# Need for Speed: Hacker's Trail

Fan Yang

Rich Chan

Calvin Chung

# Outline

- Objectives
- Overview
- Modules
- Testing
- Division of Labor
- Timeline
- Possible Extensions

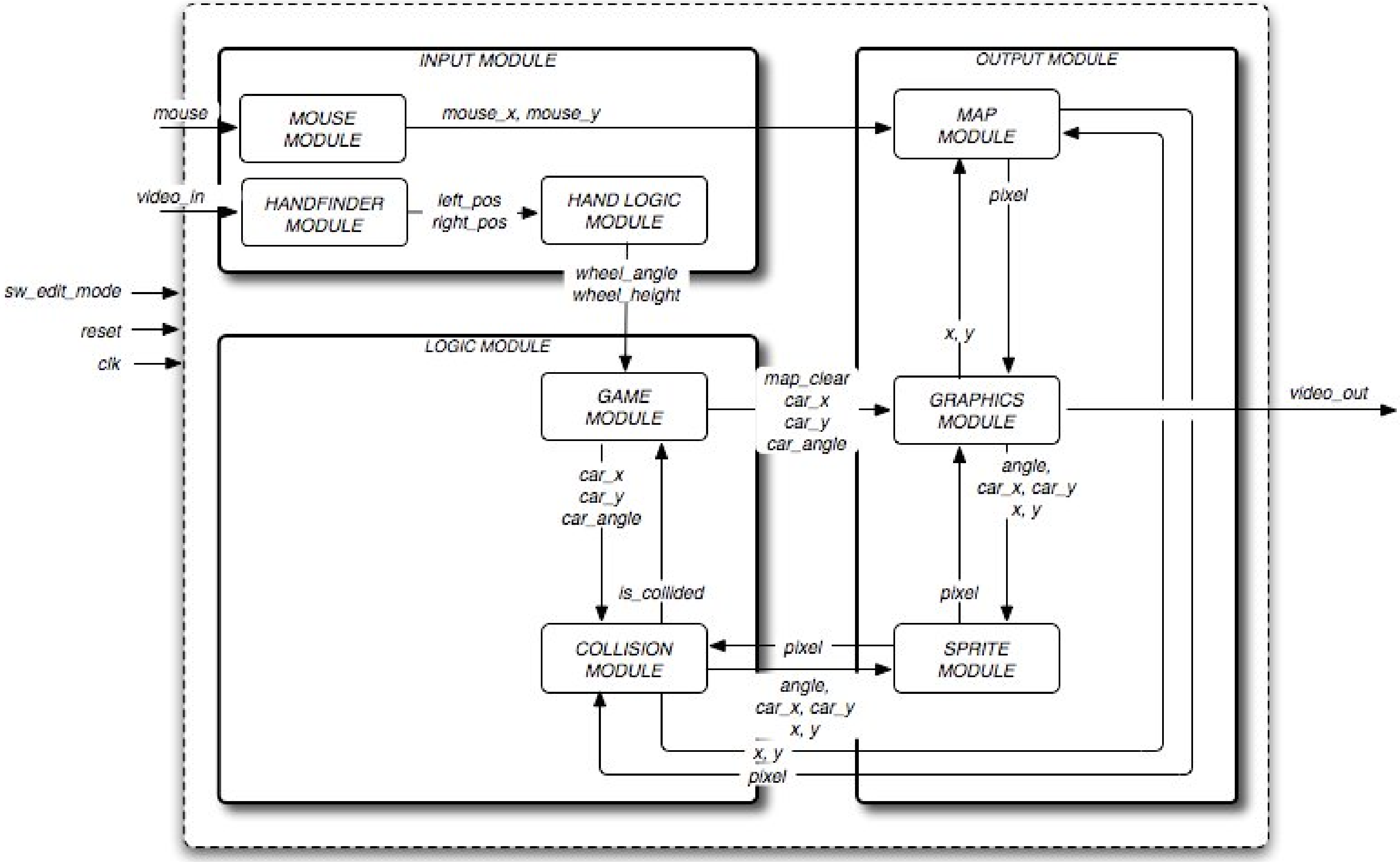
# Objectives

- Racing game with unique features:
  - Users can create the racing track.
  - Control the car with an “air wheel”.

# Overview

- 3 Main Components:
  - Input:
    - Interprets camera and mouse signals
  - Logic
    - Changes state depending on Input
  - Output
    - Displays game state

# Block Diagram



# Mouse module

- The mouse module interfaces to a mouse connected to the labkit’s PS/2 port.
- This allows users to create their own racing tracks by moving the cursor of the mouse.
- At every positive edge of the clock the module receives data input from the mouse and provides the map module with the x-y coordinates of the mouse cursor.

# Hand-finder & Hand-logic Modules

- The user controls the direction and the acceleration of his car by moving an “air wheel”.
- To detect the motion of the wheel, the user will wear colored gloves and steer the wheel in front of a camera.
- Hand locations are determined by center-of-mass algorithm.
  - Height of wheel → acceleration of the car
  - Angle of the wheel → rotation of the car

# Game Module

- Interprets user inputs sent from the input module (accelerate/decelerate, turn, etc.)
- Continuously computes new vehicle states from previous values (velocity, position).
- Keeps track of game state (laps, times, etc.)

# Collision Module

- Given the car position and angle, check for whether the car has collided with the track.
    - For each point that the sprite will occupy, check if both the corresponding pixel on the map is filled.

# Map module

- Stores map using 512k*36 BRAM
- Edit Mode: Loads empty map on reset. Creates a circular blob for each mouse position sent in from the mouse module. Real-time map display through graphics module.
- Play Mode: Displays map through graphics module. Sends in color of particular pixel to collision module per request.
- Need dual-port to read and write at the same time.

# Sprite Module

- Stores car sprite in module.
- Interacts with collision module (Play Mode): Rotates sprite based on car rotation angle from collision module to check if particular coordinate given by collision module is within car boundaries.
- Interacts with graphics module (Play Mode): creates sprite based on car position and rotation angle.

# Graphics Module

- Produces video output
- Edit Mode: Displays map real-time as track is being drawn
- Play Mode: Displays map from map module. Receives car position and car rotation angle from game module to create car sprite through sprite module. Displays car sprite on map.

# Testing

- Modules are tested individually before the system is tested as a whole
- We can simulate the input for each module and observe their outputs

# Division of Labor

**Initial Division**

- Input modules : Fan Yang
- Control modules : Rich Chan
- Output modules : Calvin Chung

- After the prototype is done and each part is tested, we'll start working together in making the game better as a whole.

# Timeline

- 11/28. Basic working prototype.
- 12/02. Refine prototype.
- 12/05. Extensions added.
- 12/07. Project wrap-up.
- 12/10. Finish project report.

# Possible Extensions

- Adding multiplayer support (1 v 1 games)
- Adding interesting behaviors when the car crashes into the track boundaries.
- Adding audio effects