數電實驗 Final Project

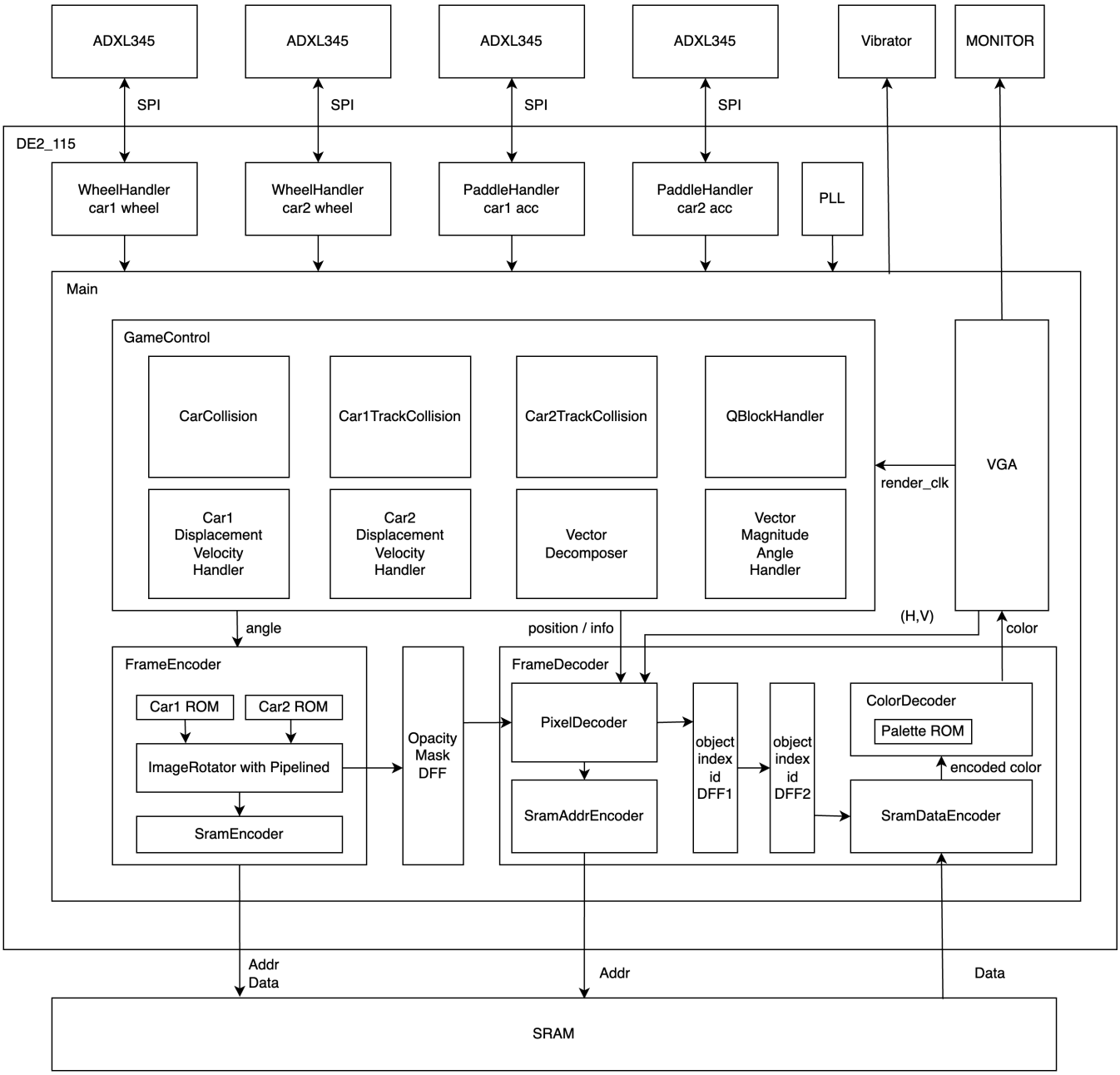
B10901176 蔡弘祥 B10901163 張顥譽 B10901179 鄭承瑞

1. File Structure

```
src
├── AccHandler
│   ├── ADXL345_SPI.sv
│   └── accHandler.sv
├── DE2_115
│   ├── DE2_115.qsf
│   ├── DE2_115.sdc
│   ├── DE2_115.sv
│   ├── Debounce.sv
│   └── SevenHexDecoder.sv
├── frameDecoder
│   ├── colorDecoder.sv
│   ├── frameDecoder.sv
│   ├── palette
│   │   ├── barDigitPalette.sv
│   │   ├── barPalette.sv
│   │   ├── car1Palette.sv
│   │   ├── car2Palette.sv
│   │   ├── carCirclePalette.sv
│   │   ├── idleBackgroundPalette.sv
│   │   ├── losePalette.sv
│   │   ├── mapPalette.sv
│   │   ├── qblockPalette.sv
│   │   ├── startPalette.sv
│   │   └── winPalette.sv
│   ├── pixelDecoder.sv
│   ├── sramAddrEncoder.sv
│   └── sramDataDecoder.sv
├── frameEncoder
│   ├── frameEncoder.sv
│   ├── lut
│   │   ├── car1LUT.sv
│   │   └── car2LUT.sv
│   ├── rotate
│   │   ├── imageRotator.sv
│   │   └── rotateImageCoor.sv
│   └── sramEncoder.sv
├── gameControl
│   ├── basicTrackCollision.sv
│   ├── carCollision.sv
│   ├── gameControl.sv
│   ├── qBlock.sv
│   └── trackCollision.sv
```

```
├── main.sv
├── math
│   ├── atanLUT.sv
│   ├── random.sv
│   ├── rotateXY.sv
│   └── sqrt.sv
├── pkg
│   ├── gamePkg.sv
│   ├── sramPkg.sv
│   └── trackPkg.sv
├── qsys
│   ├── qsys.bsf
│   ├── qsys.cmp
│   ├── qsys.html
│   ├── qsys.xml
│   ├── qsys_bb.v
│   ├── qsys_generation.rpt
│   ├── qsys_generation_previous.rpt
│   ├── qsys_inst.v
│   ├── qsys_inst.vhd
│   └── synthesis
│       ├── qsys.debuginfo
│       ├── qsys.qip
│       ├── qsys.v
│       └── submodules
│           ├── altera_reset_controller.sdc
│           ├── altera_reset_controller.v
│           ├── altera_reset_synchronizer.v
│           ├── qsys_altpll_0.v
│           └── qsys_usb_0.v
└── vga.sv
```

2. System Architecture



3. Image Process

- Use python to convert 24-bit to 4-bit
  - With alpha channel: convert to 15 colors, one is used to save transparent or not
    - Car
  - Without alpha channel: convert to 16 colors
    - Map
    - Bar
    - Bar Digit
    - Start Caption
    - Win Caption
    - Lose Caption
    - Idle Background
    - QBlock

- Use python to convert ROM (LUT) or binary files (.bin)
  - ROM (LUT): Verilog module
    - Car
  - Binary files (.bin): DE2-115 Control Panel pre-write
    - Map
    - Bar
    - Bar Digit
    - Start Caption
    - Win Caption
    - Lose Caption
    - Idle Background
    - QBlock

4. Sram

- Total 2^20 addresses, each address is composed of 16-bit data
- Each pixel is encoded into 4-bit, each address saves 4 pixels
- Decode them back to 24-bit when rendering
- Total used 1.6MB/2MB

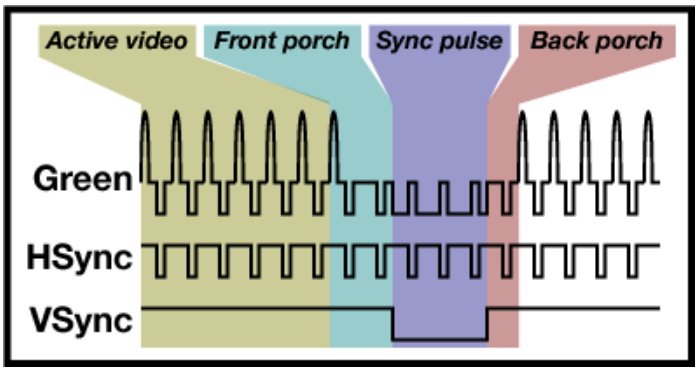
Object	# of pixels	# of address used
Map	1600*800	320000
Bar	1600*100	40000
Bar Digit	26*38*10	2470
Start Caption	664*56	9296
Win Caption	200*60	3000
Lose Caption	200*60	3000
Idle Background	1600*900	360000
QBlock	40*40*4	1600
Car	40*40*2	800

5. Math

- Vector Rotation / Angle / Magnitude
  - Cordic Algorithm
  - Data scale up to increase accuracy
  - 16 stages w/wo pipelined
- Square root
  - Support Fixed Point Number
- Random
  - LFSR from LAB1

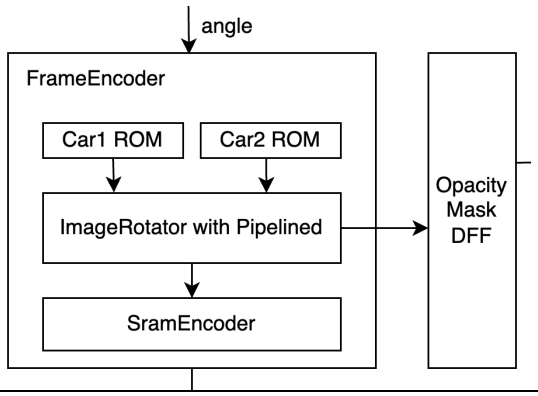
6. VGA

- Two modules to help rendering
  - FrameEncoder
    - Front porch state
    - Sync pulse state
    - Back porch state
  - FrameDecoder
    - Active state



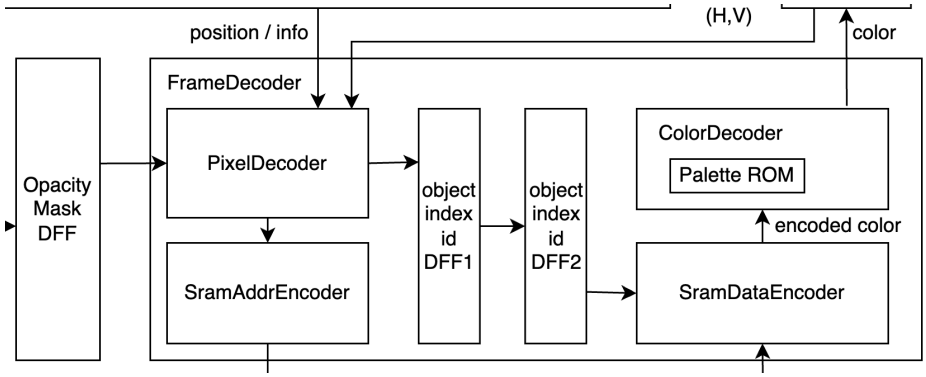
7. FrameEncoder

- Handle car rotation and opacity mask
- For faster rendering
  - Save the original car images in ROM
  - Use pipelined cordic to finish rotation before VGA active
  - Directly generate the opacity masks so that we only need to access Sram once when rendering



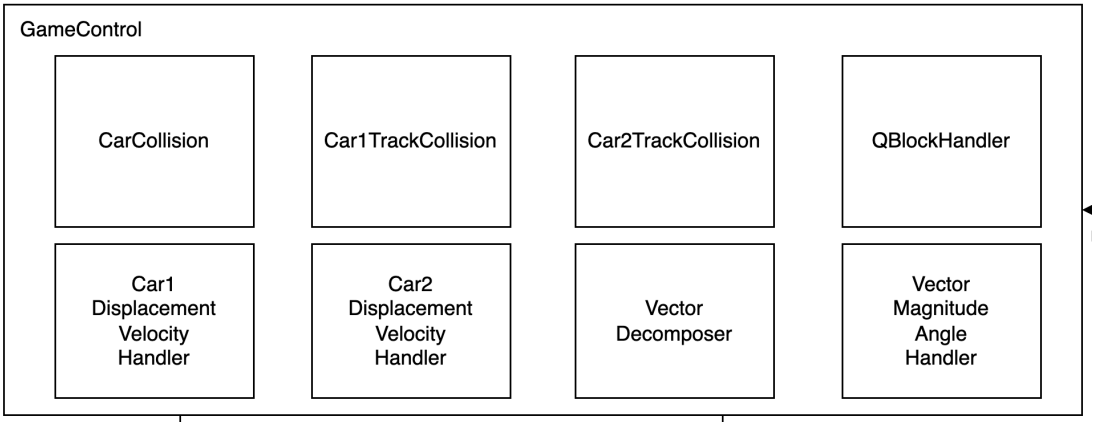
8. FrameDecoder

- PixelDecoder
  - Determine which object to render and the pixel index
- SramAddrEncoder
  - Use object ID and pixel index to set Sram address
- SramDataDecoder
  - Use pixel index to find which 4-bit to read
- ColorDecoder
  - Use Palette ROM with object ID as input to decode the color back to 24 bits



9. GameControl

- Car Status
  - Position
  - Velocity Magnitude
  - Car Angle
  - Lap
  - Mass Level
- Two Cars Collision
  - Elastic Collision
  - Decompose the velocity into x, y components
  - Calculate the velocity after collision
  - Calculate the final angle (arctan) and magnitude (square root)
- Car/Track Collision
- Qblock

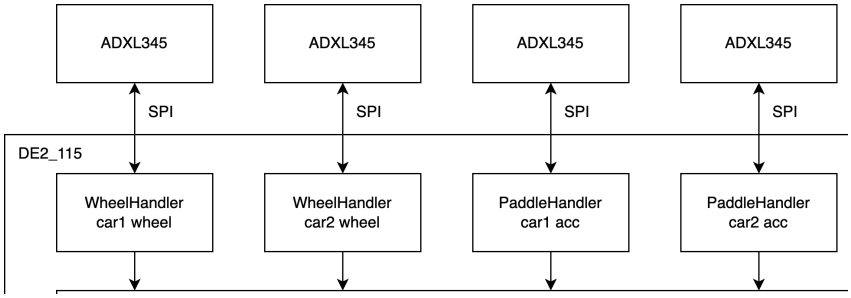


$$\mathbf{v}_1' = \mathbf{v}_1 - \frac{2m_2}{m_1+m_2}\,\frac{\langle \mathbf{v}_1-\mathbf{v}_2,\,\mathbf{x}_1-\mathbf{x}_2\rangle}{\|\mathbf{x}_1-\mathbf{x}_2\|^2}\,(\mathbf{x}_1-\mathbf{x}_2)$$

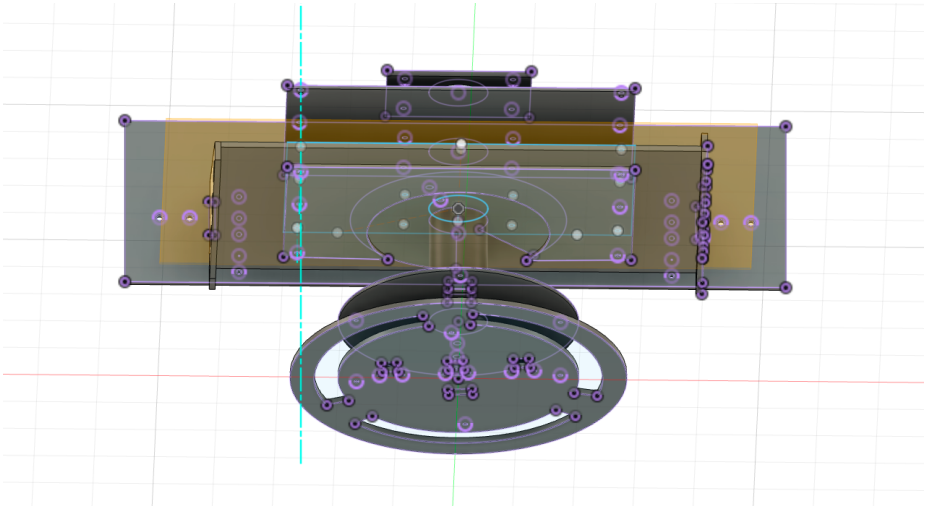
$$\mathbf{v}_2' = \mathbf{v}_2 - \frac{2m_1}{m_1+m_2}\,\frac{\langle \mathbf{v}_2-\mathbf{v}_1,\,\mathbf{x}_2-\mathbf{x}_1\rangle}{\|\mathbf{x}_2-\mathbf{x}_1\|^2}\,(\mathbf{x}_2-\mathbf{x}_1)$$

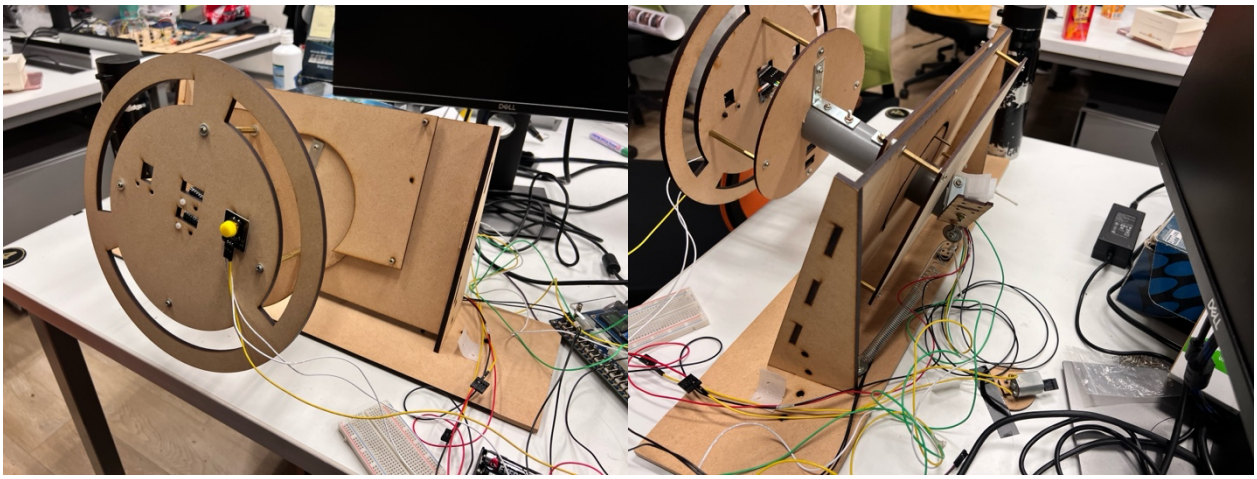
10. ADXL345

- SPI 4 wire
- 100kHz clock
- 10-bit precision
- Read gx, gy and convert to the corresponding rotation/acceleration

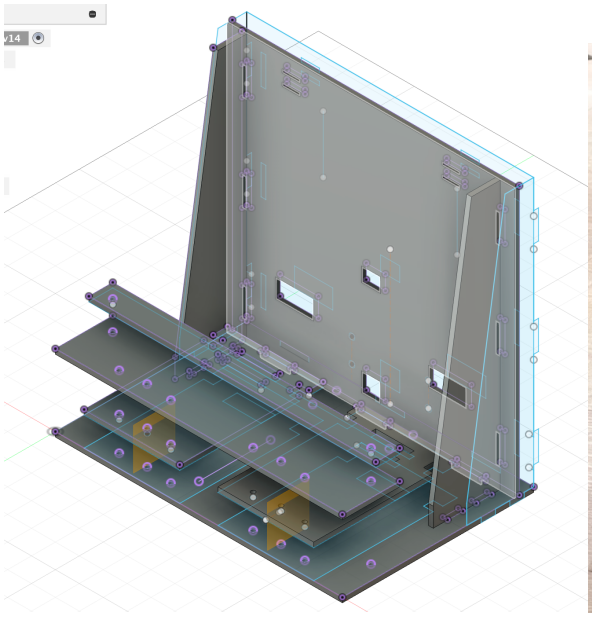


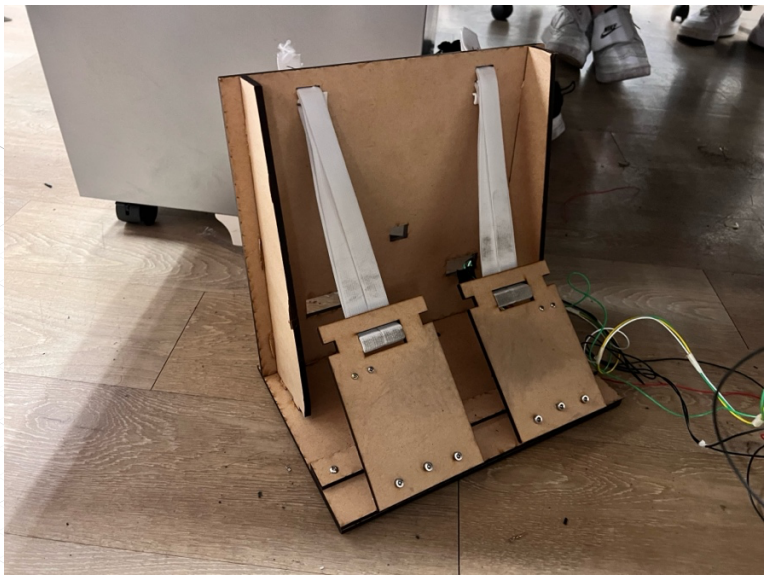
11. Wheel





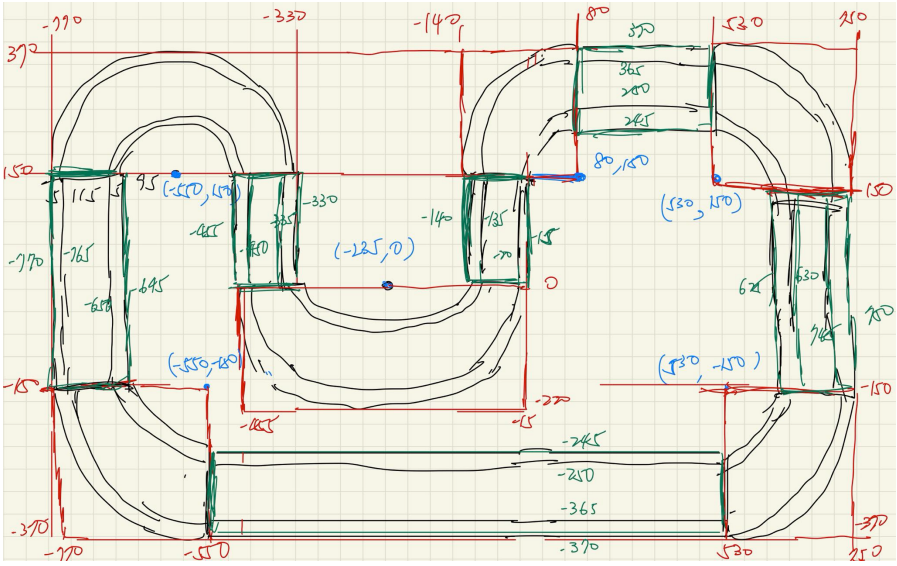
12. Paddle





13. Track

- Use specific function and bounding box to define the map
  - Horizontal Line
  - Vertical Line
  - Circle
- Different types of track
  - Ordinary
  - Sand
  - Rock



Ref1: 112-1 修課的石博允、UCB 強者、渠立宇

Ref2: 1482 低能兒頻道，歡迎訂閱按讚開啟小鈴噹，有所有 lab 的教學影片

https://www.youtube.com/@1482dinenger