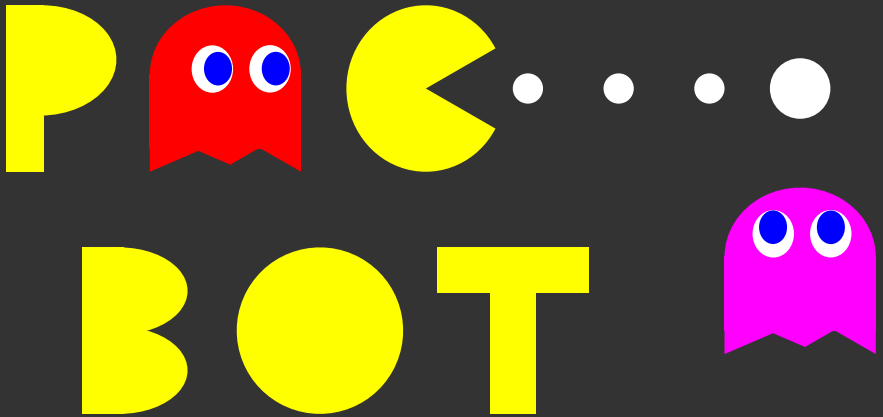
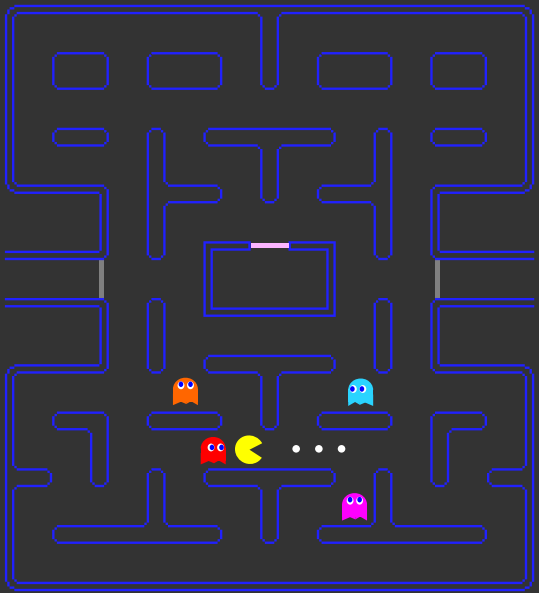
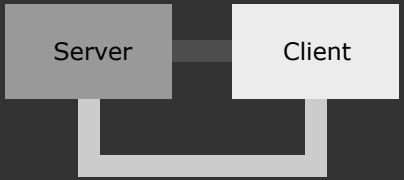
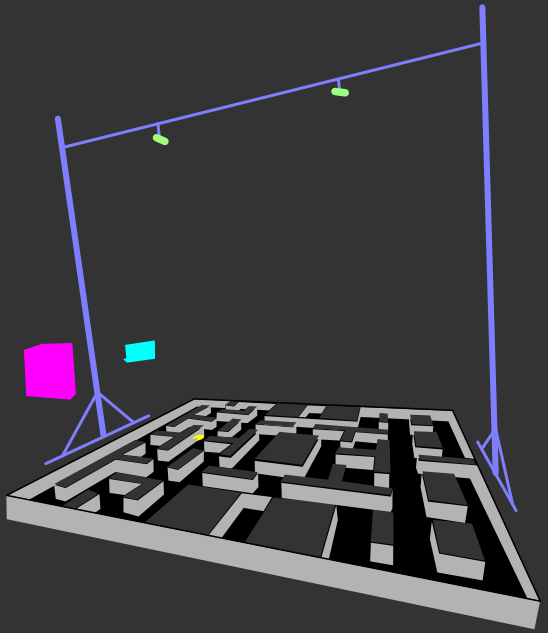
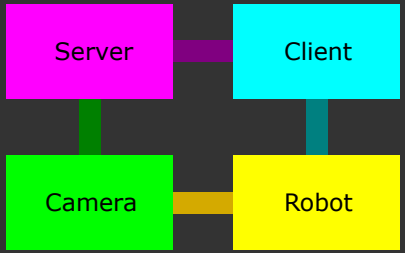
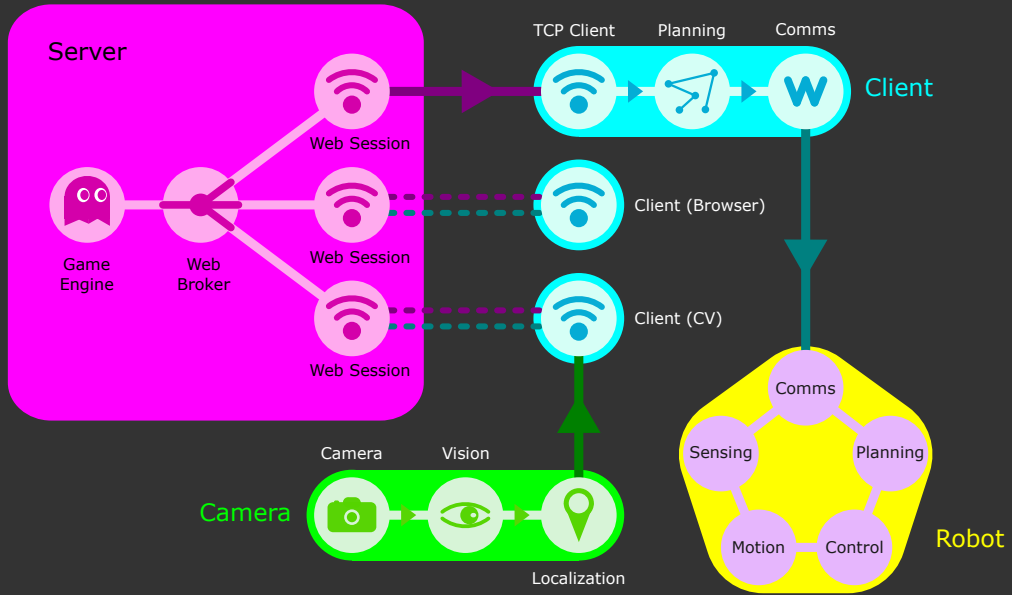

PAC
BOT




Server
Client

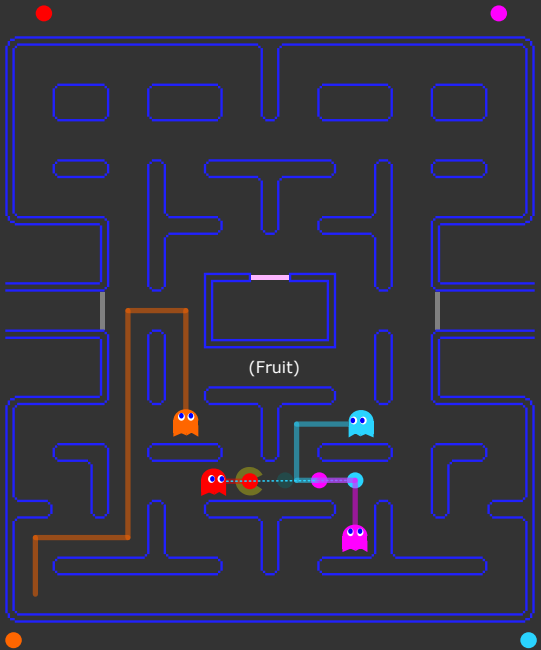



Server
Client
Camera
Robot


Server
Web Session
Game Engine
Web Broker
Web Session
Web Session
TCP Client
Planning
Comms
Client
Client (Browser)
Client (CV)
Camera
Vision
Camera
Localization
Comms
Sensing
Planning
Motion
Control
Robot

Game Engine

Ghosts:
Personalities
Scatter vs Chase



Scoring:
Fruit (+100)
Super Pellets (+50)
Pellets (+10)
Ghost Combo
(+200, +400, +800, +1600)

Max Score:
Fruit (200)
Super Pellets (200)
Pellets (2400)
Ghost Combo (12000)

14800 per level

Network Format

Game State:

Current Ticks (2 bytes)
Update Period (1 byte)
Game Mode (1 byte)
Mode Steps (1 byte)
Mode Duration (1 byte)
Level Steps (2 bytes)
Current Score (2 bytes)
Current Level (1 byte)
Current Lives (1 byte)
Ghost Combo (1 byte)
Ghosts (4 x 4 = 16 bytes)
Pacman (2 bytes)
Fruit (4 bytes)
Pellets (124 bytes)
---
159 bytes total

Helpful Reference for Bitwise Operations:

>> means right-shift, divide by power of 2
(for example, 1234 >> 5 = 1234 // 2^5 = 38)

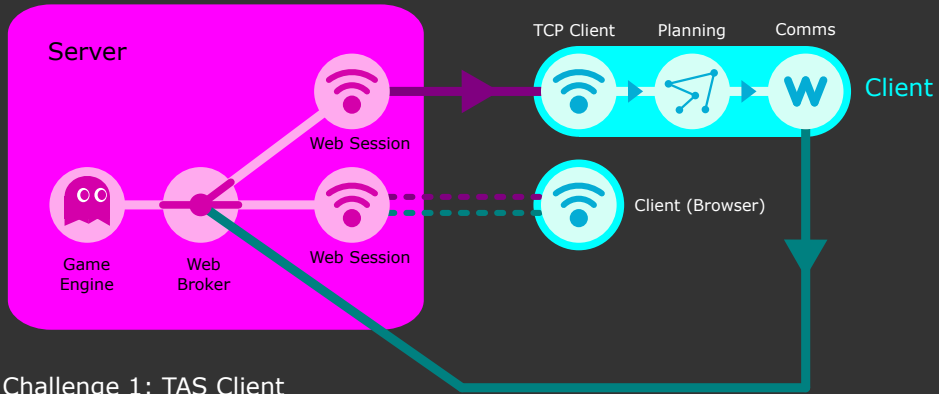
<< means left-shift, multiply by power of 2
(for example, 12 << 3 = 12 * 2^3 = 96)

~ means bitwise NOT (flip all the bits)
(for example, ~12 [8-bit] = ~ 0b00001100 = 0b11110011 = 115)

| means bitwise OR (keep bits where either is 1)
(for example, 12 | 5 = 0b1100 | 0b0101 = 0b1101 = 13)

& means bitwise AND (clear bits where either is 0)
(for example, 12 & 5 = 0b1100 & 0b0101 = 0b0100 = 4)

^ means bitwise XOR (if the bits add, would it be odd or even?)
(for example, 12 ^ 5 = 0b1100 ^ 0b0101 = 0b1001 = 9)



Challenge 1: TAS Client

Play the game with the help of an automated algorithm you write! The highest score gets bragging rights!

Software Guidelines:

Only one autonomous run script allowed
(can't open multiple client terminals)
Cite code taken from online
Document changes clearly in GitHub
No changes made to the server code

Competition Rules:

No interference
No overloading the server
No unreasonable controls
(you can't just teleport to the other side of the screen!)
No manipulation of the game state other than motion/location