Team A9

Ranganath Selagamsetty and Sourav Panda and David Yang

Add your 12 slides after this slide… [remember, 12 min talk + 3 min Q/A]

For more information about formatting or importing slides see:
https://gsuite.google.com/learning-center/products/slides/get-started/

Make sure to cover:
- Use Case
- Requirements
- Solution Approach
- Testing, Verification and Metrics
- Tasks and Division of Labor
- Schedule

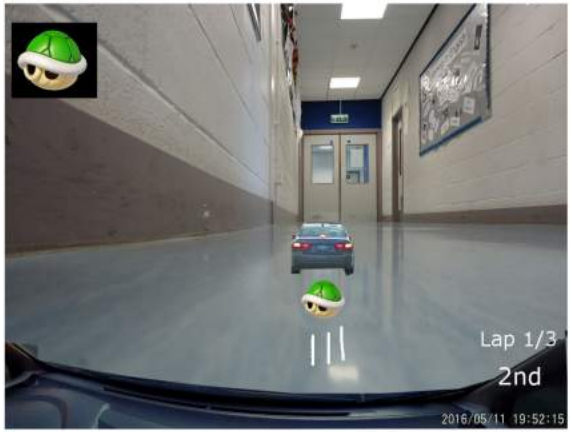
Use Case

- Proposal: Make an augmented reality racing game with networked remote controlled vehicles
- Inspired by Mario Kart, with a slalom-style, user specified race track and various virtual items and power-ups to augment the physical car race
- Both racing games and streaming video from mobile drones exist, but making video streams from vehicular toys interact with the devices environment is a challenging aspect of racing games that isn't well tackled.
- This project aims to reduce the disconnect from playing a virtual racing game to physically driving a remote controlled car.

Requirements: Game Software and Game Devices

Classic racing game with track defined by placeable slalom-style gates.

Description	Requirement
Power up interactions	Power ups that can inhibit, enhance and defend players' cars
Overlay game HUD	Streamed first person camera feed from car to create AR experience

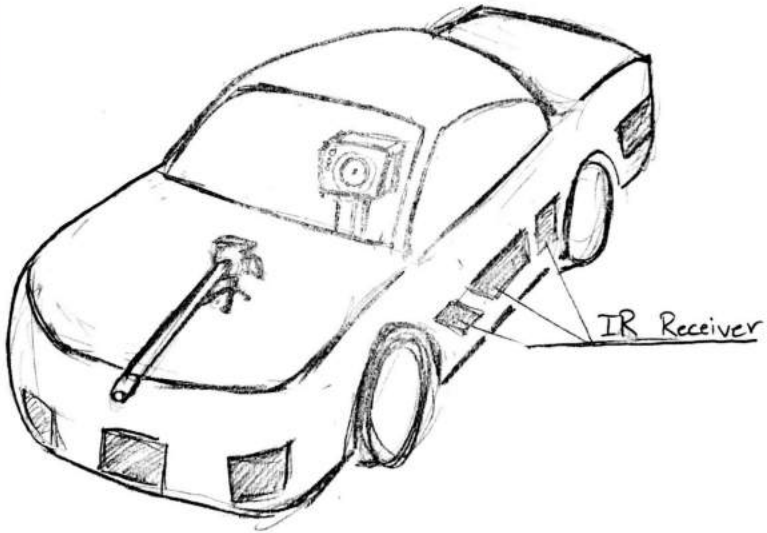


Requirements: Car

Description	Requirement
A four wheeled vehicle whose forward and reverse speeds and turn angles can be set programmatically	Turn angles should be accurate within a tolerance of ± 1º ± 2 RPM Speed should be accurate within
An onboard sensor array to detect item collisions with other vehicles and passing over gates	Time of flight from transmission to reception should be no more than 100 ms.
A front facing camera to stream video	Latency from capture to display should be no more than 100 ms

Requirements: Car and Gate

Description	Requirement
Onboard computer capable of coordinating I/O and networking	Must be able to control and direct power to motors Read and transmit sensor values Stream camera feed to a player's computer
Gates that define the course of a track	Can be driven over and can be detected by the car Are fixed in place and immobile



Key Technical Challenges

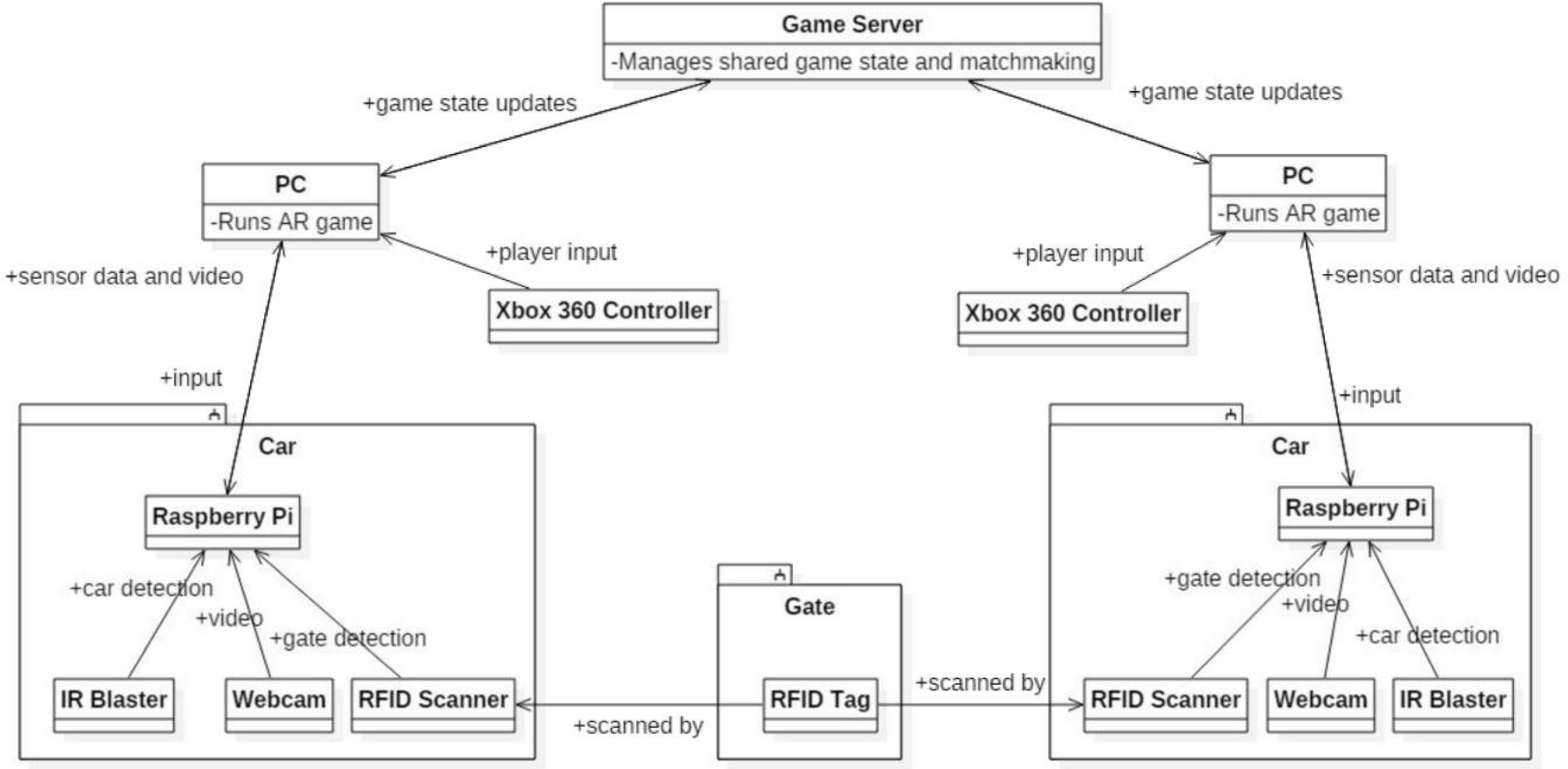
1. Sending and receiving sensor and video data, transmitting motor values of magnitudes and directions (forward, reverse, right, left) from computers to cars.

2. Making the game robust to user input such that the game only stops when specified by the users.

3. Designing a PCB/circuit to handle the power draw of the motors and route sensor data.

Solution Approach

1. Game software
    a. Develop with Unity game engine with Vuforia for AR on
    b. Controller - Xbox 360 controller
2. Cars
    a. Fabricate chassis to meet requirements for space and size
    b. Buy motors with encoders for speed control. Motor encoders will be able to read and set speed of wheels.
    c. IR blaster/receiver to detect which cars fired and were hit by items.
    d. RFID scanner on car to detect passing gates
    e. At least 1080p and 30 FPS webcam with video capabilities
    f. Raspberry pi GPIO for driving motors and reading sensors wifi connection and TCP sockets for sending video images and sensor data


Game Server
-Manages shared game state and matchmaking
+game state updates
+game state updates
PC
-Runs AR game
PC
-Runs AR game
+player input
+player input
+sensor data and video
+sensor data and video
Xbox 360 Controller
Xbox 360 Controller
+input
+input
Car
Car
Raspberry Pi
Raspberry Pi
+car detection
+video
+gate detection
+gate detection
+video
+car detection
Gate
IR Blaster
Webcam
RFID Scanner
RFID Tag
+scanned by
+scanned by
RFID Scanner
Webcam
IR Blaster

Testing, Verification, and Metrics

1. Game
    a. Game Rules:
        i. Manually test racing software by carrying RFID tag from gate to gate
    b. Power - Ups:
        i. Inhibitors: Whether IR receiver can acknowledge a pulse sent at it, and can identify which transmitter sent the pulse, measuring time of flight on the pi on the car.
        ii. Enhancements will test the motors and encoders speed control by setting the voltage on the motors and reading the output speed from the encoders
        iii. Overlay game HUD on streamed first person camera feed from car to create AR experience will be manually inspected

Testing and Verification

1. Drive the car and complete the following tasks:
    a. Use all Power-up types
    b. Drive: will be testing by setting the speed of the car wheels to match each other, and driving the car 10m, and measuring the deviance from center. Test will be repeated for different speed settings. Test will be repeated with turning
    c. Reverse: will be tested similarly to driving straight with the single caveat that it will be going reverse instead
    d. Test video streaming and sensor data sending by stress testing with large amounts of data and averaging time to send from pi and receive response from computer

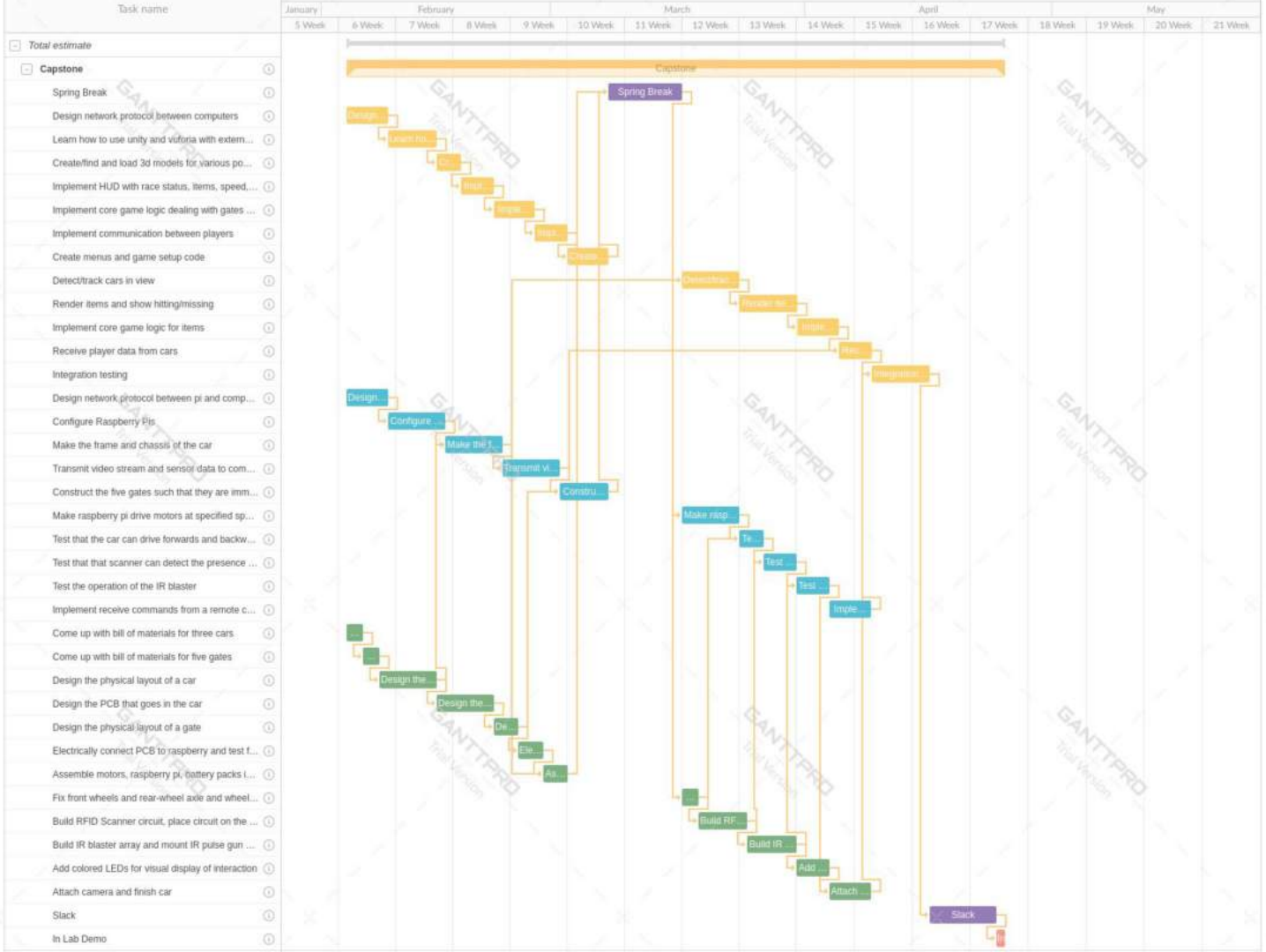
Tasks and Division of Labor

Game Tasks

- Implement core game logic dealing with gates and race: tracking place, progress, time
- Detect/track cars in view
- Render items and show hitting/missing

Car Tasks

- Design the PCB that goes in the car
- Make raspberry pi drive motors at specified speed, as well as read the speed of the motors for an encoder
- Test that that scanner can detect the presence of a RFID tag both stationary, and while the car is moving
- Build and test IR blaster array and mount IR pulse gun on car
- Transmit video stream and sensor data to computer


Task name
January
February
March
April
May
5 Week
6 Week
7 Week
8 Week
9 Week
10 Week
11 Week
12 Week
13 Week
14 Week
15 Week
16 Week
17 Week
18 Week
19 Week
20 Week
21 Week
Total estimate
Capstone
Spring Break
Design network protocol between computers
Learn how to use unity and vuforia with extern...
Create/find and load 3d models for various po...
Implement HUD with race status, items, speed,...
Implement core game logic dealing with gates ...
Implement communication between players
Create menus and game setup code
Detect/track cars in view
Render items and show hitting/missing
Implement core game logic for items
Receive player data from cars
Integration testing
Design network protocol between pi and comp...
Configure Raspberry Pis
Make the frame and chassis of the car
Transmit video stream and sensor data to com...
Construct the five gates such that they are imm...
Make raspberry pi drive motors at specified sp...
Test that the car can drive forwards and backw...
Test that that scanner can detect the presence ...
Test the operation of the IR blaster
Implement receive commands from a remote c...
Come up with bill of materials for three cars
Come up with bill of materials for five gates
Design the physical layout of a car
Design the PCB that goes in the car
Design the physical layout of a gate
Electrically connect PCB to raspberry and test f...
Assemble motors, raspberry pi, battery packs i...
Fix front wheels and rear-wheel axle and wheel...
Build RFID Scanner circuit, place circuit on the ...
Build IR blaster array and mount IR pulse gun ...
Add colored LEDs for visual display of interaction
Attach camera and finish car
Slack
In Lab Demo