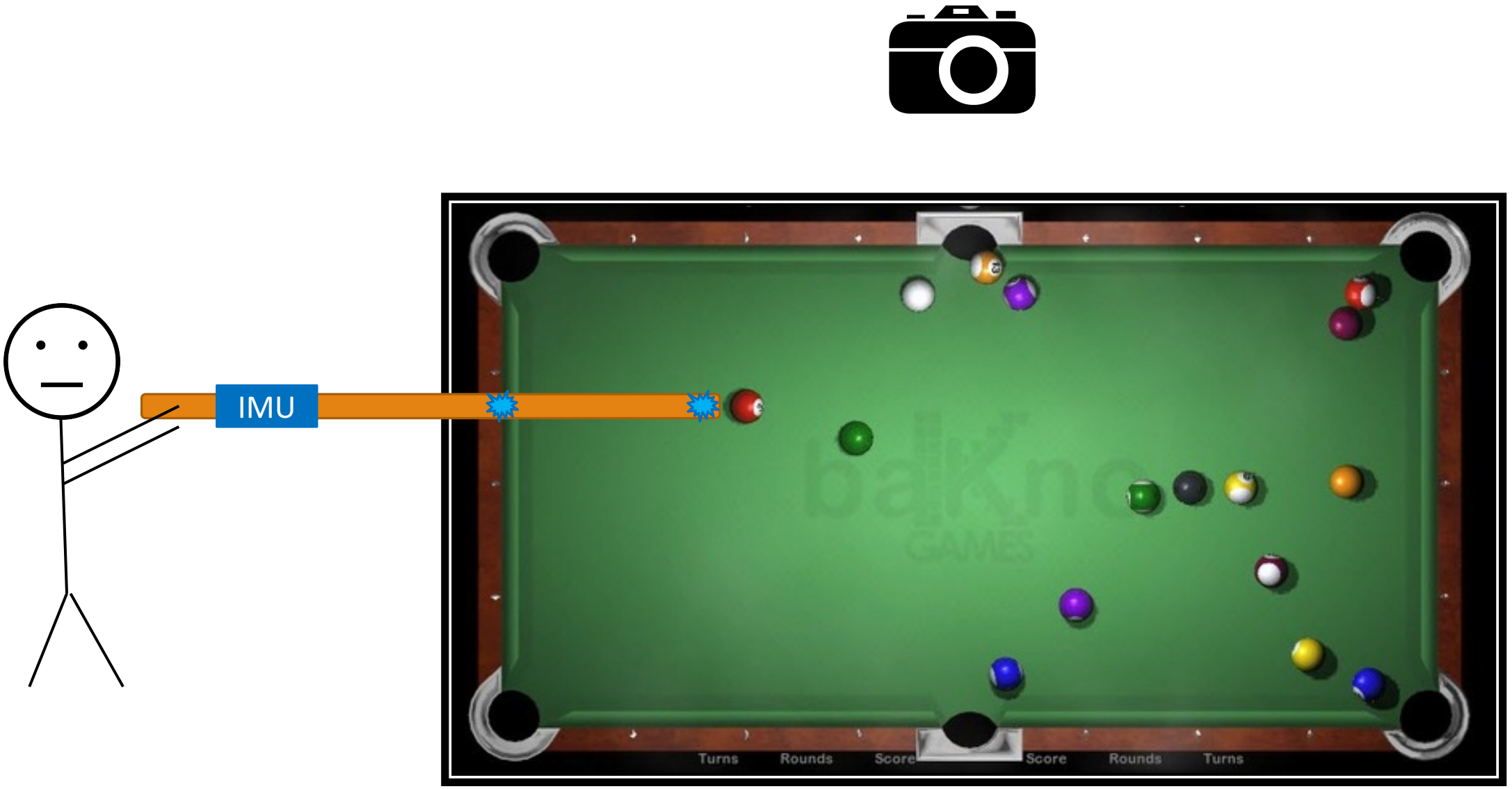
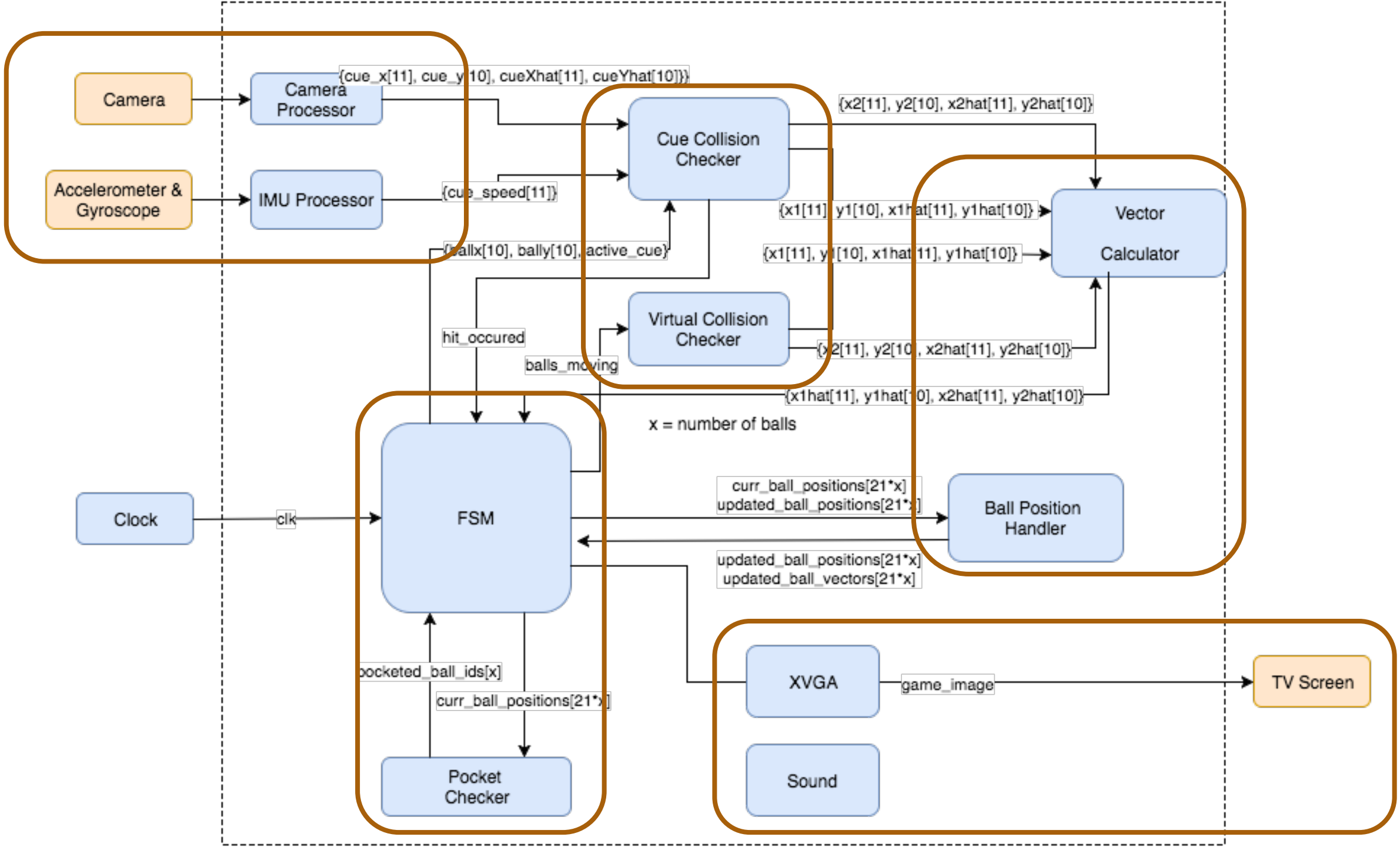
JOOLA
INSIDE

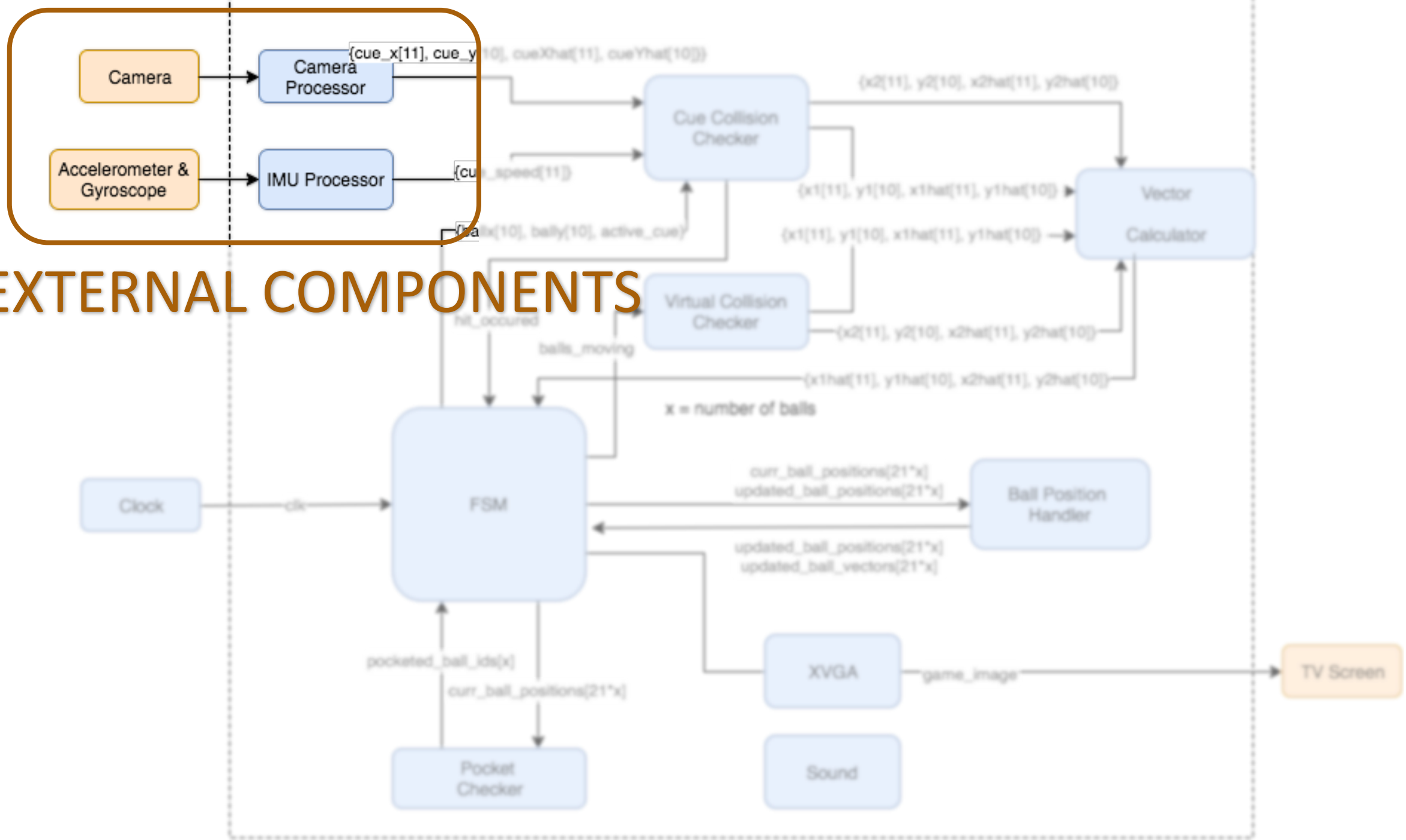
LA PC-NA

SEMI-VIRTUAL POOL

ZAREEN CHOUDHURY AND MATT BASILE


IMU
Turns
Rounds
Score
Score
Rounds
Turns


Camera
Camera Processor
{cue_x[11], cue_y[10], cueXhat[11], cueYhat[10]}
Accelerometer & Gyroscope
IMU Processor
{cue_speed[11]}
Cue Collision Checker
{x2[11], y2[10], x2hat[11], y2hat[10]}
{x1[11], y1[10], x1hat[11], y1hat[10]}
Vector Calculator
{ballx[10], bally[10], active_cue}
{x1[11], y1[10], x1hat[11], y1hat[10]}
Virtual Collision Checker
hit_occured
balls_moving
{x2[11], y2[10], x2hat[11], y2hat[10]}
{x1hat[11], y1hat[10], x2hat[11], y2hat[10]}
x = number of balls
Clock
clk
FSM
curr_ball_positions[21*x]
updated_ball_positions[21*x]
Ball Position Handler
updated_ball_positions[21*x]
updated_ball_vectors[21*x]
pocketed_ball_ids[x]
curr_ball_positions[21*x]
Pocket Checker
XVGA
game_image
TV Screen
Sound

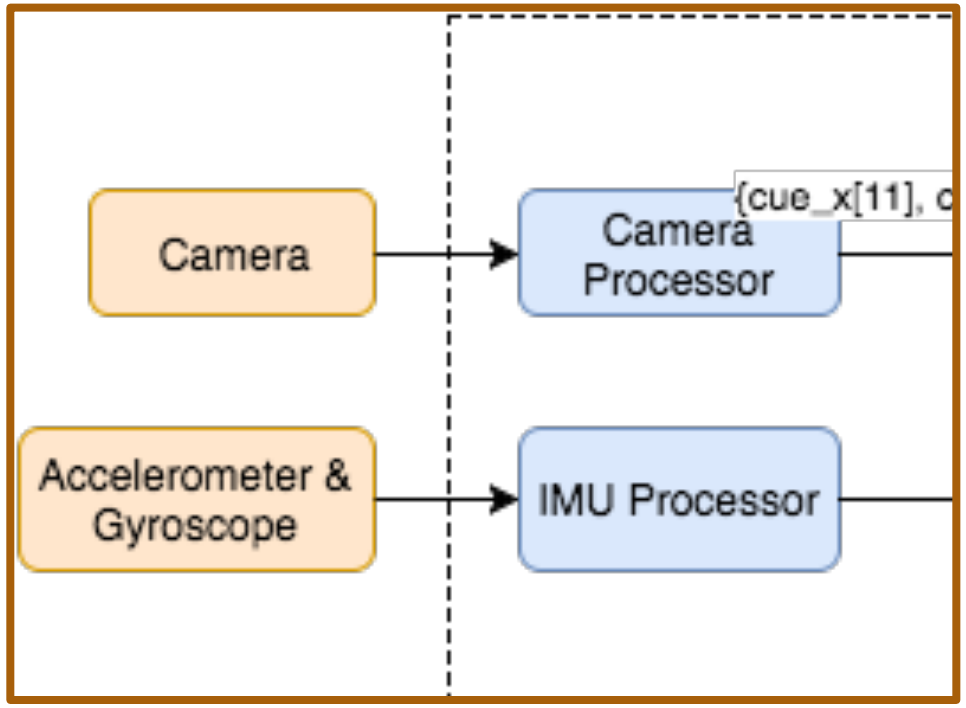


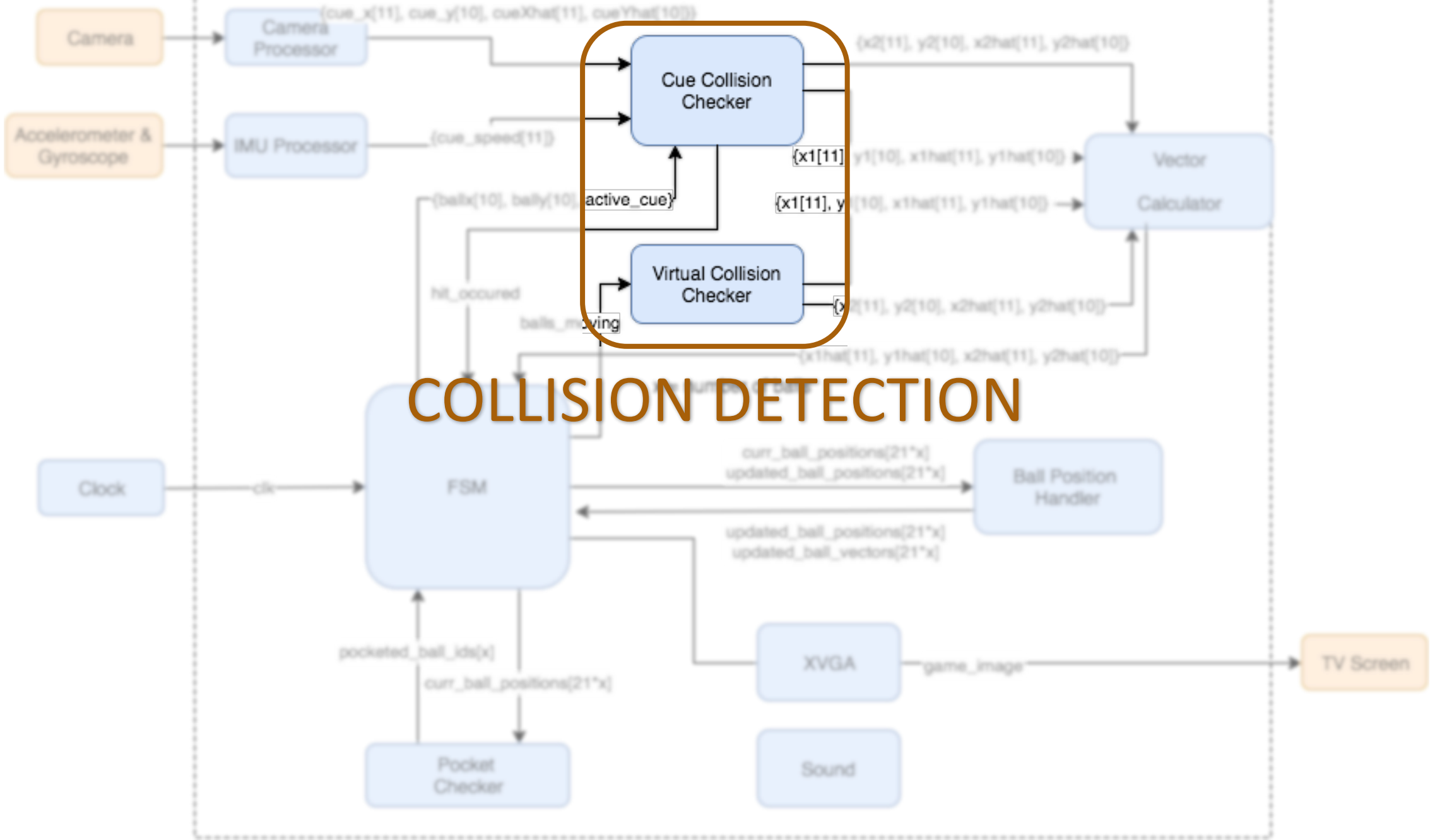
EXTERNAL COMPONENTS

External Components

Camera Processor:
Output (x,y) coordinates of IR LED's on cue

IMU Processor:
Output cue speed



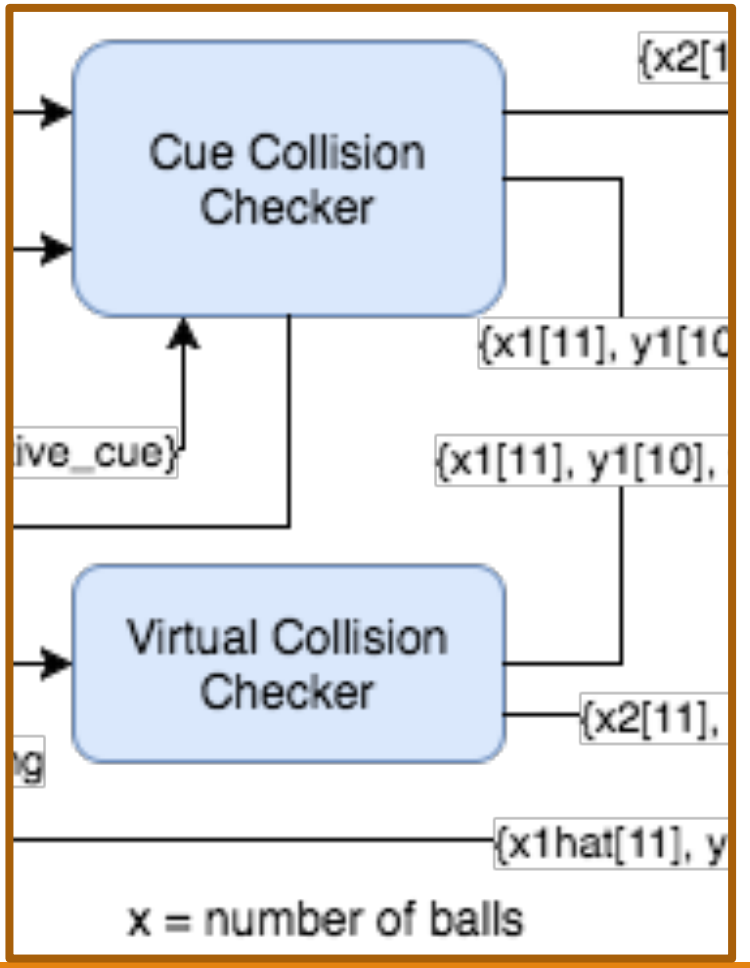

Camera
Camera Processor
{cue_x[11], cue_y[10], cueXhat[11], cueYhat[10]}
Accelerometer & Gyroscope
IMU Processor
{cue_speed[11]}
Cue Collision Checker
{x2[11], y2[10], x2hat[11], y2hat[10]}
{x1[11]
active_cue}
{x1[11], y
Vector Calculator
{balls[10], bally[10]
hit_occured
Virtual Collision Checker
balls_moving
{x1hat[11], y1hat[10], x2hat[11], y2hat[10]}
COLLISION DETECTION
Clock
clk
FSM
curr_ball_positions[21*x]
updated_ball_positions[21*x]
Ball Position Handler
updated_ball_positions[21*x]
updated_ball_vectors[21*x]
pocketed_ball_ids[x]
curr_ball_positions[21*x]
XVGA
game_image
TV Screen
Pocket Checker
Sound

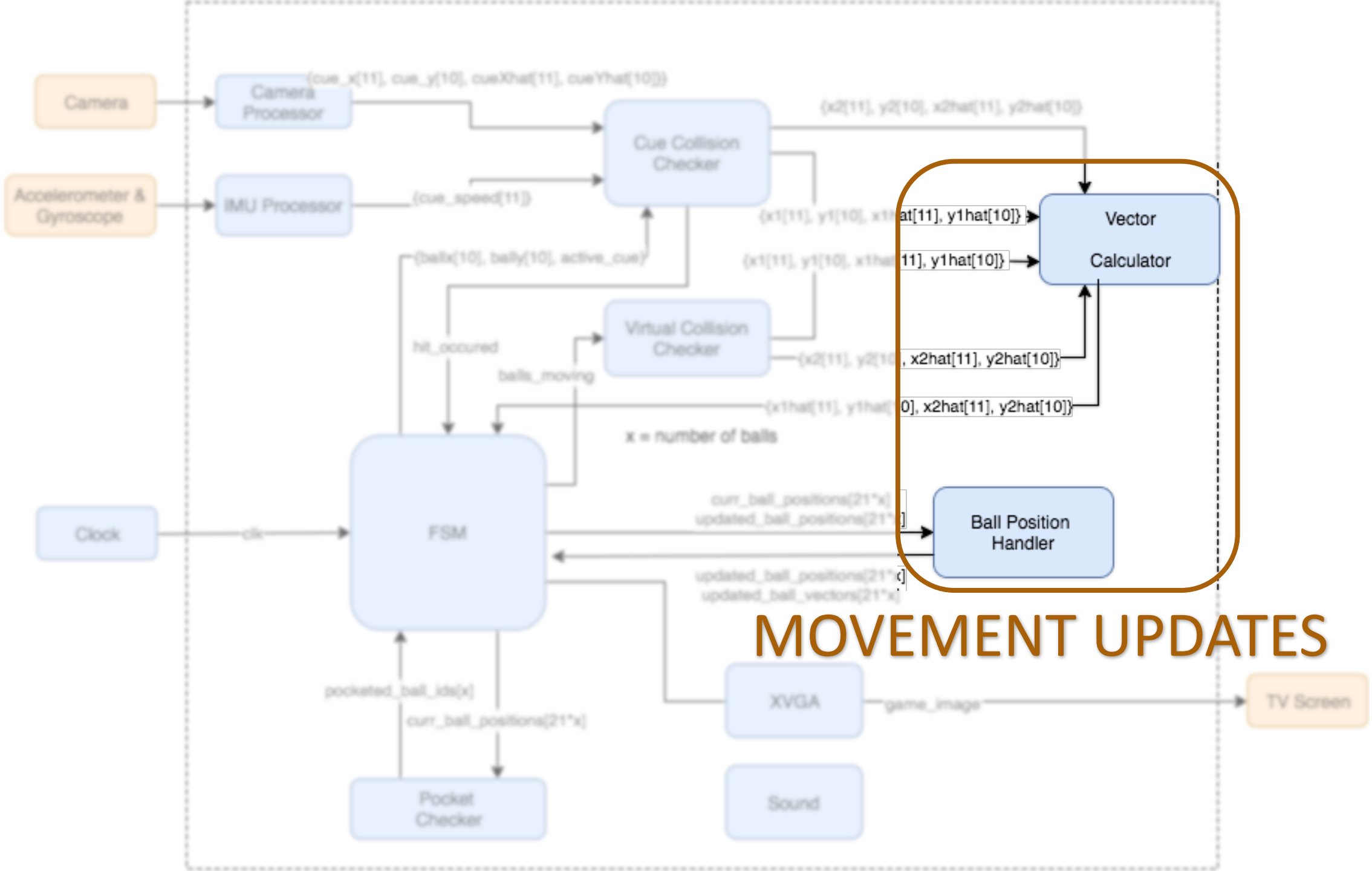
Collision Detection

Calculate distance between ball centers

Model cue tip as “pool ball”

Avoid square roots using known distances





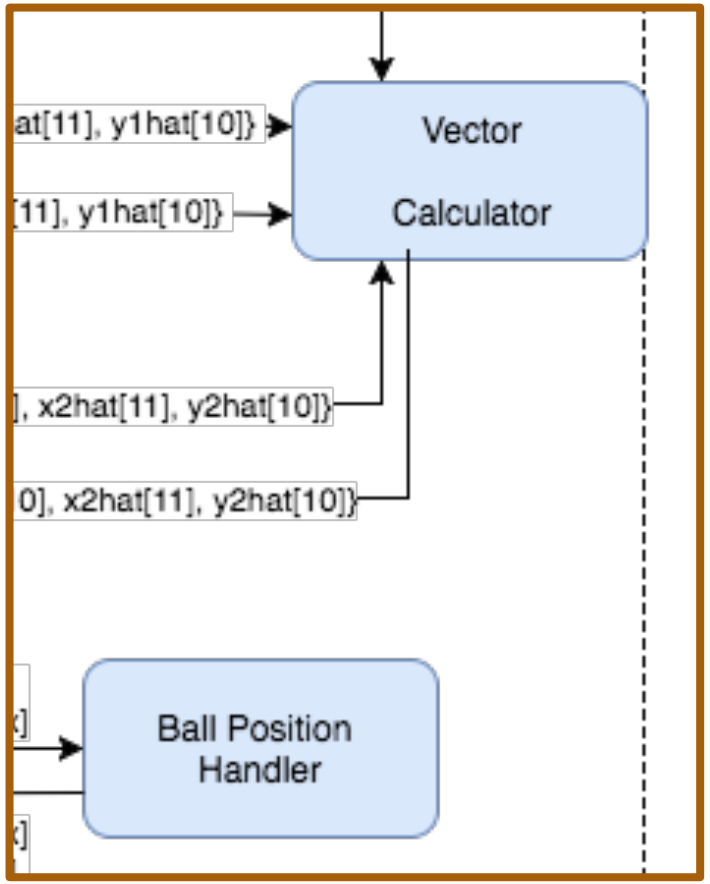
MOVEMENT UPDATES

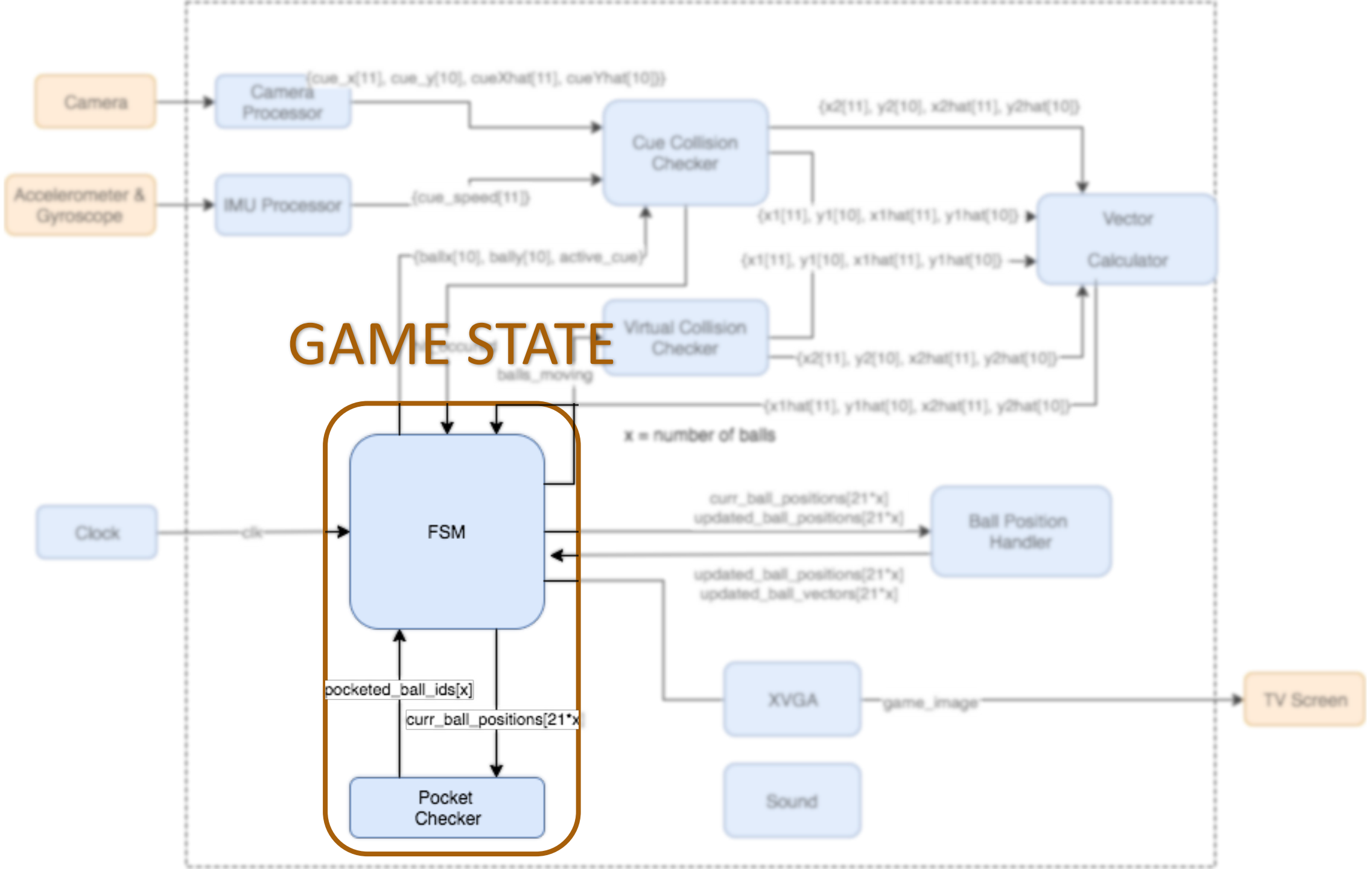
Movement Updates

Every ball has a position and movement vector
- Split into X and Y components

Update every ball's position at each time step
- Then lower speed in each direction: friction

Collisions: find normal and tangential vectors
- Tangential vectors stay the same
- Normal vectors swap



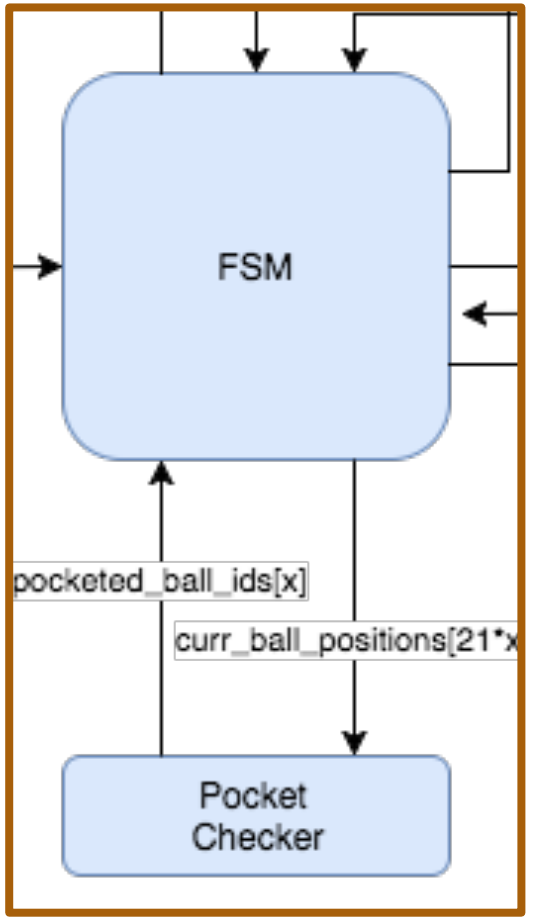

Camera
Camera Processor
{cue_x[11], cue_y[10], cueXhat[11], cueYhat[10]}
Cue Collision Checker
{x2[11], y2[10], x2hat[11], y2hat[10]}
Accelerometer & Gyroscope
IMU Processor
{cue_speed[11]}
{x1[11], y1[10], x1hat[11], y1hat[10]}
Vector Calculator
{ballx[10], bally[10], active_cue}
{x1[11], y1[10], x1hat[11], y1hat[10]}
GAME STATE
Virtual Collision Checker
{x2[11], y2[10], x2hat[11], y2hat[10]}
balls_moving
{x1hat[11], y1hat[10], x2hat[11], y2hat[10]}
x = number of balls
Clock
clk
FSM
curr_ball_positions[21*x]
updated_ball_positions[21*x]
Ball Position Handler
updated_ball_positions[21*x]
updated_ball_vectors[21*x]
pocketed_ball_ids[x]
curr_ball_positions[21*x]
XVGA
game_image
TV Screen
Pocket Checker
Sound

GAME STATE

Keeps track of:

- Player turn
- Balls remaining
- Balls moving
- Ball positions

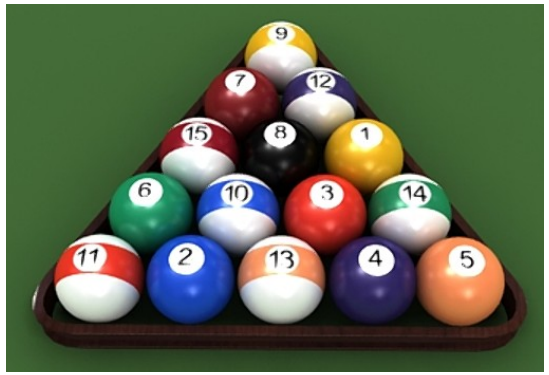
Serves as central coordinator for information for consistency and timing



	10/31	11/7	11/14	11/21	11/28	12/5
Camera + IR Setup	Zareen	Zareen				
IMU Setup		Zareen	Zareen			
Cue Collisions				Zareen	Zareen	
Ball Collisions	Matt	Matt				
Movement Updates		Matt	Matt			
Game FSM			Matt	Matt		
Integration				Both	Both	
Testing/Buffer					Both	Both

Stretch Goals

Increased number of balls



Spinning pool ball

Sound

Haptic feedback on cue-ball collision

Questions?