



# Introduction to Information Systems - Understanding the digital world

**7** Operating System and Software Systems

Liang Zhao

ILA, Doshisha University

12001102, Fall, 2024





# Today's schedule

- Review of the Scratch game (10')
- BIOS (5')
- Operating System & Software (55')
- Mini test (15')
- Information and homework (5')

# Review of the game

## Received (before 10am today):

1120201012 1120220016 1120221019 1120230004 1120230011  
1120230013 1120231003 1120231004 1120231008 1120231012  
1120231015 1120240002 1120240005 1120240008 1120240011  
1120240012 1120240014 1120240015 1120240018 1120240019  
1120240021 1120240023 1120240025 1120240028 1120240029  
1120240031 1120240032 1120241002 1120241008 1120241011  
1120241017 1120241019 1120241020 1120241021 1120241029

## No access or no submission:

Late submission is better than no submission!

1120191006 1120200019 1120211008 1120211015 1120211018  
1120211023 1120220003 1120221005 1120230001 1120230009  
1120230019 1120231009 1120231014 1120231017 1120240001  
1120240007 1120240009 1120240013 1120240016 1120240022  
1120240027 1120240030 1120241009 1120241024 1120241026

# Some of my favorite submissions



Birds

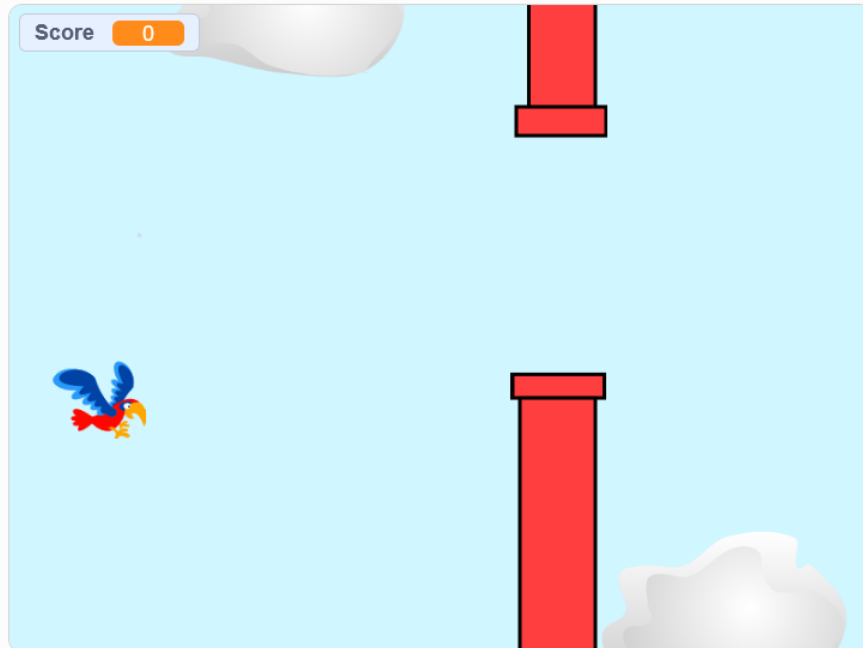
by MatthiasValentin

Simple yet fun

🔄 中を見る



Score 0



使い方

Help Polly the parrot to reach the goal, which lies somewhere in the far, far distance.  
Press "Space" to make Polly flop his wings and try to avoid any of the pipes that get into his way.

メモとクレジット

I did some research on other projects, trying to rebuild the Flappy Birds game. In the future I would like to install a feature that also increases the speed over time, a well as presents different pipes.



# Some of my favorite submissions



The screenshot shows a Flappy Bird-style game titled "Flappy Chap". A small white bird character is flying through a teal background with white particles. It is navigating between green pipes with black outlines. The game is running, as indicated by the "実行" (Run) button in the top left corner of the game area.

 **Flappy Chap**  
by [nicomakesgames1](#)

**Challenging but fun**

[🔄 中を見る](#)

**使い方**


- [1] Press space to jump
- [2] Jump through the pipes to increase your score
- [3] The score is indicated at the top left
- [4] Touching a pipe will end the game
- [5] The best score gets a cookie

Created for NOV 4th, 2024

**メモとクレジット**

Inspired by IOS hit game: 'Flappy Bird'

# Some of my favorite submissions



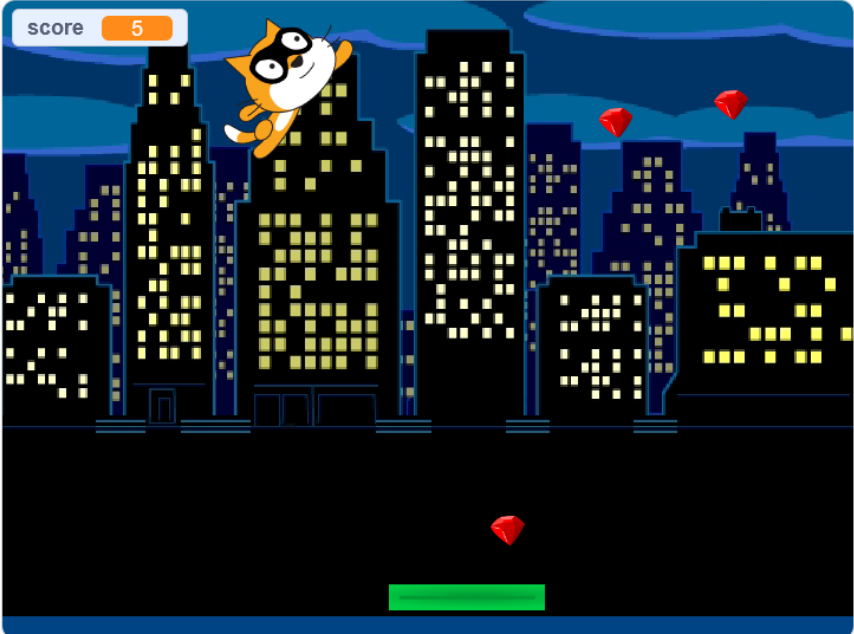
## Secret Mission!

by [jellie77](#)

I love it the best!

中を見る

使い方



score 5


Welcome to Secret Mission!

- ★ Mission Objective: Keep the cat safe and score big! Move your paddle with the mouse to catch the cat and earn points.
- ★ Special Boost: Watch out for red rubies! Each ruby adds 2 bonus points to your score.
- ★ Unlock Surprises: Hit target scores to reveal exciting surprises as you play!

Ready for the challenge? Let the mission begin!



# Some of my favorite submissions



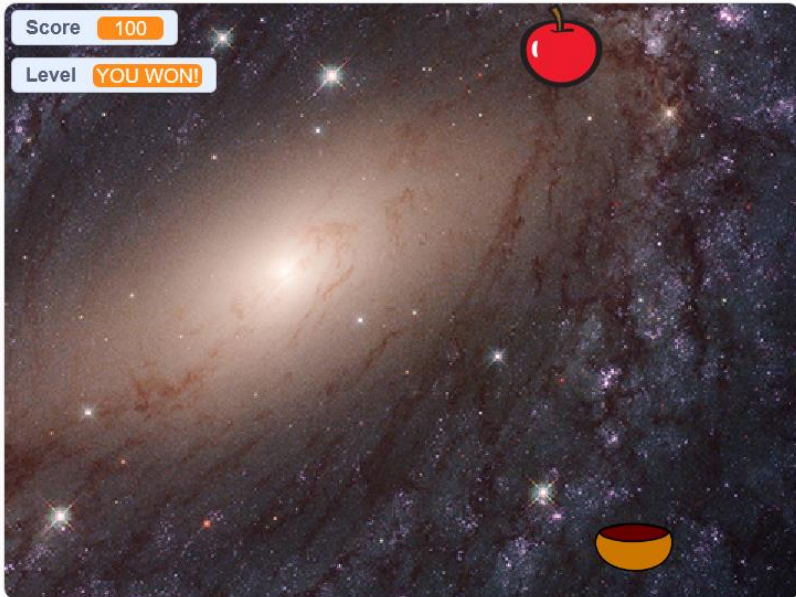
Catch the Food!  
by [mrko7853](#)

Professional game!

中を見る

🚩 📌

🔗 使い方



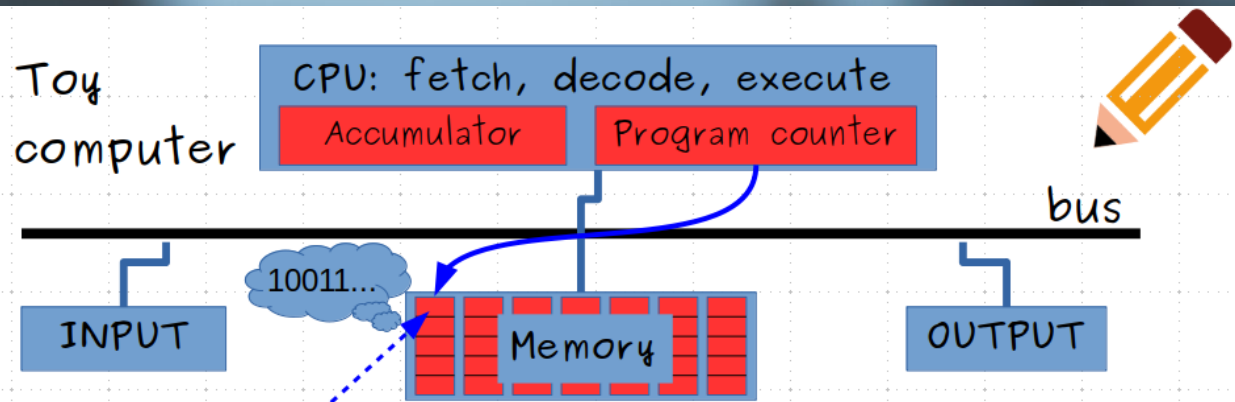
Score 100  
Level YOU WON!

Use your keyboard's left and right arrows to move the bowl and catch the falling foods!  
There are multiple levels. Try to reach the end of it. The higher the level, the faster the speed will go. Good luck!

👍 0 🌟 0 🔄 0 👁 6

© 2024年10月29日

リンクをコピー



| label | instruction | description   |
|-------|-------------|---|
|       | get         | get a number from keyboard into accumulator                     |
| L     | print       | print contents of accumulator                                   |
|       | load Val    | load accumulator with Val (Val unchanged)                       |
|       | store M     | store contents of accumulator into memory location called M     |
|       | add Val     | add Val to contents of accumulator (Val unchanged)              |
|       | sub Val     | subtract Val from contents of accumulator (Val unchanged)       |
|       | goto L      | go to instruction labeled L                                     |
|       | ifpos L     | go to instruction labeled L if accumulator is $\geq$ zero       |
|       | ifzero L    | go to instruction labeled L if accumulator is zero              |
|       | stop        | stop running  |
| M     | Num         | before program runs, set this memory location (called M) to Num |

# BIOS

- **Basic Input/Output System (BIOS):** firmware used to perform hardware initialization during the booting process (power-on startup), and to provide runtime services for operating systems and programs. (<https://en.wikipedia.org/wiki/BIOS>)
- In short, the firmware that connects hardware and software.
- <https://www.youtube.com/watch?v=D1R2ttrvbdI> (6', optional)



# OS (Operating System)

With OS, we don't need to write millions of lines of code by ourselves in order to use a computer.

**CPU (task management)**

**Disk (HDD, SSD, etc) and file**

**RAM (memory)**

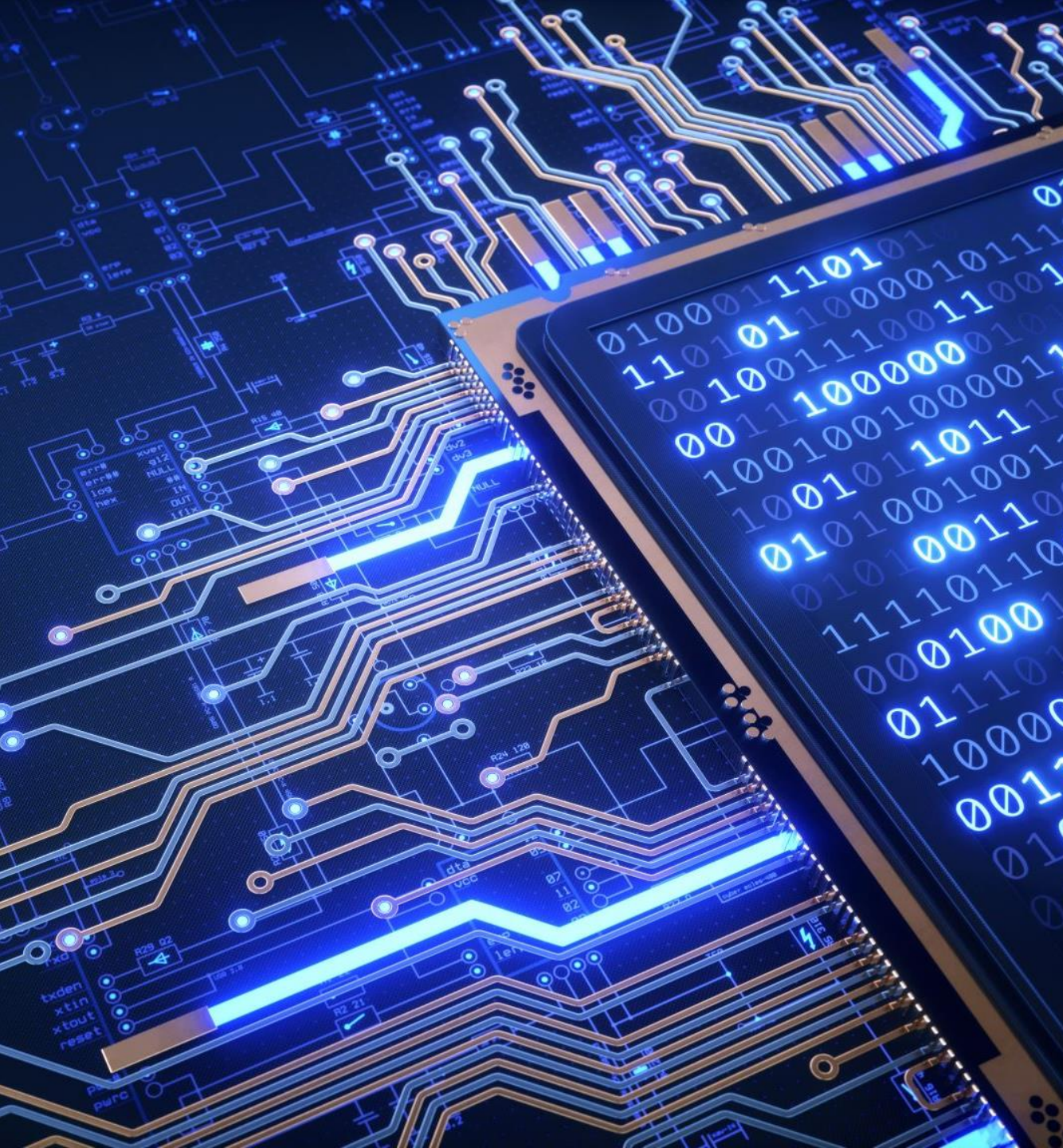
**Devices (monitor, keyboard, mice, printer, etc)**



# Memory (RAM) management

Loads programs and data into memory.  
Swaps them to disk when memory is low.  
Protects the programs from interfering.





# CPU (task management)

Single-task (DOS, etc)

Multi-task (Unix/Linux, Windows, Mac OS, etc)

Crash course -> Computer Science #18 (14')

<https://www.youtube.com/watch?v=26QPDBe-NB8>





# Disk and file management

- File System (FAT, FAT32, NTFS, ext4, APFS, etc)
- Directory (folder): special container file
- Executable files (Word, Photoshop, etc)
- Documents (txt, doc, jpg, mp3, html, etc)
- System files (lib, sys, etc)
- Extension distinguishes types: doc/docx -> Word document, jpg -> JPEG file (editable by Photoshop etc), exe -> executable file, ...

Crash Course -> Computer Science #20 (12')

<https://www.youtube.com/watch?v=KN8YgJnShPM>

# Comment of "path" and location of a file



**Absolution** path: e.g., `C:¥Users¥liang¥Desktop¥book.docx` (Windows, starting with the **drive**) and `/home/liang/Desktop/book.docx` (UNIX/Linux/Mac, starting with the root `"/`)



**Relative** path: path that is related to the **working directory** (W.D., directory where we are working with). E.g., if W.D. is `C:¥Users¥liang`, then `Desktop¥book.docx` means `C:¥Users¥liang¥Desktop¥book.docx`.



URI extends this notation with protocol and server name: e.g., (where **protocol** = https, **server name** = `ila.doshisha.ac.jp`) `https://ila.doshisha.ac.jp/en/index.html`



# Shell

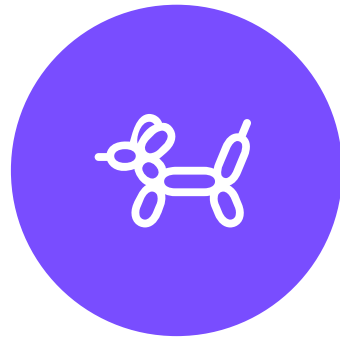
**User <-> OS interface software**

**Crash course -> Computer Science 22 (11')**

<https://www.youtube.com/watch?v=4RPtJ9UyHS0>

# Homework

---



## WATCH VIDEOS

Watch the movies mentioned so far if you have not  
(You are not expected to understand everything)



## READ CHAPTERS 1-6 (IF YOU HAVE NOT)

# Appendix: Advanced topics

---



**Device driver:** program for a special hardware. Ex: printer drivers provide detailed control (two-sided printing, etc).



**System call:** function provided by the OS to apps. Ex: input, drawing on the display (DirectX, OpenGL), network function, etc.



**Memory management:** Crash Course -> Computer Science #19  
<https://www.youtube.com/watch?v=TQCr9RV7twk>



**Other Uses:** Linux, FreeBSD, Android, etc