

# ENAE 380 Flight Software Systems

## Lecture 1: Introduction

# About This Course

Avionics using advanced sensor and computing technologies are at the heart of every modern aerospace vehicle. Advanced software systems improve cockpit safety and enable unmanned and deep-space missions. Object-oriented programming and software engineering concepts required to design and build complex flight software systems will be discussed. Software validation, verification, real-time performance analysis to assess flight software system reliability and robustness; human-machine interfaces designed for piloted systems; automatic onboard data acquisition and decision-making for unmanned air and space vehicles.

# Course Logistics

- M/W 12:00 - 12:50 pm, EGR 1202
- Labs: BRING YOUR OWN LAPTOP\*
  - If you have issues with your computer, installation, etc., please let us know ASAP
- Contact:
  - [mumu@umd.edu](mailto:mumu@umd.edu)
  - Office Hours:
    - By appointment

# Course Logistics

- TA: Adam Lahr
- TF
  - Ben Leazar
  - Ohm Sapa
  - Tom Bigot
- Contact them via elms or piazza
- Office hours and locations posted on elms

# Course Logistics

- All assignments submitted electronically on Elms
  - Check that you submit the correct version of your homework! We will not grade re-submissions.
- Questions/Discussion on Piazza
  - <https://piazza.com/umd/fall2024/enae380/home>

# Course Logistics

- Grading
  - Labs 30%
  - Midterm 25%
  - Project Proposal 5%
  - Final Project (code, report, video walkthrough)  
40%



cameo

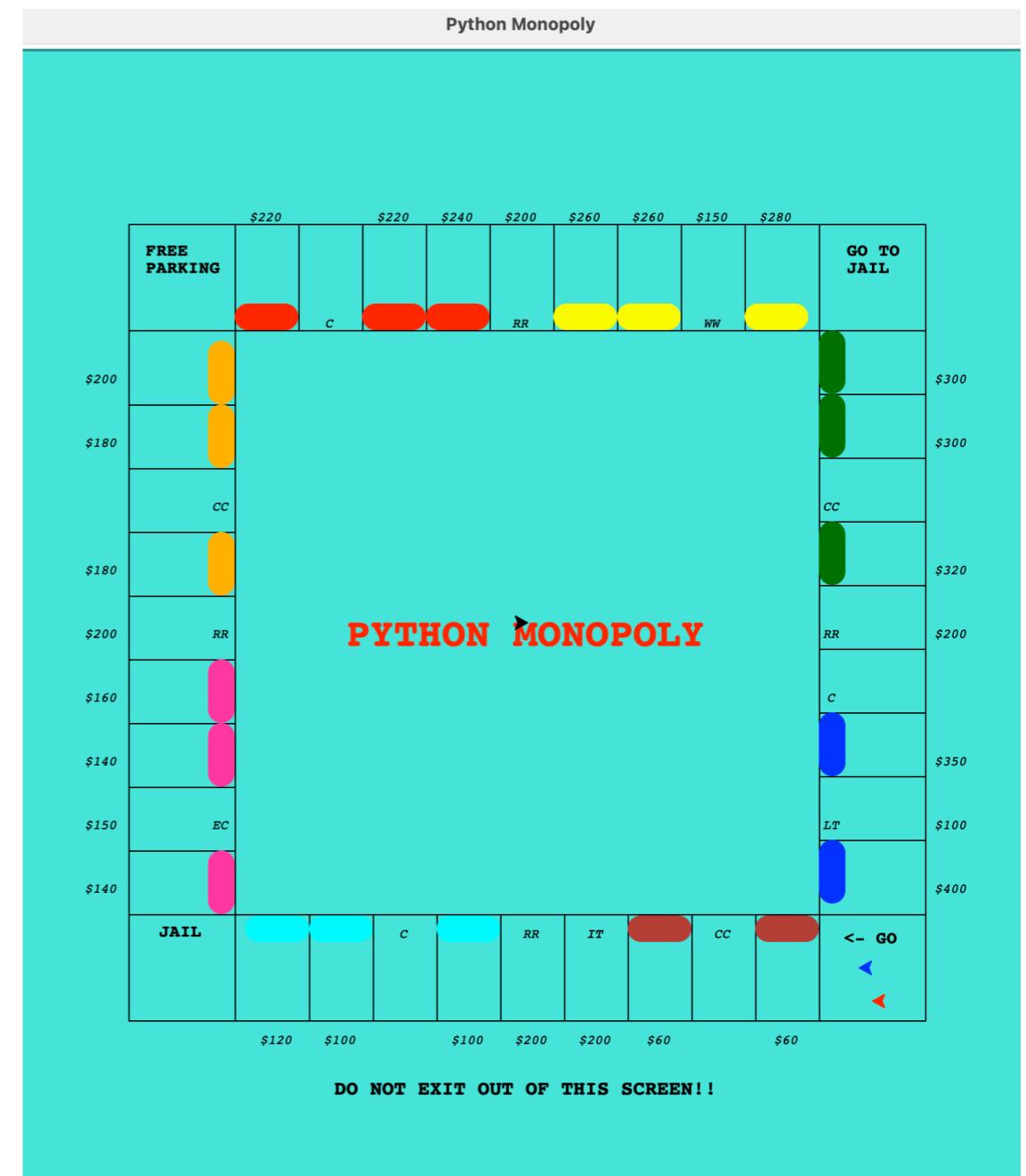
# Syllabus

# Final Project

**\*SAVE YOUR WORK**

# Final Project

- Movie scraper
- Angry Birds trajectory simulator
- Arduino Alarm clock
- Music generation using ML
- Ice Skating turning sensor
- Escape room



# Academic Integrity

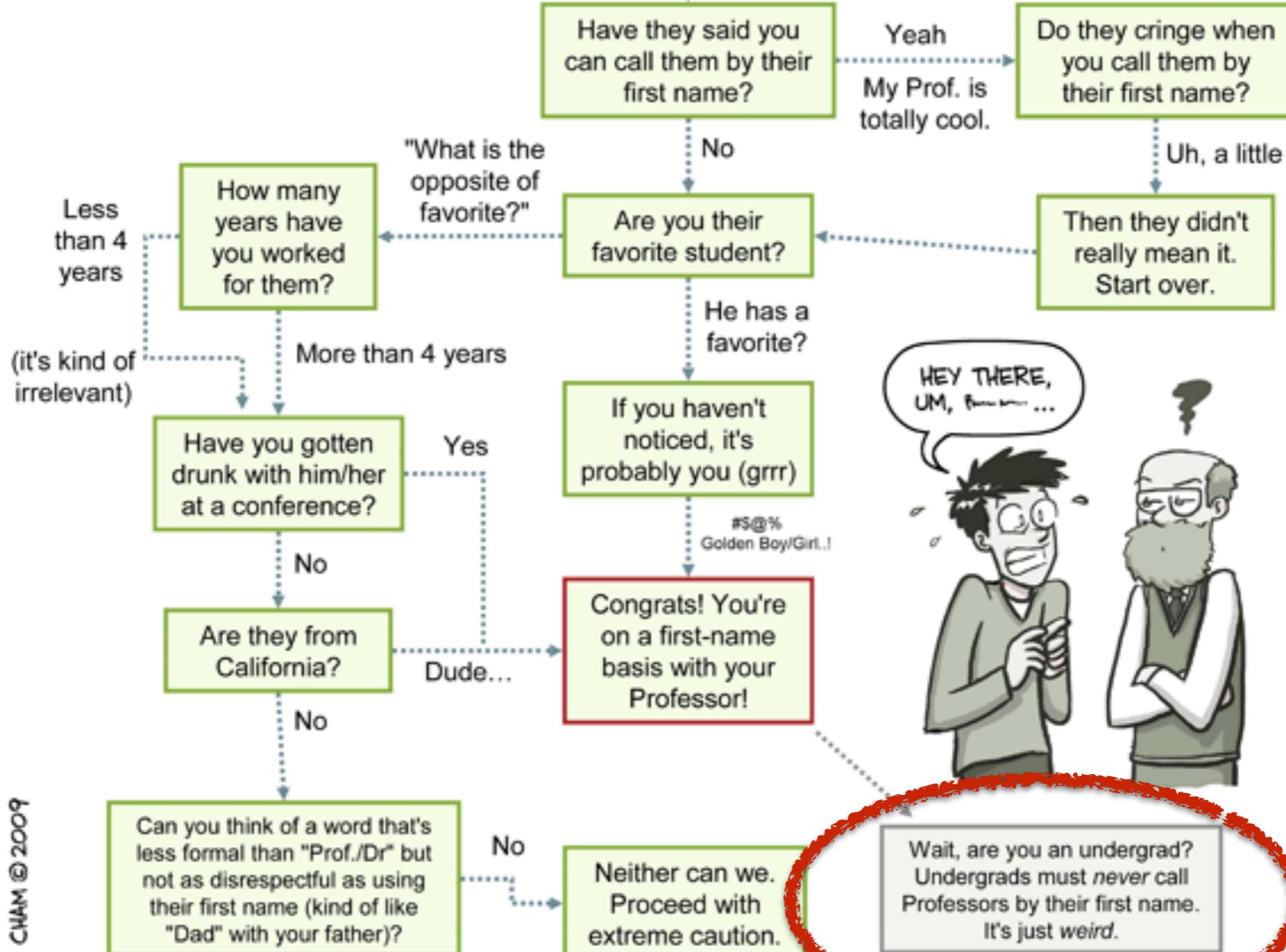
<http://www.ugst.umd.edu/courserelatedpolicies.html>

Academic integrity is an important value for our community. Because of this, we have high standards for behavior. Academic dishonesty is prohibited. All students are subject to the requirements of the Code of Academic Integrity and are responsible for upholding these standards for this course. It is very important for all students to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. If you have any questions about whether something is unauthorized, please contact the instructor.

Be reasonable. We encourage you to interact with classmates to better understand the course material. For homework, you must submit your own work. Collaboration on work turned in is not permitted. You may ask for help and give help to others so long as that help does not reduce to doing the work for the other person. If you are asking for help from someone, you may show your code to others, but you may not view theirs. Incorporating parts of code that you find online or elsewhere is ok, as long as you cite the origins, and those snippets are not the solution to the assigned problem.

Some miscellaneous things

## What to call your Professor



# **How to write an email**

Address formally

Specify class and assignment (if applicable)

Keep email brief and to the point

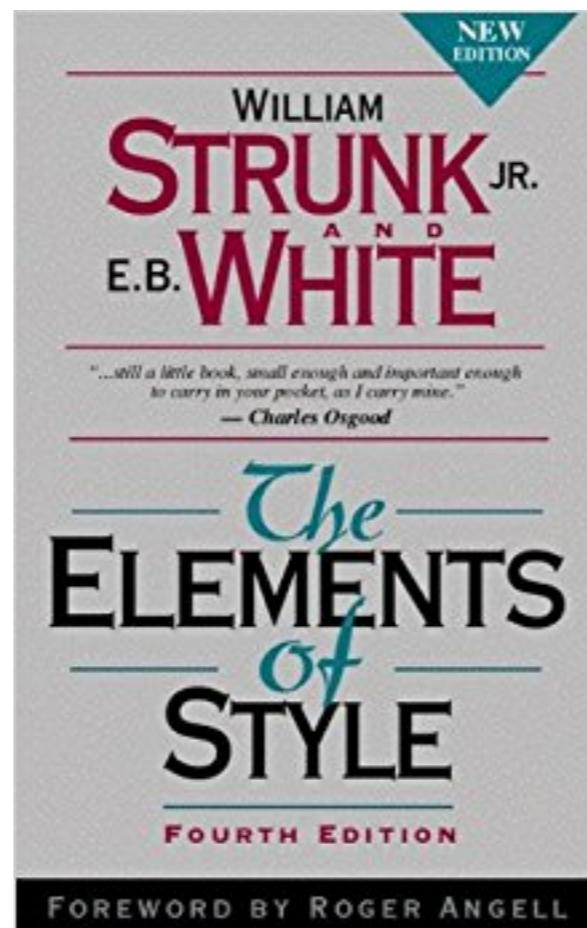
Use full sentences and proper grammar

Send email and be patient when waiting for a reply

# Grammar Matters

Elements of Style - Strunk and White

<http://www.jlakes.org/ch/web/The-elements-of-style.pdf>

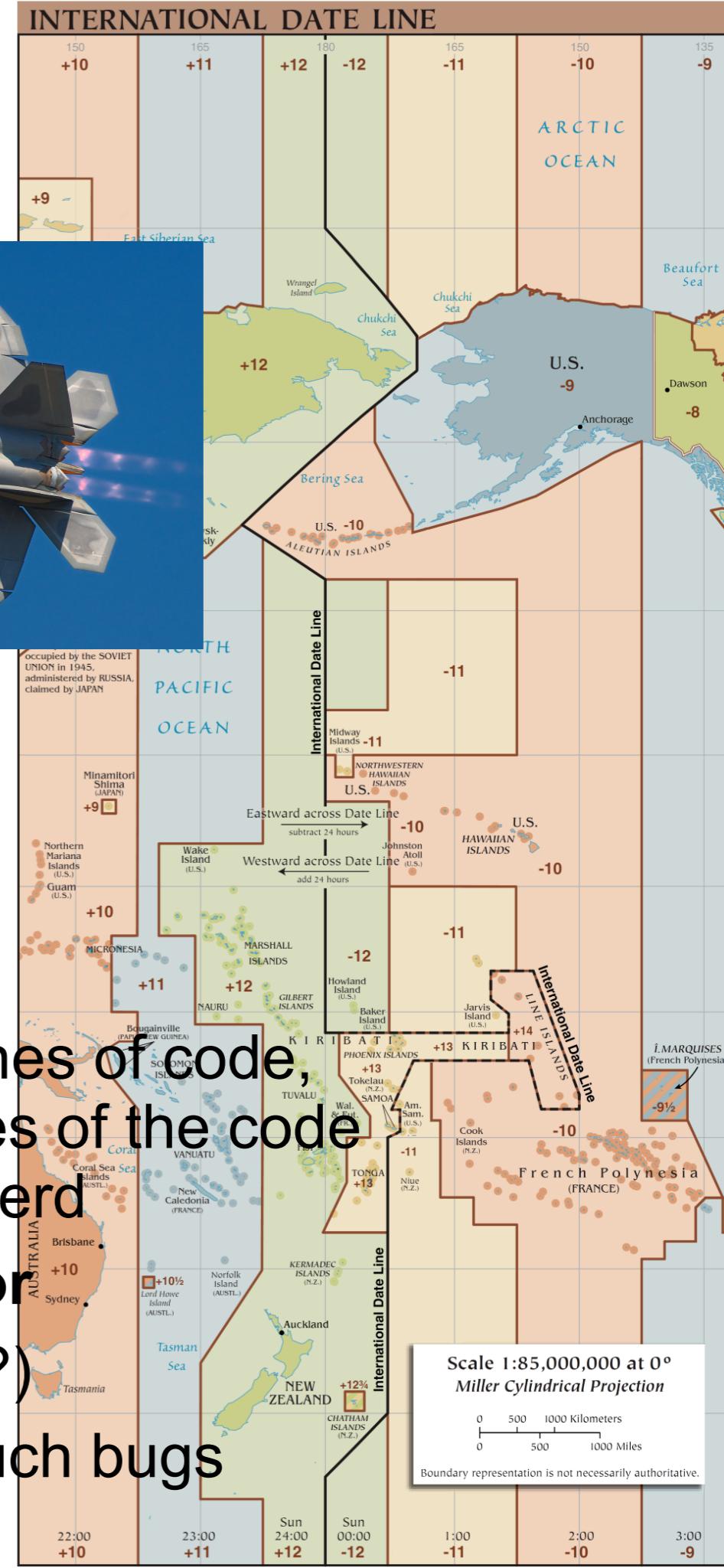


# Software Goes Awry

- F22 Raptor International Date Line
- World of Warcraft
- JPL Mars Pathfinder
- Cybersecurity
- Pokemon Go

# F-22 Raptor IDL

- **February 2007:** 6 F-22 Raptors flight from Hickam AFB, Hawaii to Okinawa, Japan
  - \$360 million per aircraft
  - Computer crash crossing IDL
    - No navigation, comms, fuel mgmt
    - Escorted back by tankers
    - Could have lost all six if poor visibility
- **Cause:**
  - “It was a computer glitch in the millions of lines of code, somebody made an error in a couple of lines of the code and everything goes.” AF Maj G Don Shepperd
- **Related: F-16 inverts when crossing equator**
  - Found in simulation (perhaps urban legend?)
  - But still should put designers on notice of such bugs



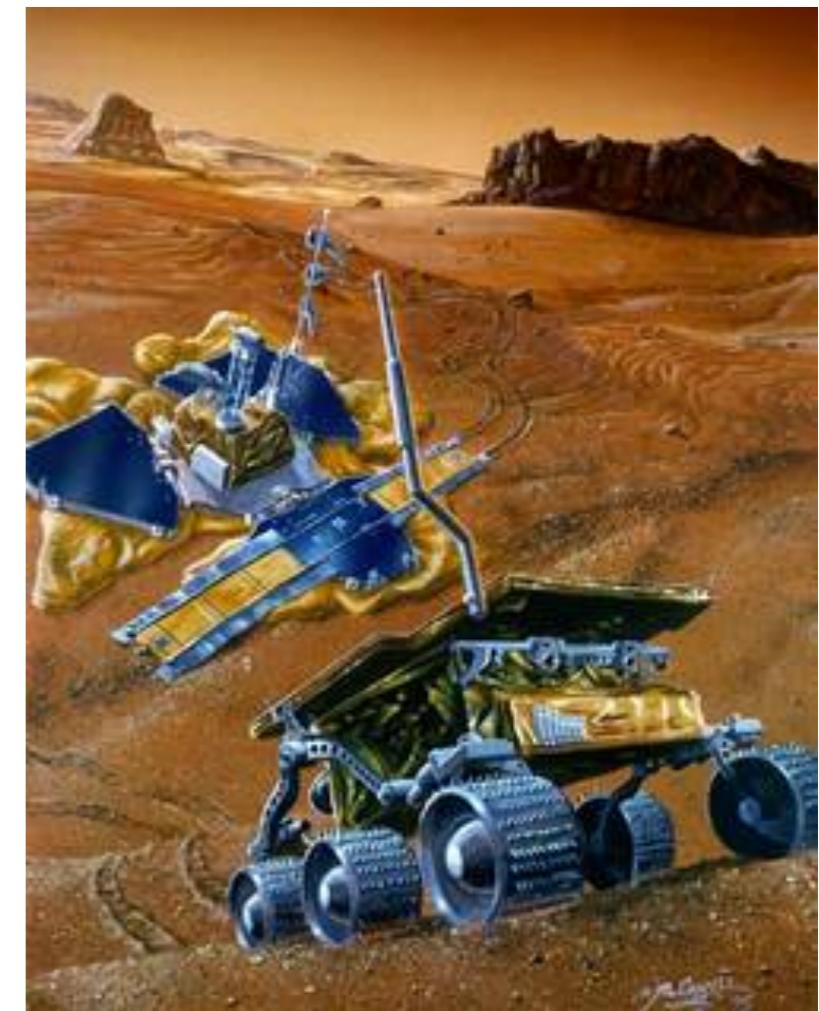
# WoW Blood Corruption

- September 2005
- <https://keithmaggio.wordpress.com/2010/06/03/game-glitch-analysis-corrupted-blood-incident/>
- [https://en.wikipedia.org/wiki/Corrupted\\_Blood\\_incident](https://en.wikipedia.org/wiki/Corrupted_Blood_incident)
- Modeling Infectious Diseases Dissemination Through Online Role-Playing Games
  - [ftp://ftp.inf.puc-rio.br/pub/docs/Estudos\\_de\\_Metrias/Internet\\_Estudos/Aula%208%20Leitura-Modeling%20Infectious%20Diseases%20Dissemination.pdf](ftp://ftp.inf.puc-rio.br/pub/docs/Estudos_de_Metrias/Internet_Estudos/Aula%208%20Leitura-Modeling%20Infectious%20Diseases%20Dissemination.pdf)



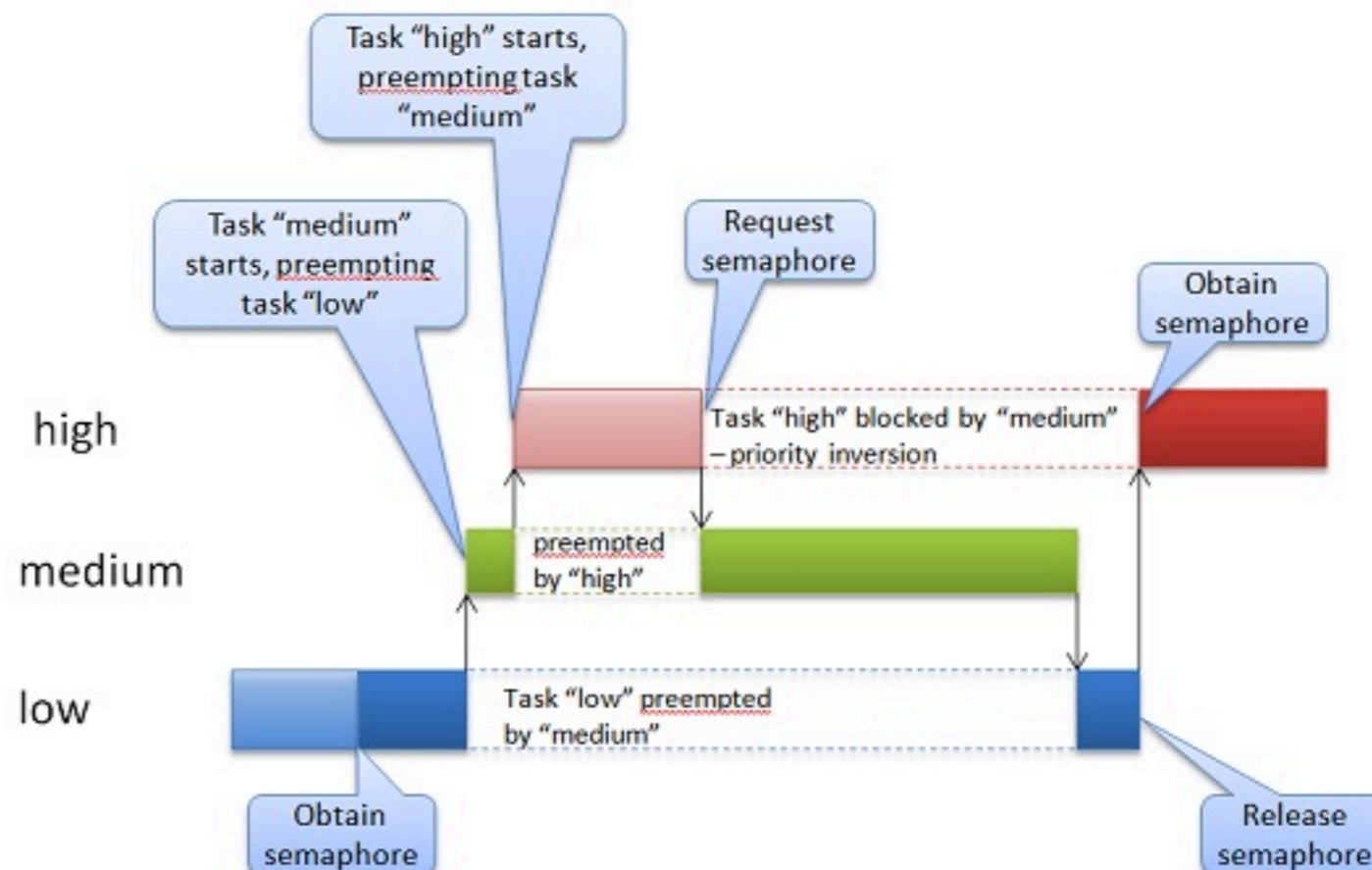
# Mars Pathfinder Incident

- **July 4, 1997 - Pathfinder lands on Mars**
  - First US Mars landing since 1976; first rover
- **A few days later...**
  - Multiple system resets occur via VxWorks RTOS
    - Watchdog timer saves the day! Sets system to safe state
    - Reproduced on ground; path uploaded to fix it
- Medium priority task delayed things
  - Developers did not turn on priority inheritance (worried about processing speed)



# Mars Pathfinder

- Now become textbook case for priority inversion
- <http://www.embedded.com/electronics-blogs/beginner-s-corner/4023947/Introduction-to-Priority-Inversion>
- [http://research.microsoft.com/en-us/um/people/mbj/Mars\\_Pathfinder/Mars\\_Pathfinder.html](http://research.microsoft.com/en-us/um/people/mbj/Mars_Pathfinder/Mars_Pathfinder.html)



# Heart Devices

- <https://www.wired.com/story/pacemaker-hack-malware-black-hat/>

 **WIRED**

A New Pacemaker Hack Puts Malware Directly on the Device

LILY HAY NEWMAN SECURITY 08.09.18 12:30 PM

**SHARE**

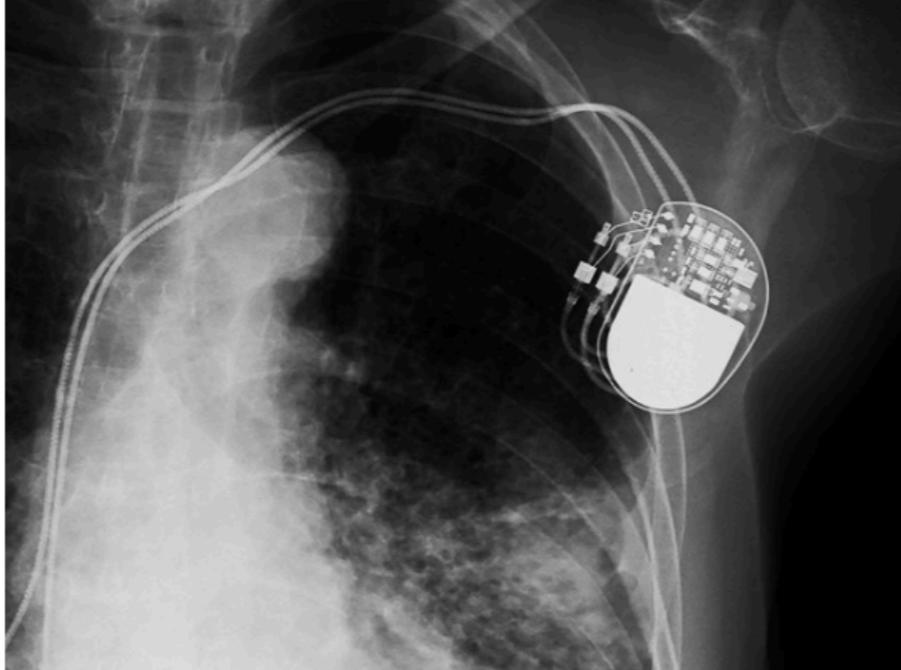
 SHARE

 TWEET

 COMMENT

 EMAIL

**A NEW PACEMAKER HACK PUTS MALWARE DIRECTLY ON THE DEVICE**



 CHOO CHIN/GETTY IMAGES

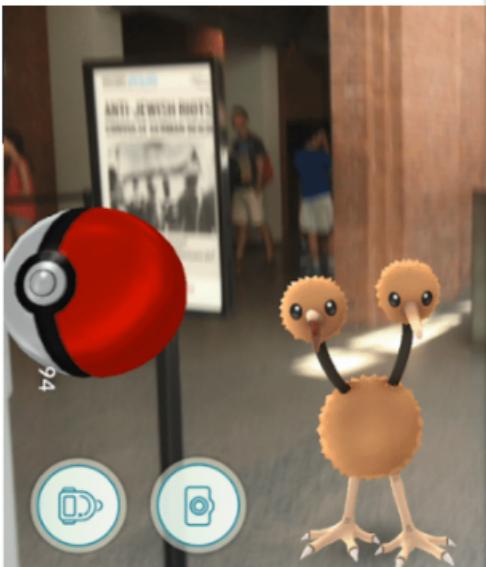
THE FIRST PACEMAKER hacks emerged about a decade ago. But the latest variation on the terrifying theme depends not on manipulating radio commands, as many previous attacks have, but on malware installed directly on an implanted

# Pokemon Go

The Switch

## Holocaust Museum to visitors: Please stop catching Pokémons here

By Andrea Peterson July 12



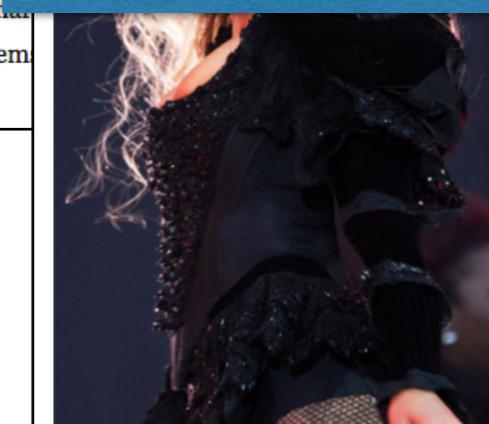
A Doduo in the Pokemon Go app found in the Holocaust Museum in

Almost everywhere you turn, it seems, people are playing Pokémon Go. Since its launch last week, the phenomenon that has fans of all ages hunting for digital creatures that appear on players' screens.

But there's at least one place that would really like you to stop: the Holocaust Museum.

The museum, along with many other landmarks, is a place where players can get free in-game items by visiting various parts of the museum.

- Two players in San Diego fell off a cliff
- Bosnian government issues warning for players to avoid minefields
- Man in Oregon was stabbed but refused to stop playing
- Man in Baltimore ran into a parked cop car
- Players in Isla Vista, CA took something floating from water thinking it was a pokeball. Actually science experiment.



## 'Pokemon Go' robber arrested

Man arrested after robbing game when robbed

Text Size: A A A



WBAL-TV 11 has arrested a 22-year-old man suspected of robbing multiple people in the University of Maryland-College Park campus earlier this

# Motivation: Embedded Systems

# Embedded Systems

- Computer system with dedicated function within larger mechanical/electrical system
- Major difference between embedded software and other types of software is the direct interface with the physical world
- Physical requirements are more complex and at times at odds with what the software can achieve
- Hard timing requirements, power, cost, size
- Debugging, simulating and testing is much more difficult for embedded software

# Embedded Flight S/W Systems

- Mostly, it's about messages and timing.
  - A late answer is a wrong answer
- Getting a message to the right place, without errors, and on time
  - If there are problems, handle the errors in a definable manner
- It concerns a little bit of everything
- It requires understanding something about all aspects of the physical system
- It requires adjusting all the compromises to achieve required acceptable results

# Embedded Flight SW Systems

- Getting the equations right is just a small part of solving the engineering project
  - Collecting the input data, processing data, outputting data
- Details matter
- Standalone operations
  - When the power is applied, the system goes directly to the application
- It allows a system to gracefully degrade
  - Handle all contingencies in a known predictable manner

Start thinking like a computer

inputs —> **algorithms** —> outputs



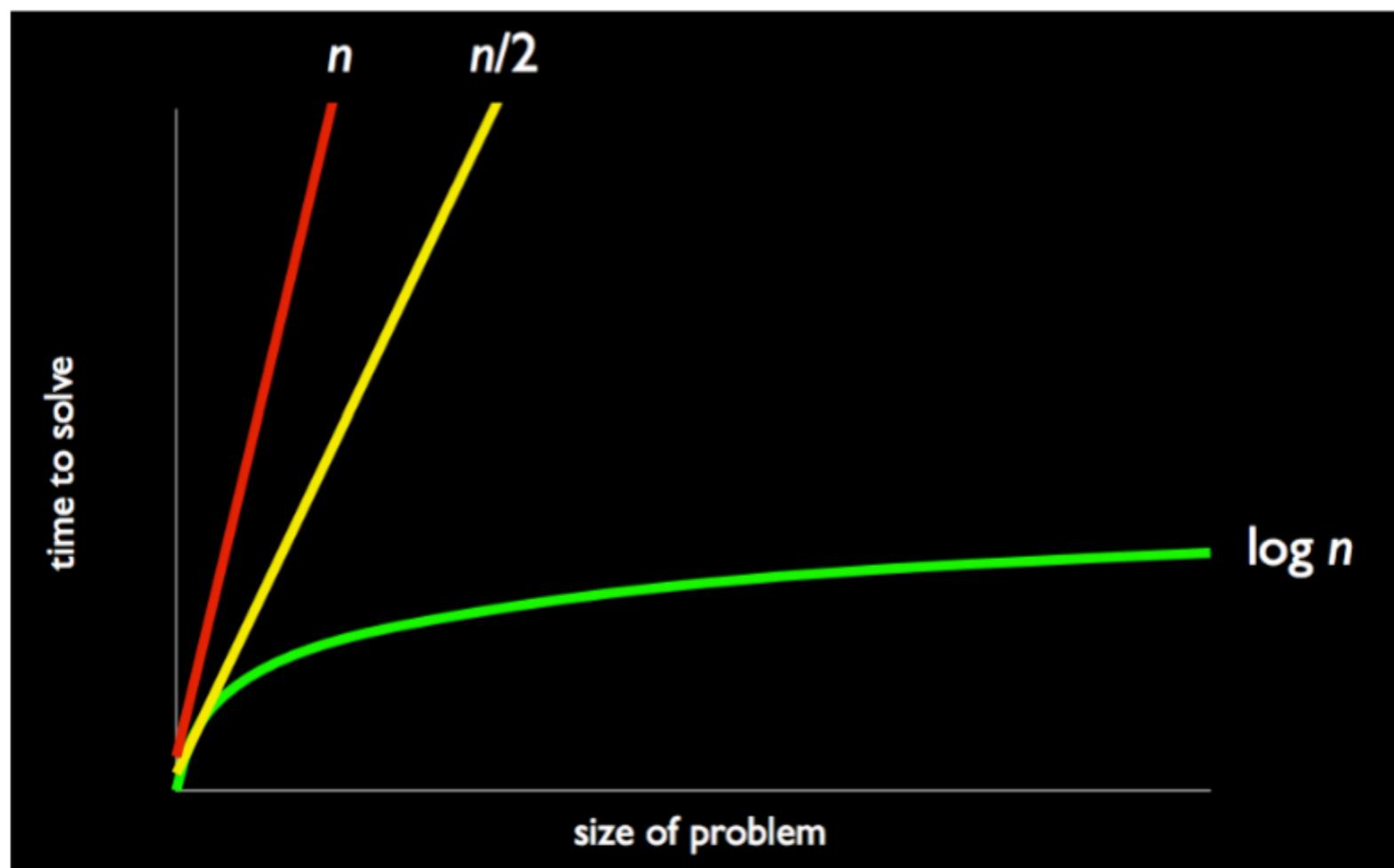
# Objective: Find “Taylor Swift” in the phone book

1. Look at each page in order (correct by slow)
2. Flip two pages at a time (almost twice as fast)

# Objective: Find “Taylor Swift” in the phone book

1. Look at each page in order (correct by slow)
2. Flip two pages at a time (almost twice as fast)
3. Open to middle (we know it's sorted alphabetically). Throw away half the phone book, keep the half we want. Keep repeating until left with single page containing Taylor Swift's phone number.

1000 pages  $\rightarrow$  10 steps



# Pseudocode describes an algorithm in English rather than any programming language

```
1 pick up phone book
2 open to middle of phone book
3 look at names
4 if "Swift" is among names
5   call Taylor
6 else if "Swift" is earlier in book
7   open to middle of left half of book
8   go to line 3
9 else if "Swift" is later in book
10  open to middle of right half of book
11  go to line 3
12 else
13   give up
```

# Pseudocode describes an algorithm in English rather than any programming language

```
1 pick up phone book
2 open to middle of phone book
3 look at names
4 if "Swift" is among names
5   call Taylor
6 else if "Swift" is earlier in book
7   open to middle of left half of book
8   go to line 3
9 else if "Swift" is later in book
10  open to middle of right half of book
11  go to line 3
12 else
13   give up
```

# Pseudocode describes an algorithm in English rather than any programming language

```
1 pick up phone book
2 open to middle of phone book
3 look at names
4 if "Swift" is among names
5     call Taylor
6 else if "Swift" is earlier in book
7     open to middle of left half of book
8     go to line 3
9 else if "Swift" is later in book
10    open to middle of right half of book
11    go to line 3
12 else
13     give up
```

# Pseudocode describes an algorithm in English rather than any programming language

```
1 pick up phone book
2 open to middle of phone book
3 look at names
4 if "Swift" is among names
5   call Taylor
6 else if "Swift" is earlier in book
7   open to middle of left half of book
8   go to line 3
9 else if "Swift" is later in book
10  open to middle of right half of book
11  go to line 3
12 else
13   give up
```

# Pseudocode describes an algorithm in English rather than any programming language

```
1 pick up phone book
2 open to middle of phone book
3 look at names
4 if "Swift" is among names
5   call Taylor
6 else if "Swift" is earlier in book
7   open to middle of left half of book
8   go to line 3
9 else if "Swift" is later in book
10  open to middle of right half of book
11  go to line 3
12 else
13   give up
```

# Attendance

Hello, World

Sign up for Piazza here:

[piazza.com/umd/fall2024/nae380](https://piazza.com/umd/fall2024/nae380)