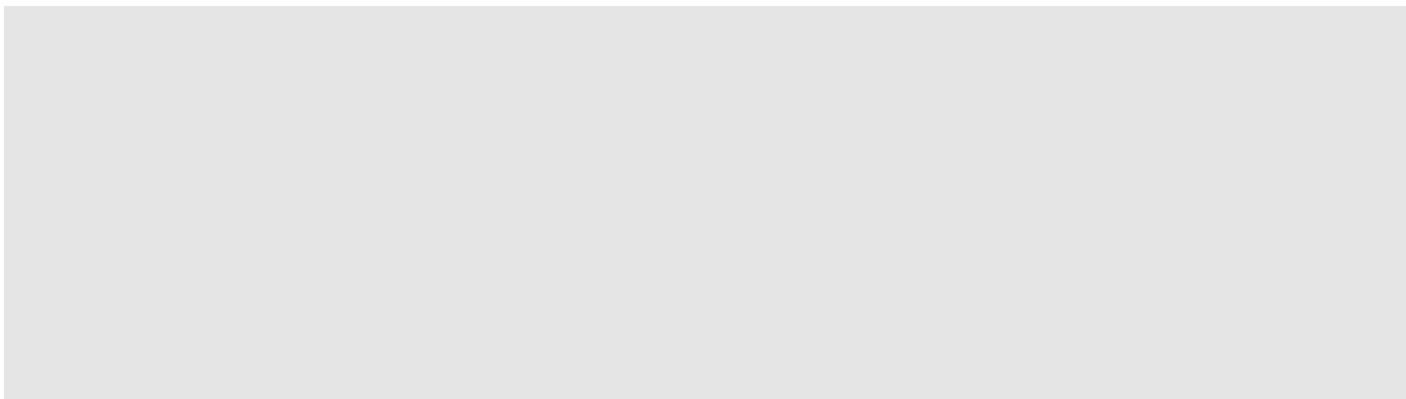
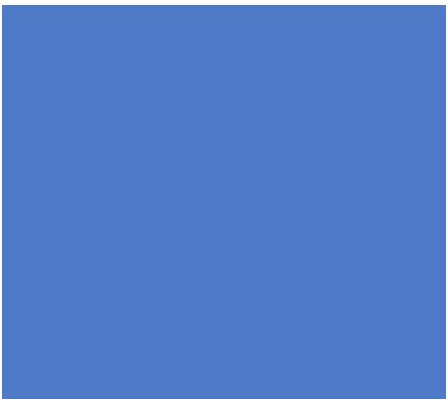
A* is a graph traversal and path search algorithm, which is often used in many fields of computer science due to its completeness, optimality, and optimal efficiency. One major practical drawback is its space complexity, as it stores all generated nodes in memory.

Troubleshoot: Connect Github page via command in VS Code

- Input the command below to the terminal (change the blue to your info)
 - *git config --global user.name weisongwen*
 - *git config --global user.email wenwsrobo@gmail.com*



Path Planning Code Guide



The Path Planning Code

- You can find the path planning code inside the course GitHub repository
- There are 2 set of codes:
 - A default one
 - A noted one
- The default one is a basic A* path planning code without any extra information and features
- The noted one provides an example of what your code should look like after modifications (**Remember each group should complete a different set of obstacles and requirements**)
- Repository link: [https://github.com/IPNL-POLYU/PolyU_AAE2004 Github Project](https://github.com/IPNL-POLYU/PolyU_AAE2004_Github_Project)

Where you can find the code

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with a dropdown for 'main' and a blue link 'PolyU_AAE2004_Github_Project / Sample Codes /'. A red oval highlights this link. To the right are buttons for 'Go to file', 'Add file', and three dots. Below the navigation is a status message: 'This branch is up to date with qmohsu:main.' On the right, there are 'Contribute' and 'Fetch upstream' buttons. The main area displays a list of commits:

Author	Commit Message	Date	Actions
qmohsu	Merge branch 'main' into LT2	de19c32 on Oct 15, 2021	History
..			
	Tutorial 1 Sample.py	Update Tutorial 1 Sample.py	5 months ago
	a_star_noted.py	Add files via upload	5 months ago
	a_star_original.py	Merge branch 'main' into LT2	4 months ago
	animation.gif	update sample code	5 months ago
	readme.md	Update readme.md	5 months ago

A red oval also highlights the 'a_star_noted.py' and 'a_star_original.py' entries in the commit list.

Noted Version Guide

- Line 50,51: Declaration of cost intensive area cost modifier
- Line 53: Declare cost per grid

```
34
35         self.resolution = resolution # get resolution of the grid
36         self.rr = rr # robot radius
37         self.min_x, self.min_y = 0, 0
38         self.max_x, self.max_y = 0, 0
39         self.obstacle_map = None
40         self.x_width, self.y_width = 0, 0
41         self.motion = self.get_motion_model() # motion model for grid search expansion
42         self.calc_obstacle_map(ox, oy)

43
44         self.fc_x = fc_x
45         self.fc_y = fc_y
46         self.tc_x = tc_x
47         self.tc_y = tc_y

48
49
50         self.Delta_C1 = 0.2 # cost intensive area 1 modifier
51         self.Delta_C2 = 0.4 # cost intensive area 2 modifier
52
53         self.costPerGrid = 1
54
```

Noted Version Guide

- Line 115: Showing the final calculation of total trip time
- Line 135-144: Adding additional cost during cost intensive area

```
103    if show_animation: # pragma: no cover
104        plt.plot(self.calc_grid_position(current.x, self.min_x),
105                  self.calc_grid_position(current.y, self.min_y), "xc")
106        # for stopping simulation with the esc key.
107        plt.gcf().canvas.mpl_connect('key_release_event',
108                                      lambda event: [exit(
109                                          0) if event.key == 'escape' else None])
110        if len(closed_set.keys()) % 10 == 0:
111            plt.pause(0.001)
112
113        # reaching goal
114        if current.x == goal_node.x and current.y == goal_node.y:
115            print("Total Trip time required -> ",current.cost )
116            goal_node.parent_index = current.parent_index
117            goal_node.cost = current.cost
118            break
119
120        # Remove the item from the open set
121        del open_set[c_id]
122
123        # Add it to the closed set
124        closed_set[c_id] = current
125
126        # print(len(closed_set))
127
128        # expand_grid search grid based on motion model
129        for i, _ in enumerate(self.motion): # tranverse the motion matrix
130            node = self.Node(current.x + self.motion[i][0],
131                             current.y + self.motion[i][1],
132                             current.cost + self.motion[i][2] * self.costPerGrid, c_id)
133
134            ## add more cost in cost intensive area 1
135            if self.calc_grid_position(node.x, self.min_x) in self.tc_x:
136                if self.calc_grid_position(node.y, self.min_y) in self.tc_y:
137                    # print("cost intensive area!!")
138                    node.cost = node.cost + self.Delta_C1 * self.motion[i][2]
139
140            # add more cost in cost intensive area 2
141            if self.calc_grid_position(node.x, self.min_x) in self.fc_x:
142                if self.calc_grid_position(node.y, self.min_y) in self.fc_y:
143                    # print("cost intensive area!!")
144                    node.cost = node.cost + self.Delta_C2 * self.motion[i][2]
145
146        # print()
```

Noted Version Guide

- Line 263-270: Declaring motions for the aircraft
- Line 279-284: Declaring starting point and end point

```
260     @staticmethod
261     def get_motion_model(): # the cost of the surrounding 8 points
262         # dx, dy, cost
263         motion = [[1, 0, 1],
264                    [0, 1, 1],
265                    [-1, 0, 1],
266                    [0, -1, 1],
267                    [-1, -1, math.sqrt(2)],
268                    [-1, 1, math.sqrt(2)],
269                    [1, -1, math.sqrt(2)],
270                    [1, 1, math.sqrt(2)]]
271
272         return motion
273
274
275     def main():
276         print(__file__ + " start the A star algorithm demo !!") # print simple notes
277
278         # start and goal position
279         sx = 0.0 # [m]
280         sy = 0.0 # [m]
281         gx = 50.0 # [m]
282         gy = 0.0 # [m]
283         grid_size = 1 # [m]
284         robot_radius = 1.0 # [m]
```

Noted Version Guide

- Line 309-329: Adding obstacles
- Line 337-348, Adding cost intensive areas (**Hint: Refer to this part for your task 2!**)

```
308 # set obstacle positions for group 9
309 ox, oy = [], []
310 ▼
311 for i in range(-10, 60): # draw the button border
312     ox.append(i)
313     oy.append(-10.0)
314 for i in range(-10, 60): # draw the right border
315     ox.append(60.0)
316     oy.append(i)
317 for i in range(-10, 60): # draw the top border
318     ox.append(i)
319     oy.append(60.0)
320 for i in range(-10, 60): # draw the left border
321     ox.append(-10.0)
322     oy.append(i)
323 for i in range(-10, 30): # draw the free border
324     ox.append(20.0)
325     oy.append(i)
326
327 for i in range(0, 20):
328     ox.append(i)
329     oy.append(-1 * i + 10)
330
331 # for i in range(40, 45): # draw the button border
332 #     ox.append(i)
333 #     oy.append(30.0)
334
335
336 # set cost intensive area 1
337 fc_x, fc_y = [], []
338 for i in range(30, 40):
339     for j in range(0, 40):
340         fc_x.append(i)
341         fc_y.append(j)
342
343 # set cost intensive area 1
344 tc_x, tc_y = [], []
345 for i in range(10, 20):
346     for j in range(20, 50):
347         tc_x.append(i)
348         tc_y.append(j)
```

Noted Version Guide

- If you wish to do the calculation using the program, you should add the calculation function under line 117, inside the reaching goal condition
- It would be even better if the program could distinguish viable and non-viable aircraft types!
- Use the noted version as your sample to modify your own code!

Program Calculation for Task 1

- When you add in a cost calculation function, the output should look something like this, it should be able to:
 1. Calculate each aircraft types' operating costs
 2. Mention which type might not be viable for certain scenarios

```
min_x: -10
min_y: -10
max_x: 60
max_y: 60
x_width: 70
y_width: 70
Total travelling time -> 93.35575746753788
A321 not viable!
Total cost of operating A330 in this scenario: 27360.167918740684
Total cost of operating A350 in this scenario: 30752.648960130347
```