




# Mario Bros Classic

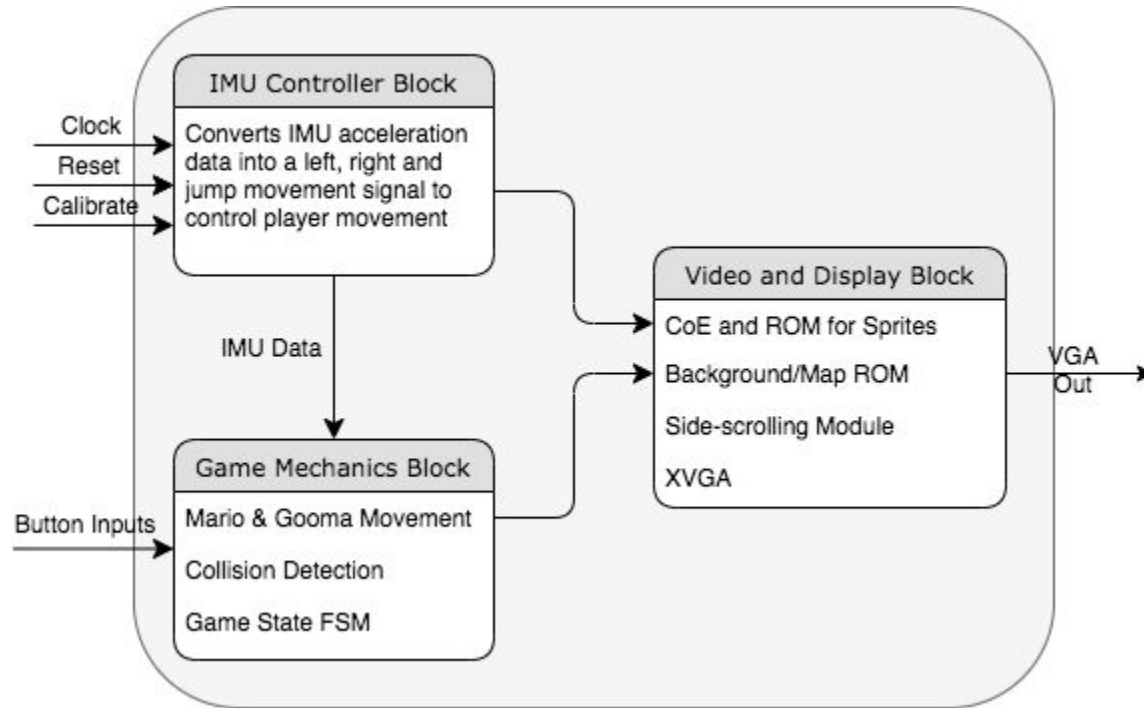
Isabelle Chong  
Jose Guajardo  
Nancy Hidalgo



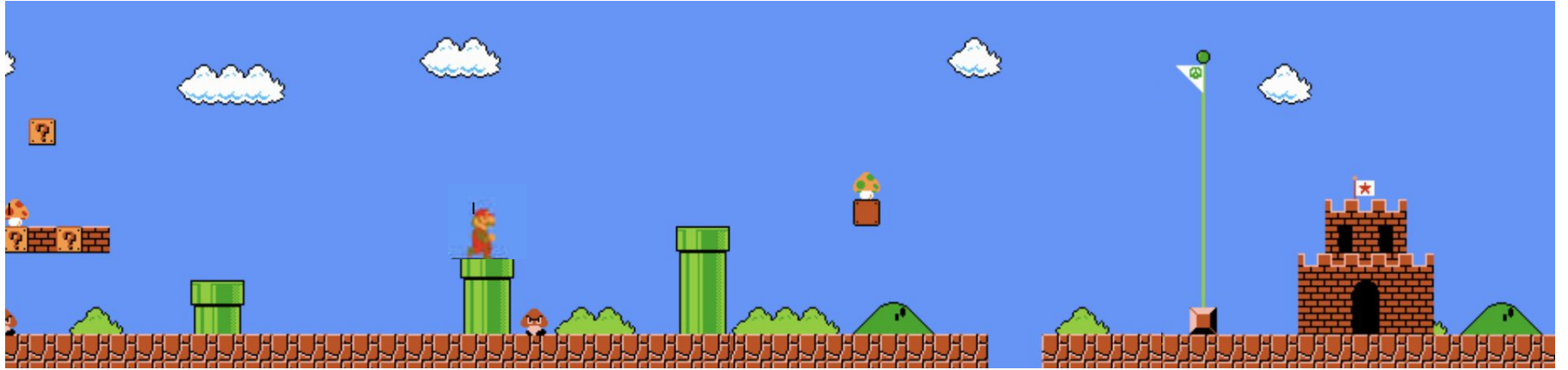
# What is Mario Bros Classic?



# System Block Diagram



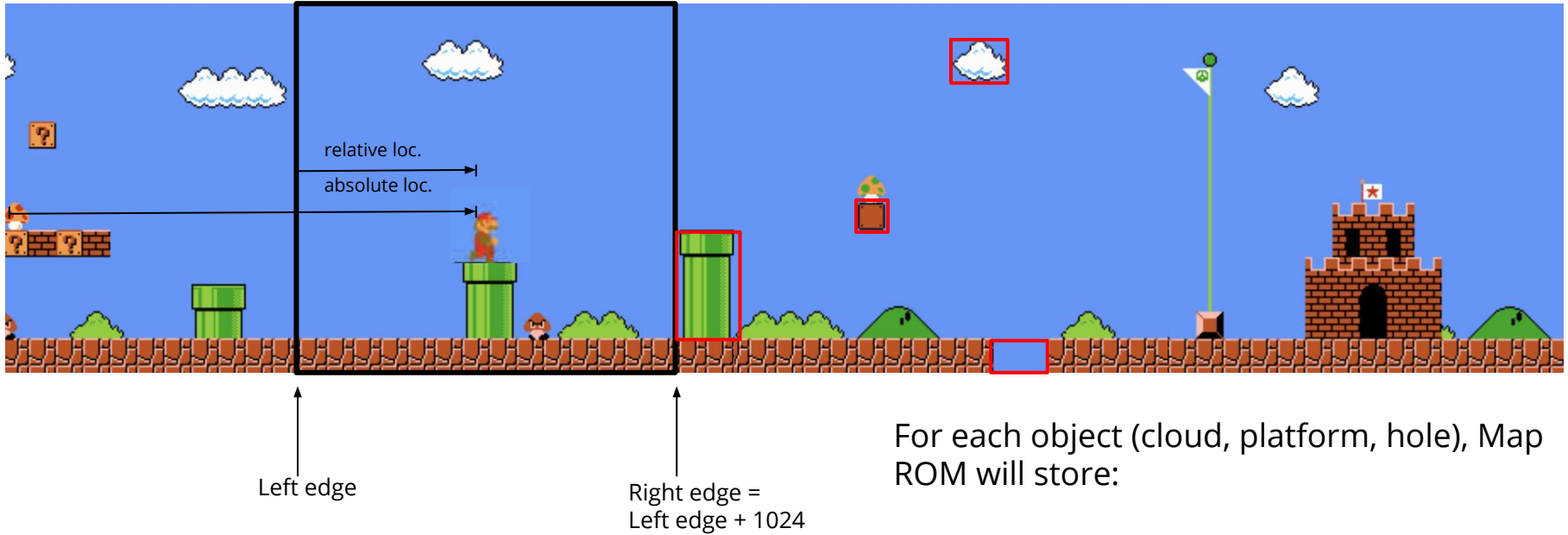
# Our Implementation - Terminology







# Our Implementation - Terminology



For each object (cloud, platform, hole), Map ROM will store:

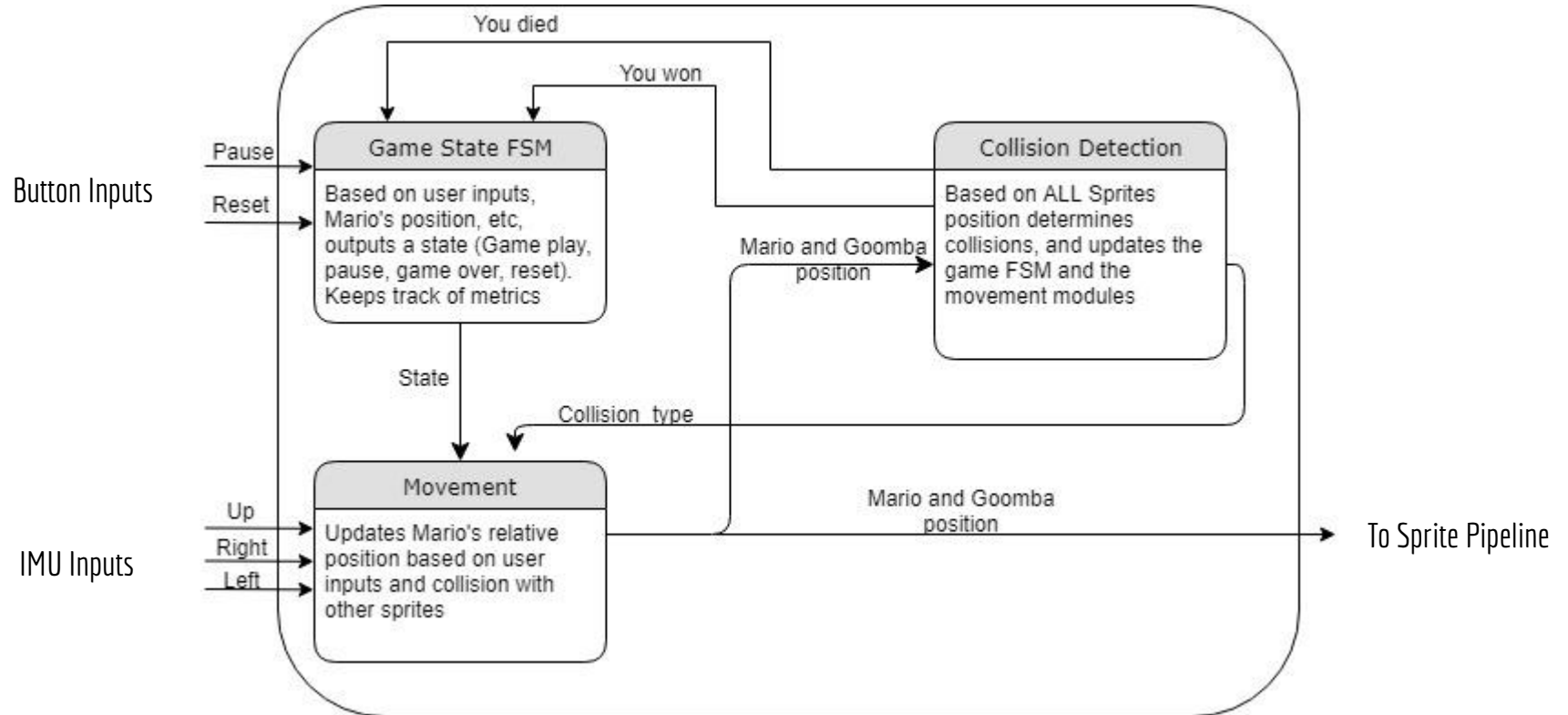
{abs\_loc\_x, abs\_loc\_y, width, height, sprite\_id}

# Different ROMs

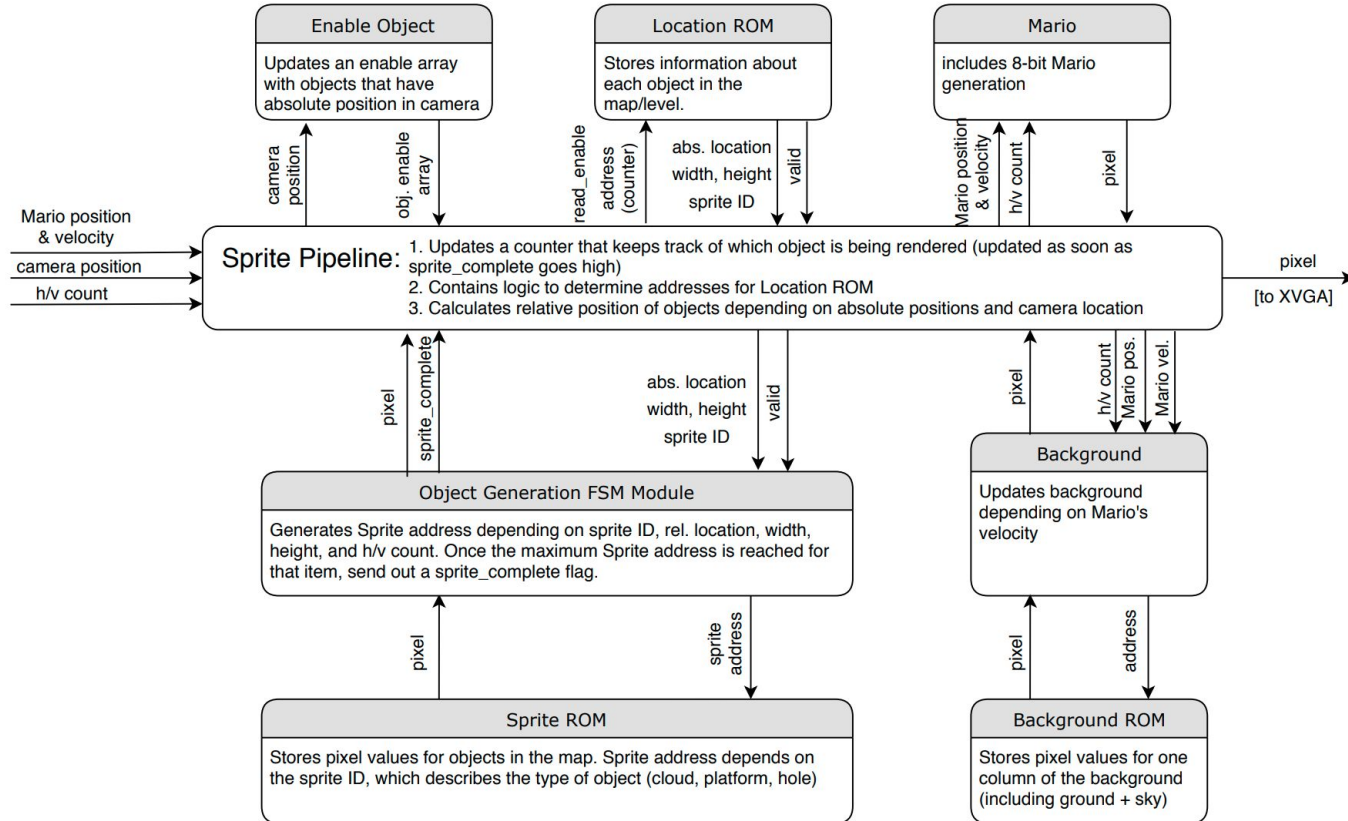
- Location ROM
- Sprite ROM
- Background ROMs



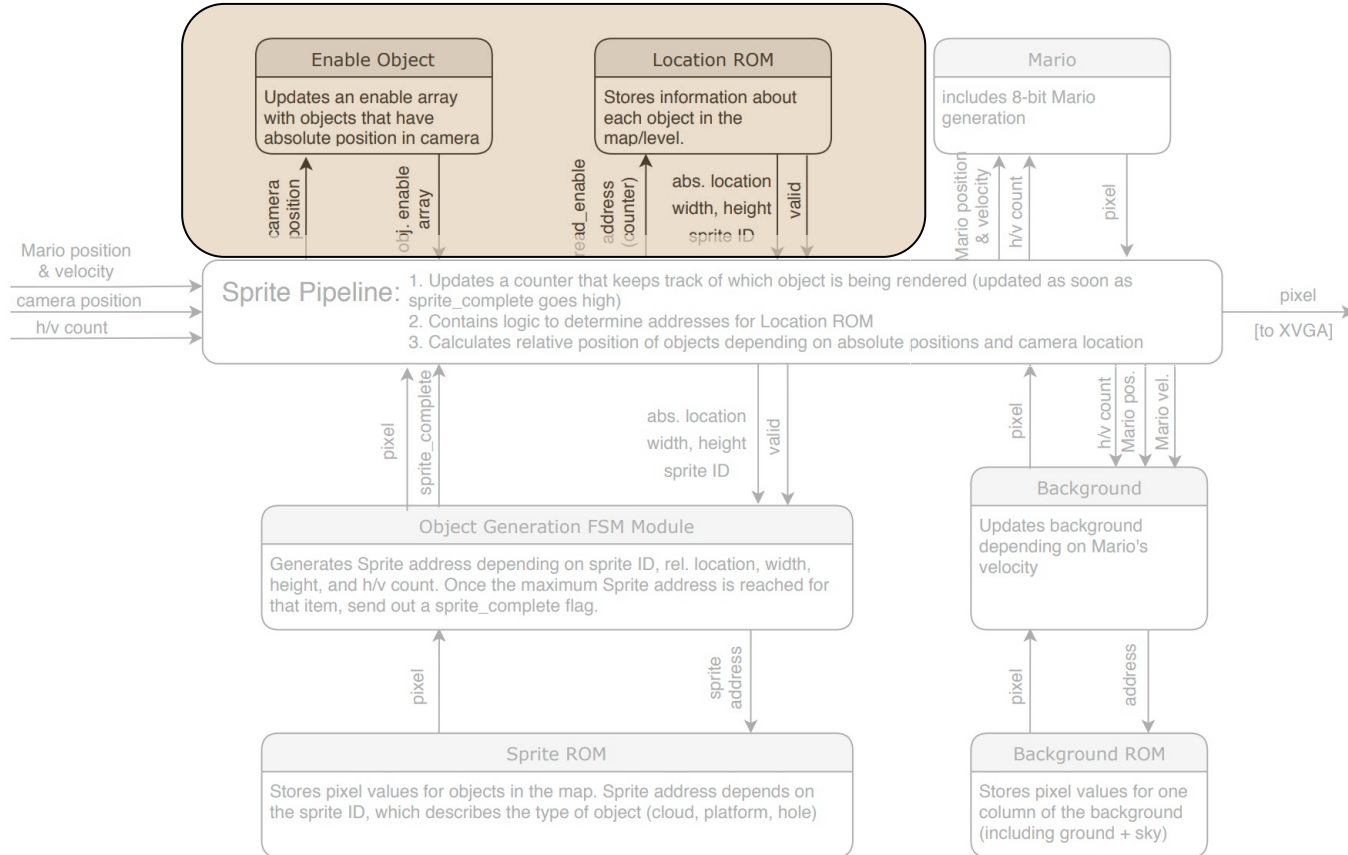
# Game Mechanics



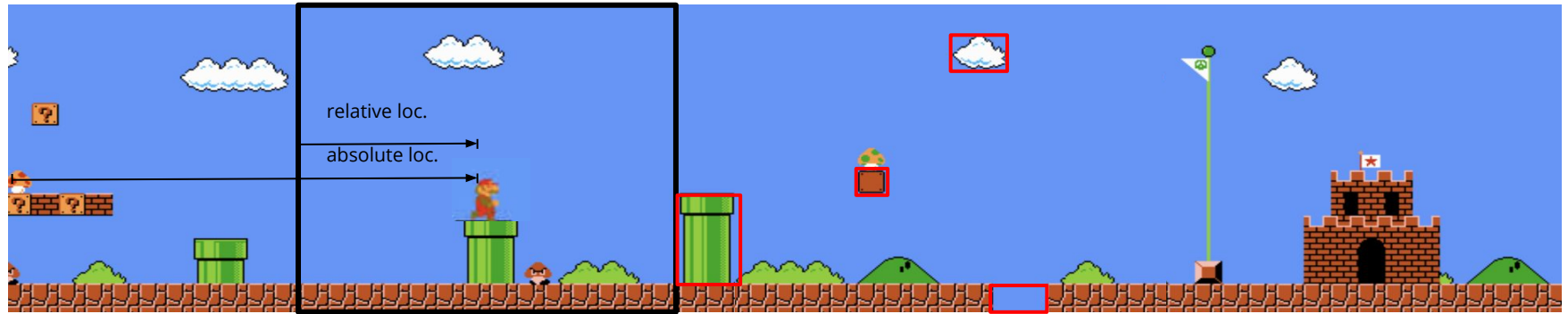
# Video and Graphics Block



# Enable Object and Location ROM Modules

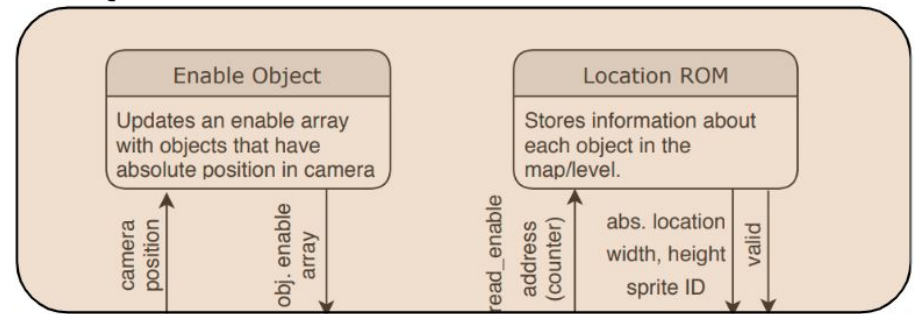


# Which objects should be displayed, and where?

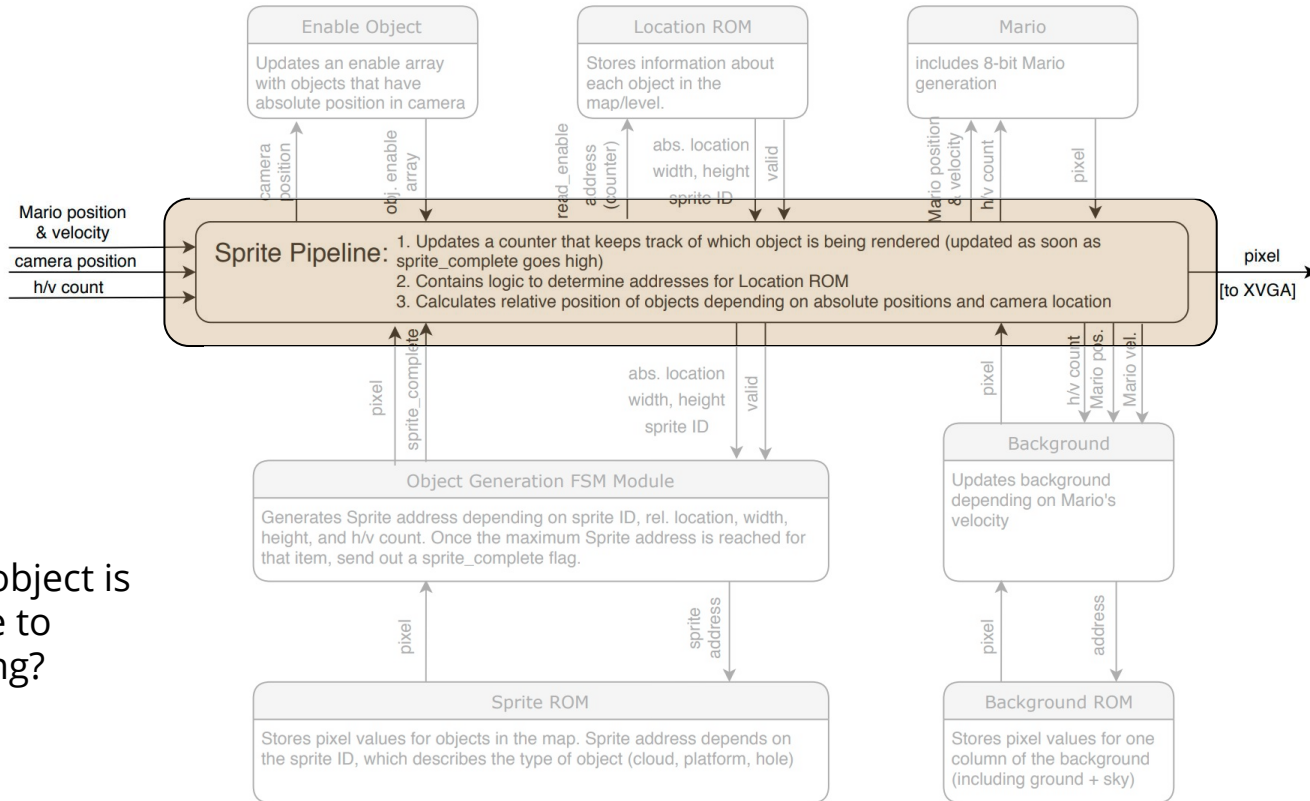


Left edge

Right edge =  
Left edge + 1024

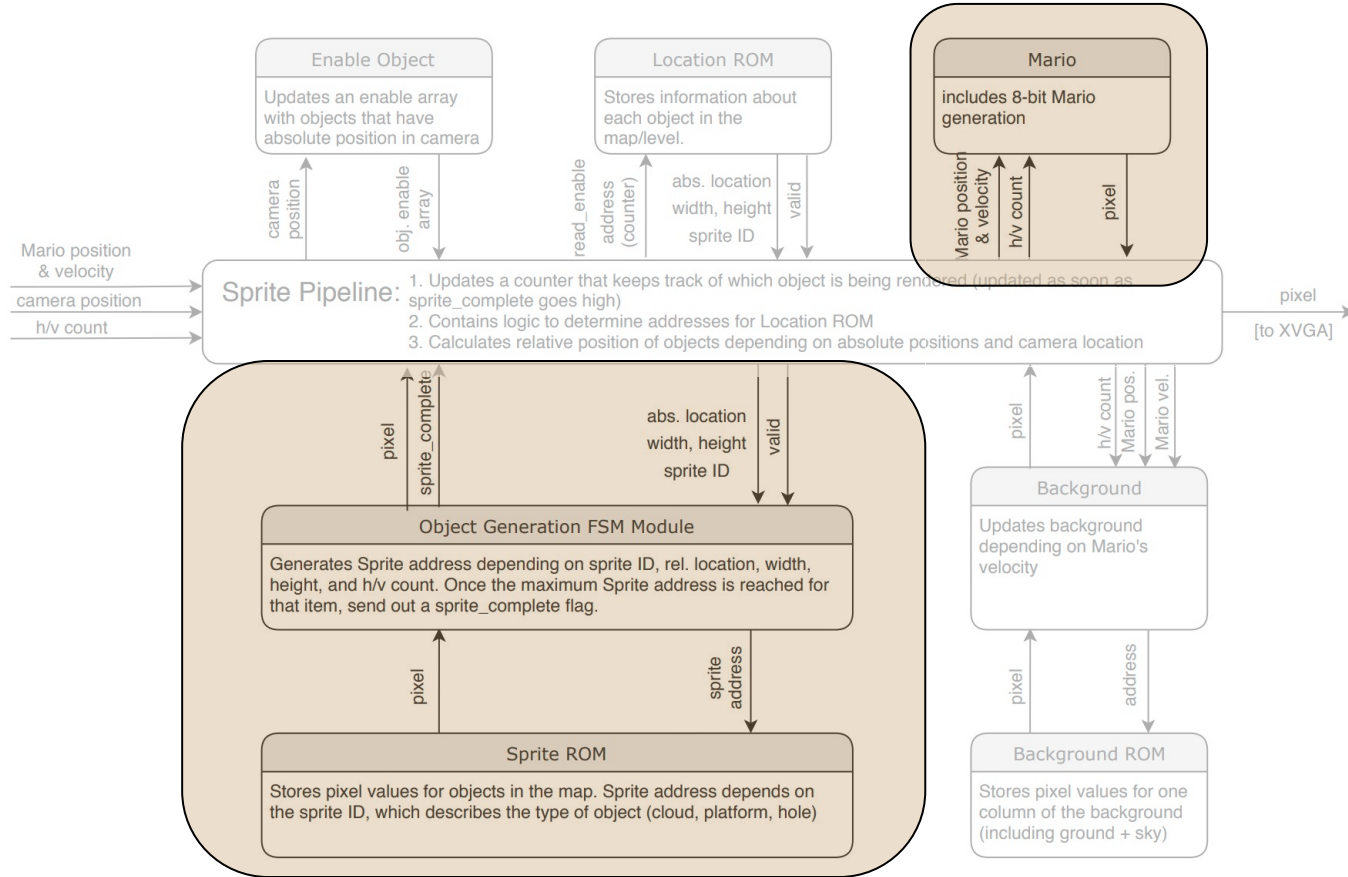


# Sprite Pipeline

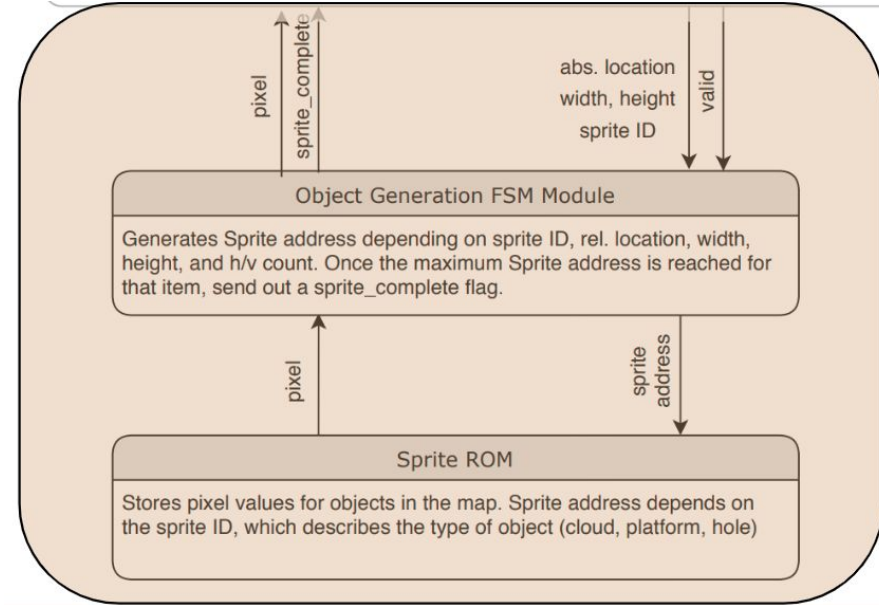
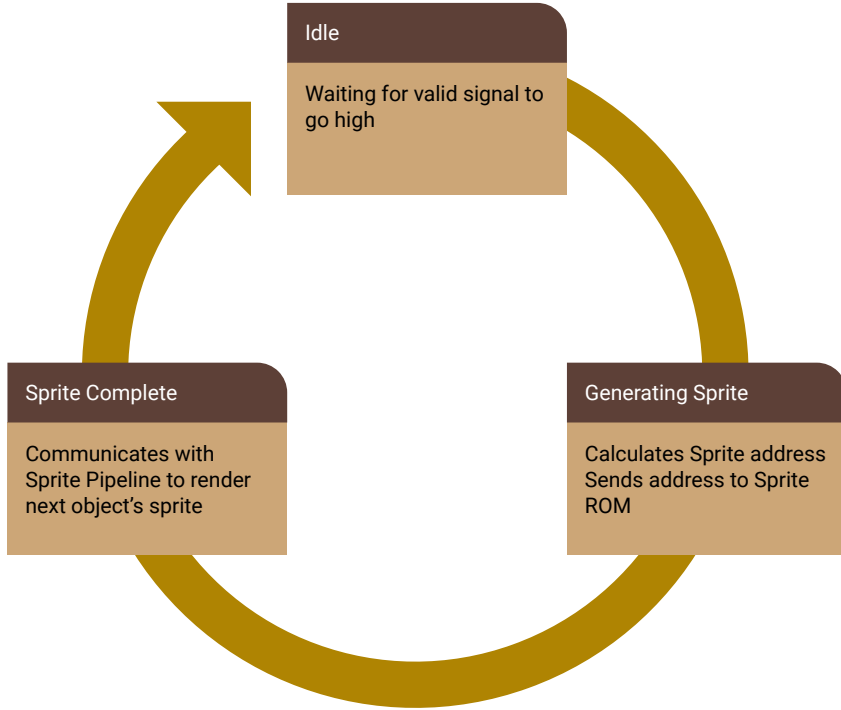


What if an object is moving due to side-scrolling?

# Object and Sprite Generation Modules

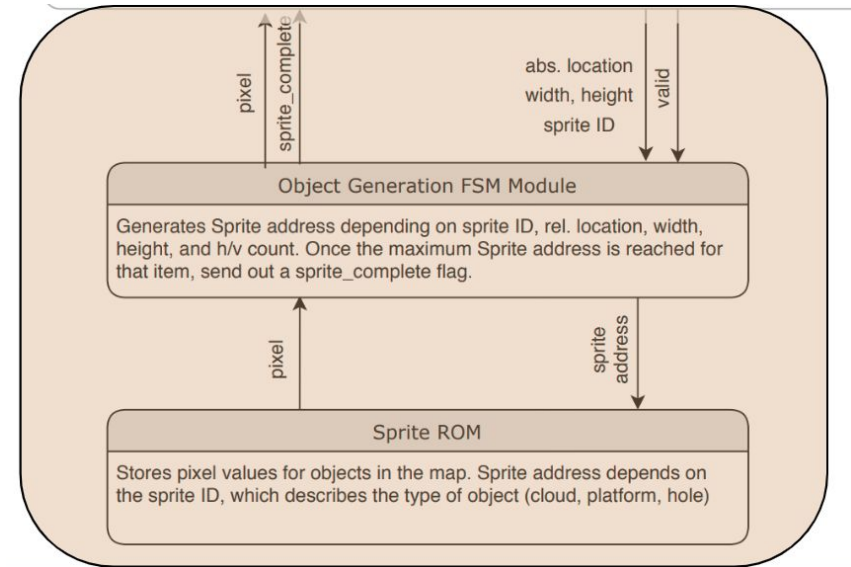
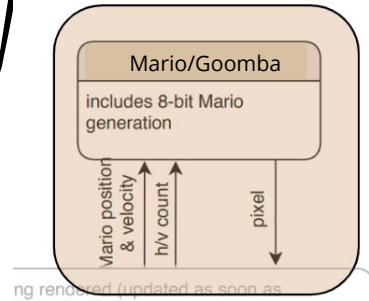
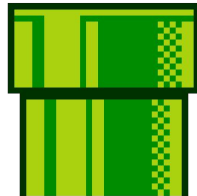
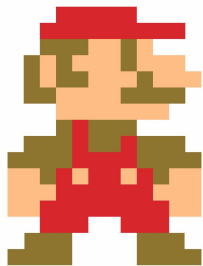


# Accessing and Generating Sprites



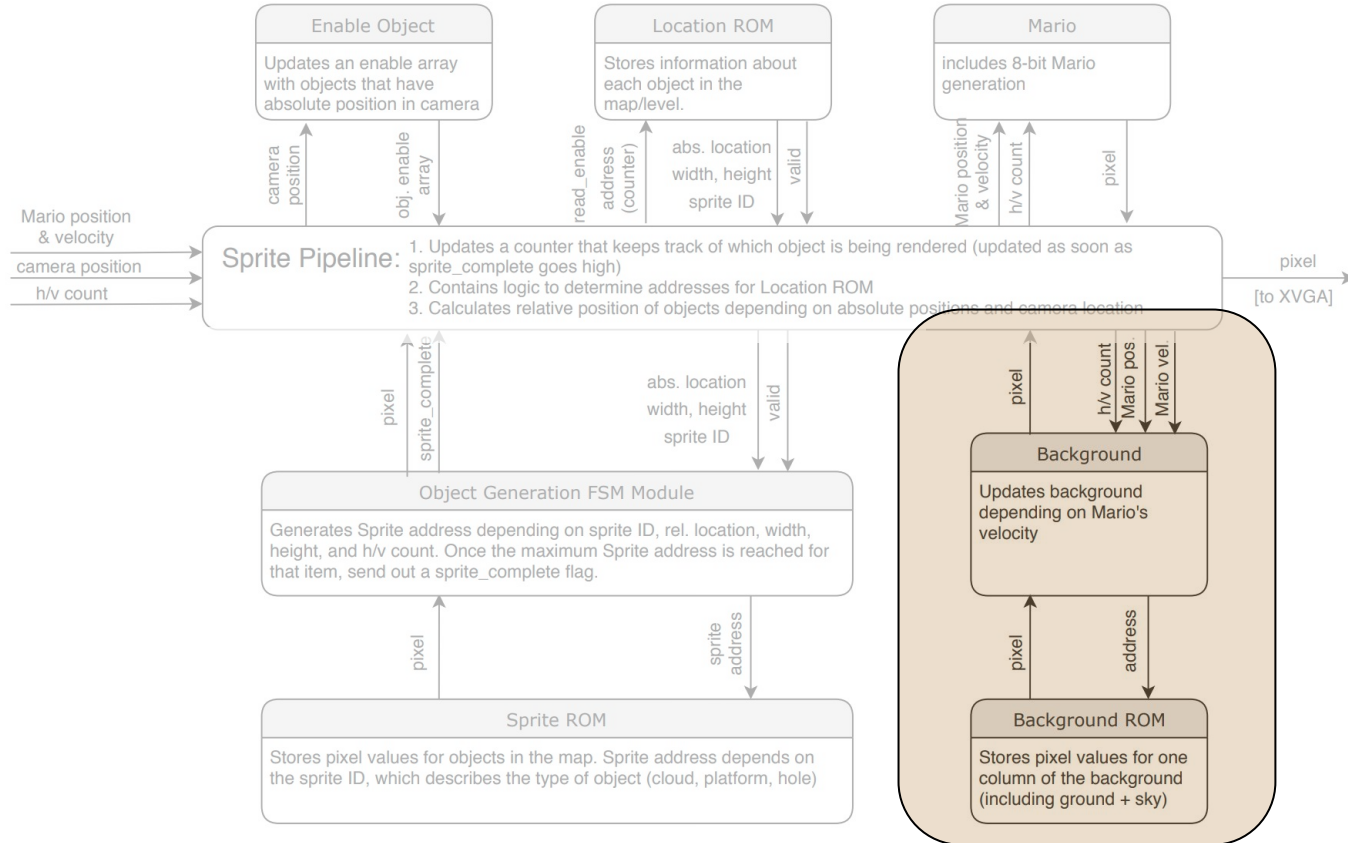
# Saving Sprites as Logic and in Memory

- 8-bit sprite drawing modules
  - Mario
  - Goomba
  - Floor gaps
- COE files
  - Platforms
  - Pipes
  - Clouds and bushes
  - Gameboy background interface



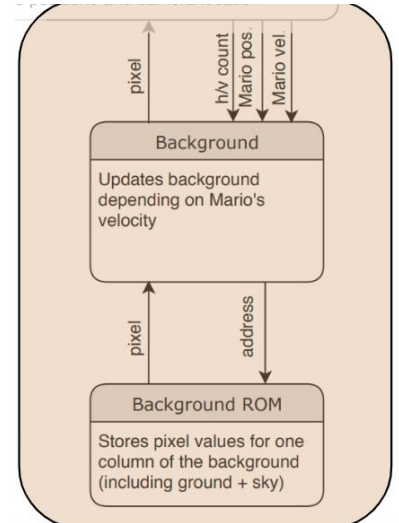
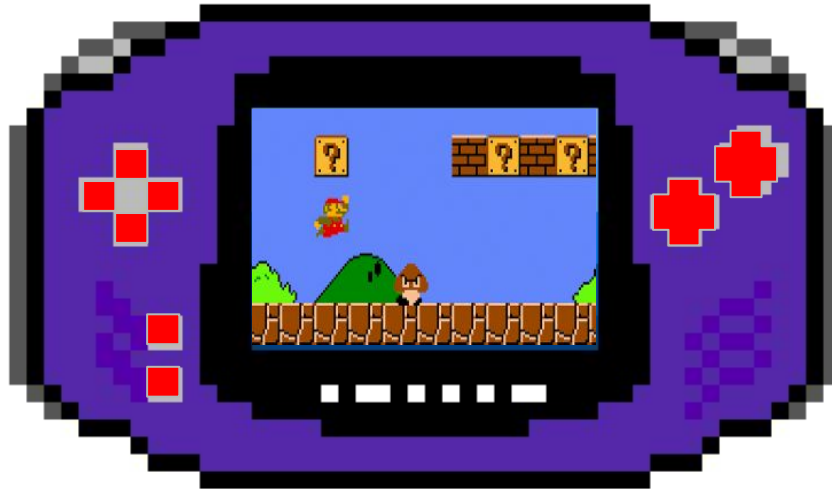


# Background Generation Modules



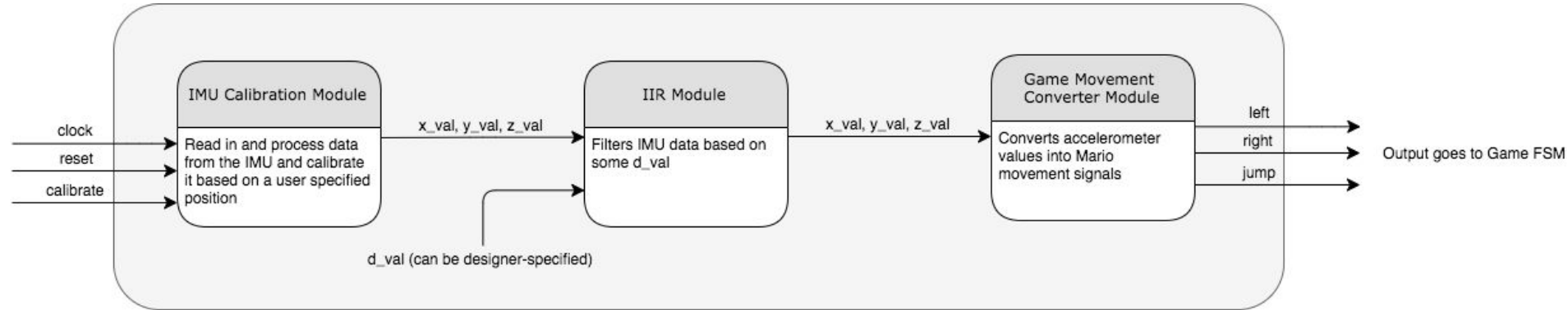
# Background Generation

- Display objects that are always always on screen
  - Sky and floor in game environment background
  - Gameboy interface from COE
    - Saves memory by making gameplay area smaller
    - Enhances user experience- buttons on interface will be overlaid with blob sprites that change color to appear “pressed”



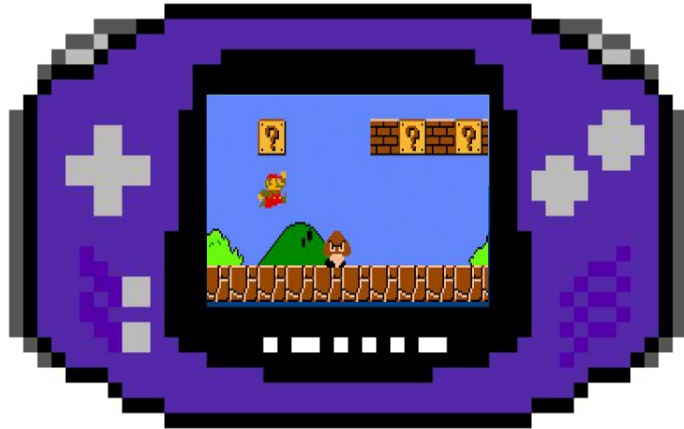
# IMU Controller

- Similar to implementation used in lab 5b- interface with Teensy
- Tilting in the x direction will determine if Mario moves forward or backward
- Tilting in the y direction determines if Mario jumps



# Goals

| Commitment  | Baseline   | Stretch   |
|---|--|---|
| Working game with button inputs<br>One direction of movement and side scrolling and jumping<br>End game flag movement | Working game with IMU controller<br>Two-directional movement<br>Add a Star Coin<br>SD Card Audio | Pick out of the following:<br>IMU Speed<br>Player Avatars<br>Two-Player Mode<br>Wireless IMU Controller |



# Timeline

|              | <b>Week 1</b>  | <b>Week 2</b>   | <b>Week 3</b>   | <b>Week 4</b>                              |
|--------------|--|---|---|--|
| <b>Nancy</b> | Create Game FSM and collision detection modules.                           | Create movement module  | Integrate modules and start working on add-ons        | Add-ons and integration                    |
| <b>Jose</b>  | Create Enable Object Modules, Location ROM and Sprite Pipeline             | Create Object Generation Module and integrate with Sprite Pipeline and Sprite ROM | Integrate Graphics block with remaining system blocks | Add-ons and system-level debugging         |
| <b>Izzy</b>  | Generate COE files and bit art for game sprites and environment background | Implement IMU controller and Gameboy User Interface (GUI ;)                       | Integrate sprites with game map and memory module     | Integration and debugging for full project |

