



FPGA Fruit Ninja

Nadia Salahuddin & Lydia Sun

What is Fruit Ninja?

- ❖ The objective is to slash through fruit with an on-screen cursor
- ❖ Points system as the Skinner box mechanism

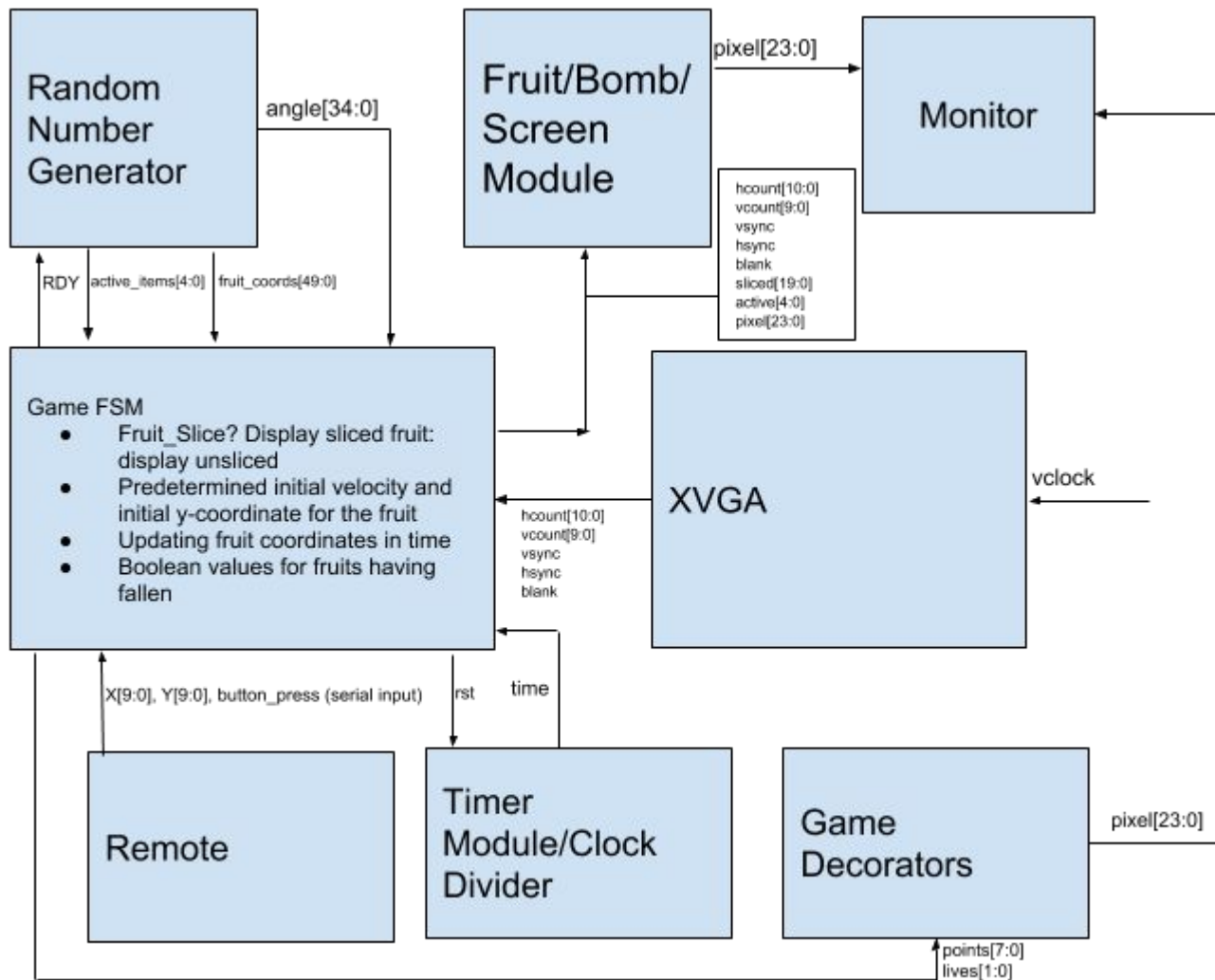


Our implementation

- Finger swipe → Wii-like remote
- Device screen → monitor via VGA
- Software → hardware!

Things to consider:

- Launching fruit
- Fruit mechanics
- Detecting slashes

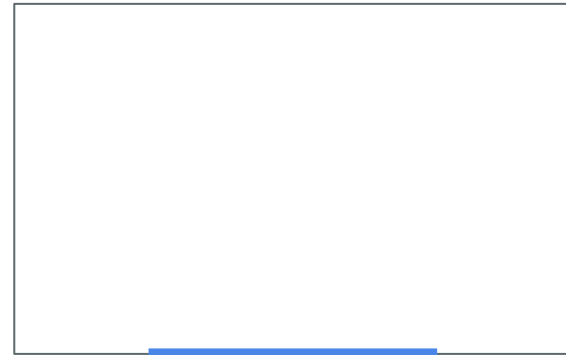
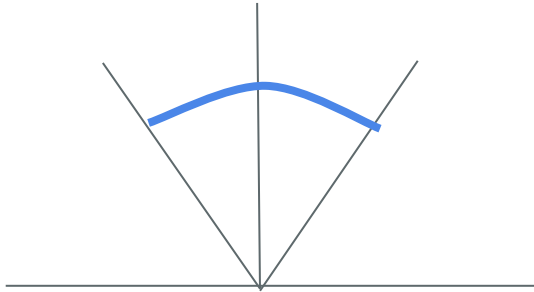


Main Modules

Module	Function
Remote	Generate position of a cursor
Random number generator	Provide random numbers for fruit launching
Fruits & bombs	Draw fruits and bombs depending on slice and active states
Game FSM	Keep track of slicing and score

Random number generator

Function	Generates random launch angle and launch x coordinate Determines which fruit are active at any time
Details	Captures specific # of bits from a CRC output

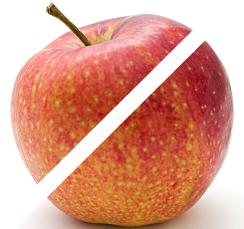
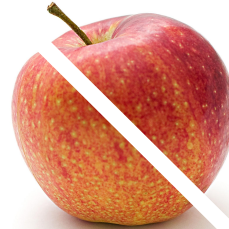
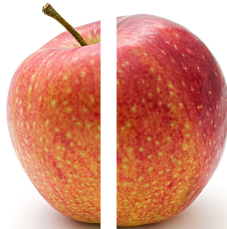


Remote

Function	Generate x and y coordinates for a game cursor
Implementation	IMU and button connected to ESP32 microcontroller
Details	<ul style="list-style-type: none">● Integrate to calculate x and y position from velocity● Transfers data to labkit via serial: 10 bits for x coordinate, 10 bits for y coordinate, 1 bit for button press
Challenges & considerations	<ul style="list-style-type: none">● Adjusting for drift and filtering out noise● Factoring in the angle the remote is held at (orientation)

Fruits & bombs

- Similar to the blob module from lab 3
- Considers two attributes of the item to be drawn: slice and active

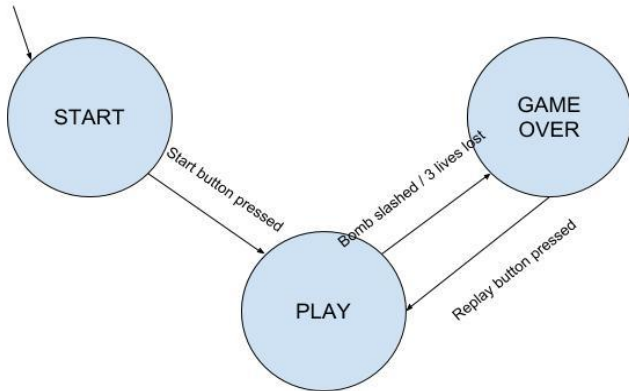


Game Finite State Machine

1. Start -- press button to enter play
2. Play -- Keep playing the game until you lose three lives by letting fruit fall to the bottom of the screen or you slash a bomb. FSM calculates fruit coordinates and slice position, then sends to fruit module.
3. Game over -- allows you to return to “Play” again

Calculating fruit motion:

- $v_x = v_i \cdot \cos\theta$
- $v_y = v_i \cdot \sin\theta$
- $x = v_x \cdot t \cdot \cos\theta$
- $y = v_y \cdot t \cdot \sin\theta - 0.5gt^2$



Goals

Base Goals:

1. Random fruit generation -- more or less unpredictable conception of fruit on screen
2. Basic functionality -- fruit can be cut without hassle
3. Game physics -- the fruit and bombs obey semi-realistic physics

Stretch Goals:

1. Slicing -- fruit halves separate and fall on their own trajectories
2. Combos -- the ability to accumulate more points if the user slashes multiple fruit at once
3. Wireless connection -- using Bluetooth for the remote as opposed to a wired connection

Timeline

November 5: Interfacing ESP and IMU/remote module

November 12: VGA Output, random number generation, fruit/bomb generation

November 19: Game FSM, integration of modules into game FSM

November 26 (Thanksgiving): Debugging/further integration, testing main FSM with buttons

December 3: Integrating remote module, slice mechanics

December 10: Debugging and stretch goals

Questions?

