# TiltGolf: Tilt-Controlled Golf Game on a BeagleBone

Team TiltGolf

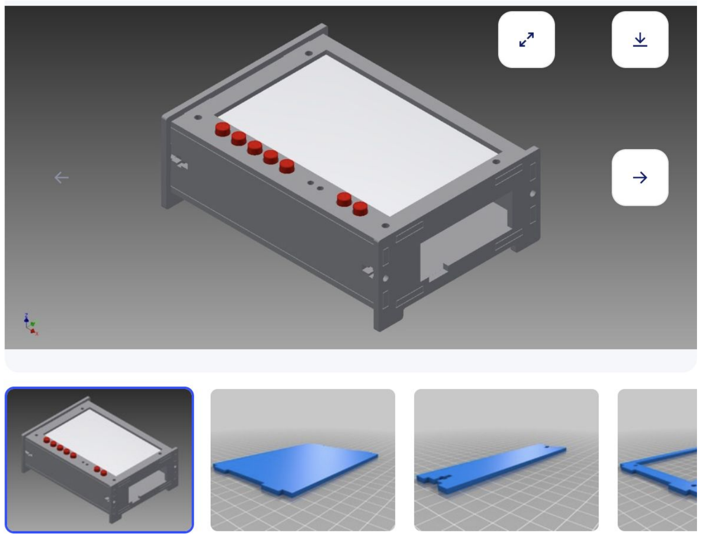
Noah Hathout (nhathout@bu.edu), Bennett Taylor (betaylor@bu.edu), Skanda Nadig(skandan@bu.edu)

## Motivation and Project Idea

- **Problem:** Create a fun, interactive game that uses simulated real-world physics without relying on buttons, joysticks, or keyboard inputs. We also want to make this module as polished as possible with a custom 3D printed enclosure and fun level design.
- **Who Cares?** Students and/or instructors in embedded systems could use the game as a fun demo of kernel modules and divers + real-time graphics and simulated physics. Kids would also have a lot of fun playing a “GameBoy”-like game that is both challenging and fulfilling.
- **Main Project Idea:**
  - Golf game running entirely on Breaglebone + LCD cape; 3D printed handheld enclosure with built-in IMU for tilt sensing; Qt UI + physics engine that make gameplay smooth and realistic.

# 3D Printed Enclosure Inspiration

- Design Goals
  - Handheld & intuitive to use
  - Arcade/GameBoy -like feel

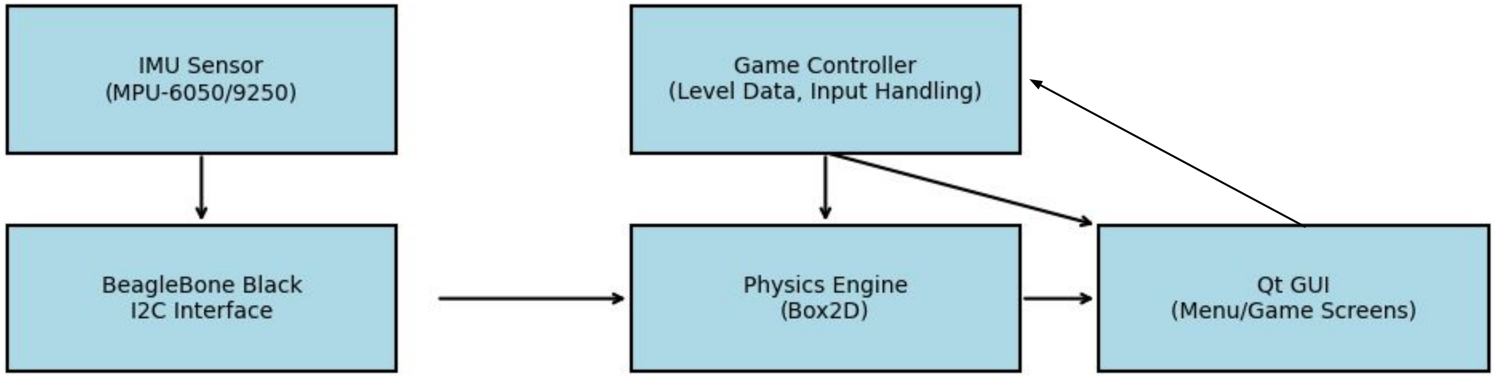


- Inspo1 (left)
  - Ref: https://www.printables.com/model/648016-beaglebone-black-bb-view-lcd-43-case/files
- Inspo2 (above)
  - Ref: https://www.thingiverse.com/thing:550532

## Project Description

- **Summary:**
  TiltGolf is a tilt-controlled golf game that runs on a BeagleBone Black with an attached LCD cape. Also in the enclosure will be an IMU that senses the device’s tilt. This tilt will be filtered in the kernel-space and fed to the Qt-based game. The layer that takes care of the physics engine is based on Box2D with realistic physics and acceleration properties so that the ball rolls, slows down, and bounces off walls realistically as the player tils the device. QT is used to handle menuing, LCD user input, and rendering the Box2D golf simulation.

TiltGolf System Architecture



## System Components

1. Hardware Platform
    a. BeagleBone Black
    b. Sensors: IMU (MPU-6050 / MPU-9250)
    c. Display: Touchscreen connected to BeagleBone
2. Input
    a. IMU tilt for ball control
    b. Touchscreen for menu interaction
3. Communication
    a. IMU communicates via I2C only
4. Display
    a. 4.3" TFT LCD transmissive display
        i. 480(RGB) x 272 resolution
        ii. With LED backlight
        iii. With capacitive touch
    b. Cape interface board with connector expandability
    c. Board ID EEPROM with kit identity storage

## System Components

5. Physics Engine
   a. Box2D used for realistic ball movement and collision detection
   b. Physics engine updates ball state each frame
   c. Integrated into the game binary (statically linked)
6. Rendering
   a. Qt GUI for all game screens and visual updates
   b. Game loop:
      i. Choose level → Start simulation → Read IMU → Update Physics → Render Screen → Repeat → Exit or Restart
7. Output / Audio
   a. Visual output only (Qt GUI)
   b. No audio output

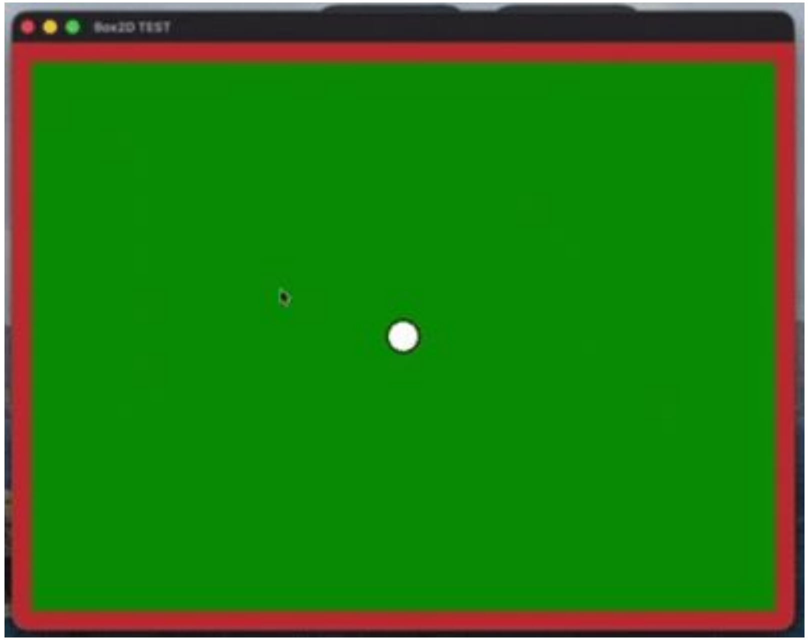
## Milestones, timeline, and breakdown of tasks

- Creation of level data, game loop, and leaderboard
  - Assigned to: Bennett Taylor
  - Complete by: December 3rd
  - Progress: Level start and restart have been completed
  - Leaderboard and separate users is “nice to have”
- Integration of level data and Box2D simulation
  - Assigned to: Noah Hathout
  - Complete by: December 6th
  - Progress: Box2D simulation framework has been established
- Integration of menuing and Box2D simulation rendering
  - Assigned to: Bennett Taylor
  - Complete by: November 31st
  - Progress: Menuing and Box2D rendering have both been completed
- Integration of IMU output and Box2D simulation
  - Assigned to: Skanda Nadig
  - Complete by: December 3rd
  - Progress: IMU output is being read and Box2D simulation is in working order

| Work Package | Nov | Nov | Nov | Nov | Nov | Dec | Dec | Dec |
|---|---|---|---|---|---|---|---|---|
| 1.0. Planning | Design | Implementation | | Test, Debug, Improve | | | | |
| 2.0. Screen Design | Design | Implementation | | Test, Debug, Improve | | | | |
| 3.0. Sensor testing | Tech. Req. Definitic | Test, Debug, Improve | | | | | | |
| 4.0. Hardware integration | Design | Implement logic | | Test, Debug, Improve | | | | |
| 5.0 Game Physics Box2D | Plan & Design | Implement logic | Integrate with screen & hardware | | | | | |
| 6.0 Report | | Structure | Write-up | Editing | | | | |
| 7.0. Presentation | | | | Structure; Content | | | | |

## Box2D Physics Engine Test

Here is a clip showing a golf ball working with the Box2D physics engine in a simple 4-wall map. This was our proof-of-concept (running on a Mac with arrow key inputs)  that now allows us to use the physics on the hardware and create new levels and maps.

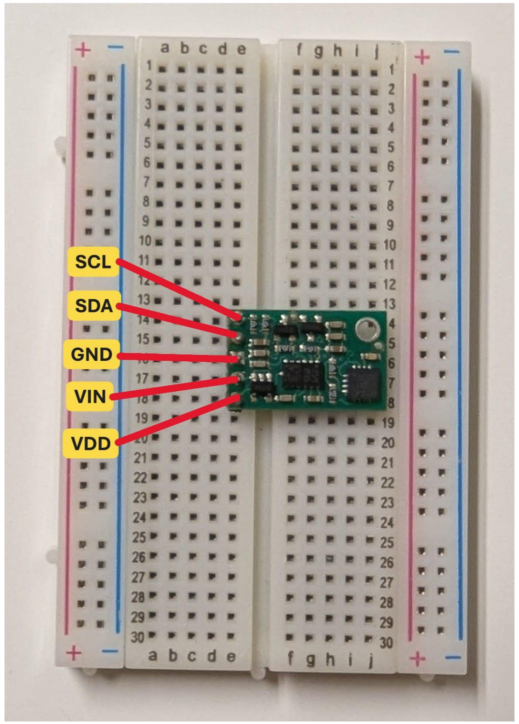


## IMU sensor(MPU-6050 / MPU-9250)

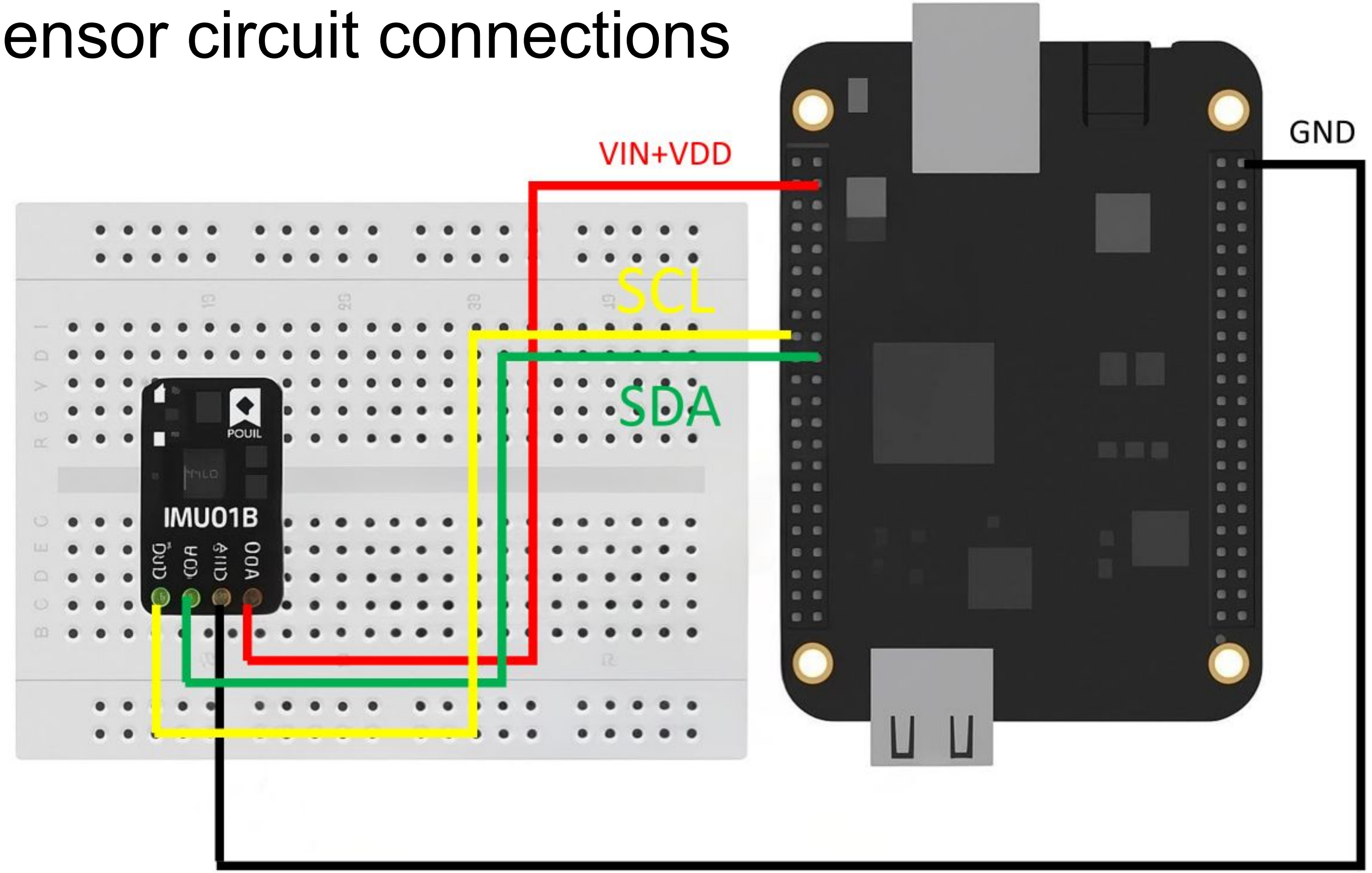
1. Purpose:
   a. Measures device orientation (tilt along X, Y, Z axes)
   b. Provides accelerometer and gyroscope data for game input
2. Connection:
   a. Connected to BeagleBone Black via I2C interface
3. Data Acquisition:
   a. Registers used:
      i. Accelerometer: ACCEL_XOUT_H/L, ACCEL_YOUT_H/L, ACCEL_ZOUT_H/L
      ii. Gyroscope: GYRO_XOUT_H/L, GYRO_YOUT_H/L, GYRO_ZOUT_H/L
      iii. Configuration: PWR_MGMT_1, CONFIG, etc.
4. Reading values:
   a. Using I2C commands (i2cget, i2c-tools) or hexdump for debugging
   b. Example: i2cget -y 1 0x68 0x3B to read ACCEL_X register

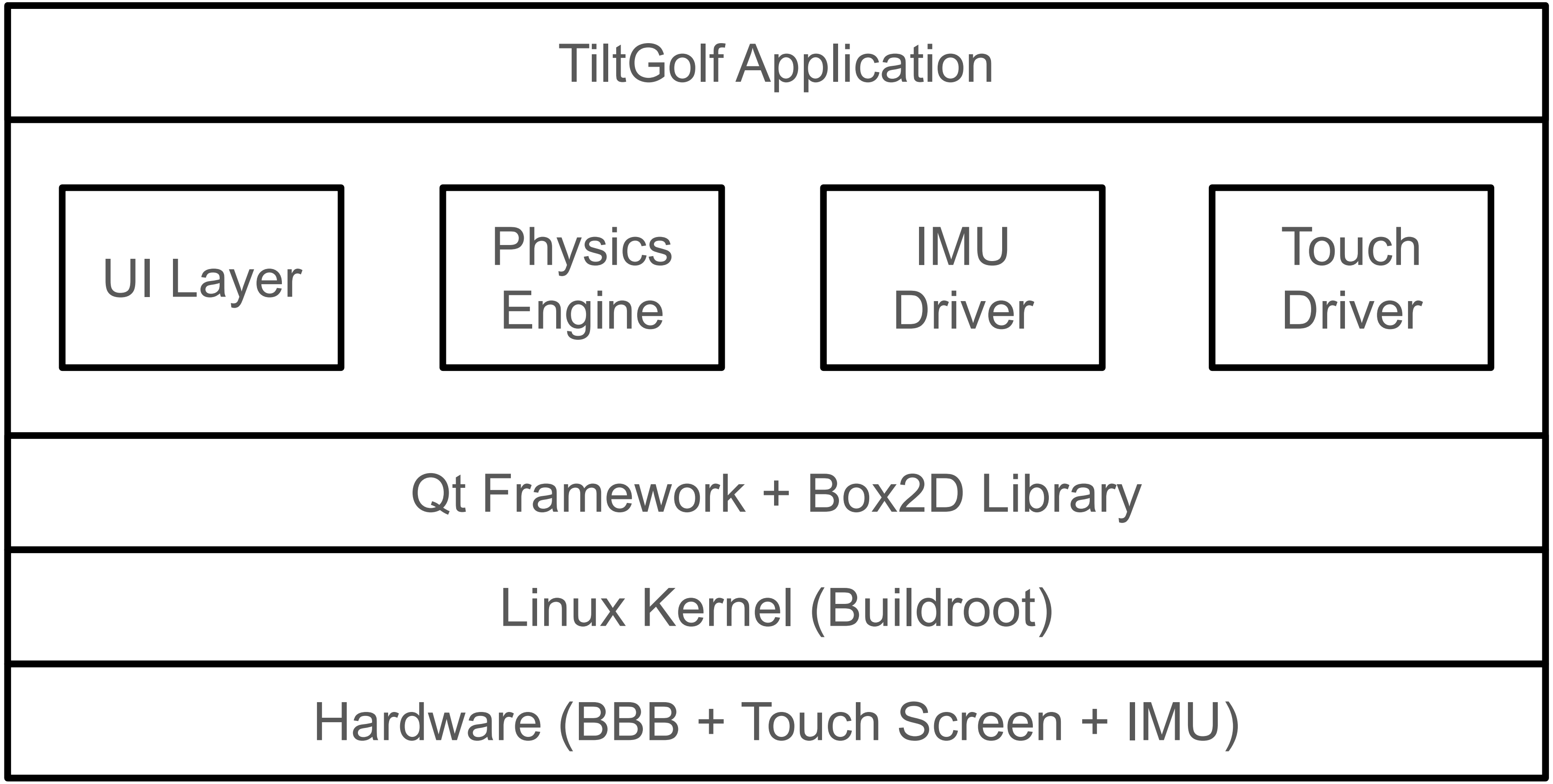
# IMU sensor Pin layout

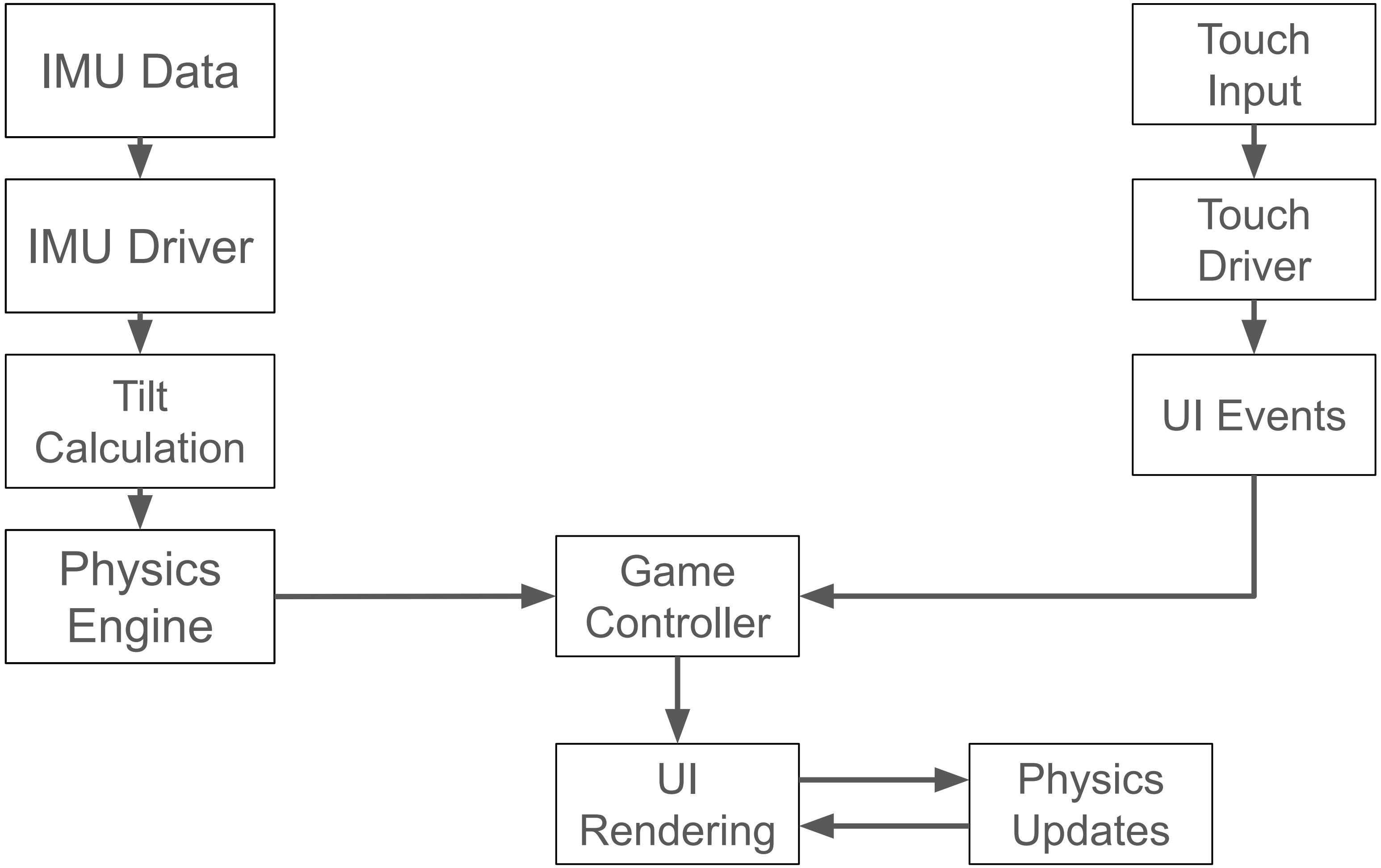
- **SCL** – I2C clock line, synchronizes data between IMU and BBB. Connected to P9_19 of Beaglebone
- **SDA** – I2C data line, carries sensor readings to the BeagleBone. Connected to P9_20 of Beaglebone
- **GND** – Ground reference for IMU and BeagleBone. Connected to P8_2 of Beaglebone
- **VIN** – Input voltage to power the IMU (e.g., 3.3V or 5V). Connected to P9_4 of Beaglebone
- **VDD** – Logic supply voltage for the IMU's internal circuitry. Connected to P9_4 of Beaglebone



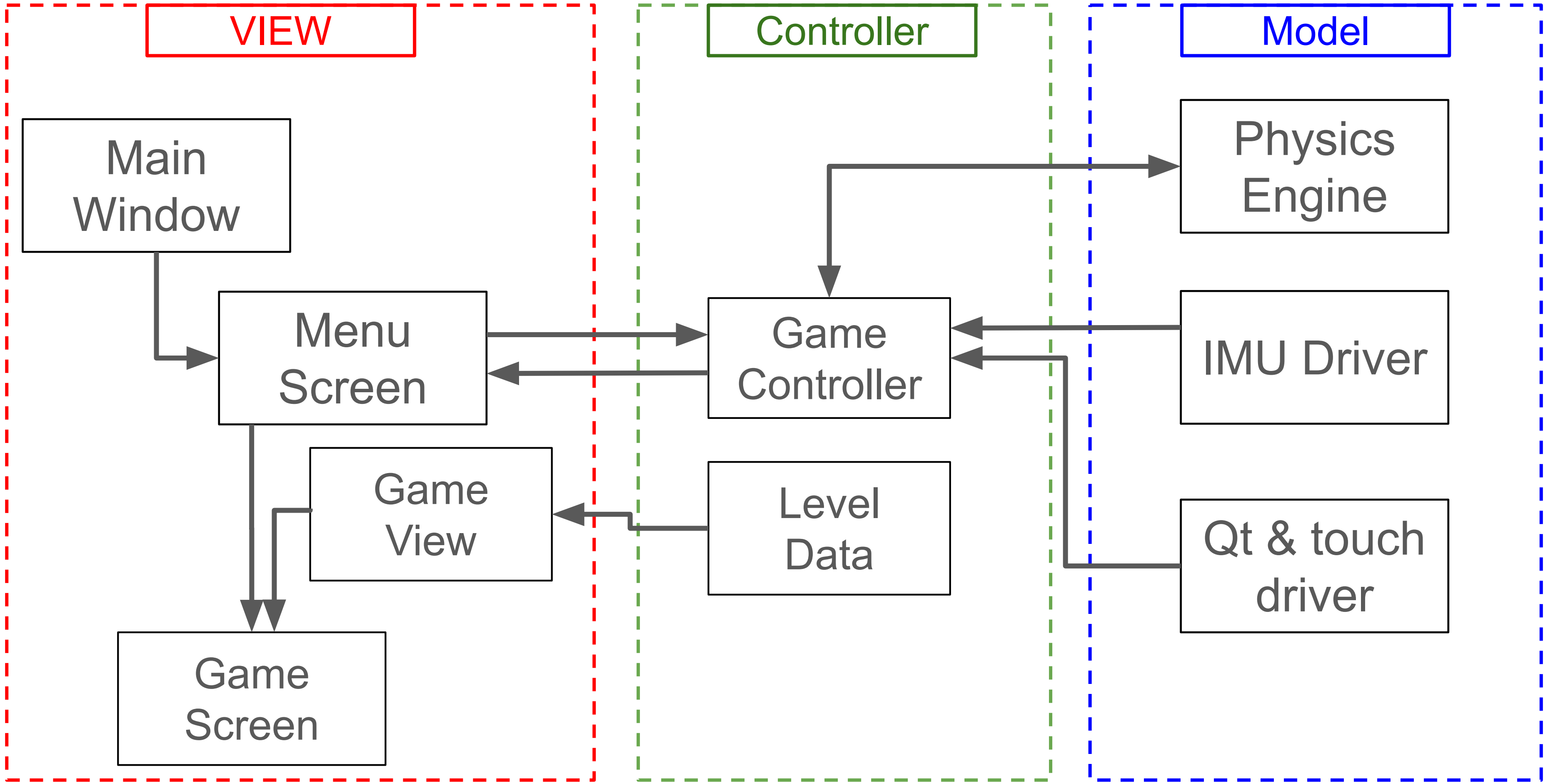
## IMU sensor circuit connections




TiltGolf Application
UI Layer
Physics Engine
IMU Driver
Touch Driver
Qt Framework + Box2D Library
Linux Kernel (Buildroot)
Hardware (BBB + Touch Screen + IMU)


IMU Data
IMU Driver
Tilt Calculation
Physics Engine
Touch Input
Touch Driver
UI Events
Game Controller
UI Rendering
Physics Updates

TiltGolf


VIEW
Controller
Model
Main Window
Menu Screen
Game View
Game Screen
Game Controller
Level Data
Physics Engine
IMU Driver
Qt & touch driver