Project Lightgun

History:

- The first light guns were produced in the 1930s, following the development of light-sensing vacuum tubes. In 1936, the technology was introduced in arcade shooting games, beginning with the Seeburg Ray-O-Lite

- Nintendo released the Beam Gun in 1970 and the Laser Clay Shooting System in 1973, followed in 1974 by the arcade game *Wild Gunman*, which uses film projection to display the target on the screen. In 1975, Sega released the early co-operative light gun shooters *Balloon Gun* and *Bullet Mark*.

Working principles:

Sequential targets

- The first detection method, used by the NES Zapper, involves drawing each target sequentially in white light after the screen blacks out.
- The computer knows that if the diode detects light as it is drawing a square (or after the screen refreshes), then that is the target at which the gun is pointed.
- Essentially, the diode tells the computer whether or not the player hit something, and for *n* objects, the sequence of the drawing of the targets tell the computer which target the player hit after 1 + ceil(log$_2$(*n*)) refreshes (one refresh to determine if any target at all was hit and ceil(log$_2$(*n*)) to do a binary search for the object that was hit).
- A side effect of this is that in some games, a player can point the gun at a light bulb or other bright light source, pull the trigger, and cause the system to falsely detect a hit on the first target every time. Some games account for this either by detecting if all targets appear to match or by displaying a black screen and verifying that no targets match

Infrared emitters

- The Wii Remote uses an infrared video camera in the handheld controller, rather than a simple sensor.
- Wesley Yin-Poole stated that the Wii Remote was not as accurate as a traditional light gun.
- GunCon 3 is an infrared light gun used for the PlayStation 3 port of *Time Crisis 4*

Rectangular positioning

- Rectangular positioning is similar to image capture, except it disregards any on-screen details and only determines the rectangular outline of the game screen.
- By determining the size and distortion of the rectangle outline of the screen, it is possible to calculate where exactly the light gun is pointing. This method was introduced by the Sinden Lightgun.

Positional gun

The positional gun is common in video arcades, as a non-optical alternative to a light gun. The positional gun is permanently mounted on a swivel on the cabinet, as an analog joystick for aiming crosshairs onscreen. This is typically more expensive initially but easier to maintain and repair. Positional gun games include **Silent Scope,the arcade version of Resident Evil Survivor 2, Space Gun,Revolution X, and Terminator 2: Judgment Day**. Console conversions may use light guns.

A positional gun is essentially an analog joystick that records the position of the gun to determine the player's aim on the screen. The gun must be calibrated, which usually happens after powering up. Early examples of a positional gun include Sega's *Sea Devil* in 1972, Taito's *Attack* in 1976, and *Cross Fire* in 1977, and Nintendo's *Battle Shark* in 1978.

light gun components comparison

Sinden Lightgun:

1. It features a high speed 60fps camera, a CPU, USB interface, a trigger, 4 assignable side action buttons, pump action reload, a 4 way D-Pad and offscreen reload.

Category	Component / Feature	Description / Function
Optical System	High-Speed Camera (60 FPS)	Captures screen border and movement data for real-time position tracking.
Processing Unit	Embedded CPU / Microcontroller	Processes camera feed, computes aiming position, and handles USB communication.
Connectivity	USB Interface	Provides both power and data transfer to the host (PC or console).
Input Controls	Trigger	Main firing input for shooting actions.
Input Controls	4 Assignable Side Buttons	Customizable buttons for secondary actions (e.g., weapon change, crouch).
Input Controls	Pump-Action Reload Mechanism	Physical reload motion for immersive gameplay.
Input Controls	4-Way D-Pad	Navigation control for menus or in-game movement.
Special Functionality	Offscreen Reload Detection	Allows reloading when pointing the gun away from the screen.
Firmware / Software	Internal Control Firmware	Interprets camera input, processes logic, and interfaces with host system drivers.
Power	USB-Powered	No external power supply required; powered directly from USB port.
Optional Accessories	Custom Lightgun Shells / Mounts	Various casing designs and add-ons for different play styles.

Floor Dull Lightgun:

1. Mpu6050 gyroscope, a bunch of pushbuttons, Arduino r4 minima, esp32, and a bunch of wires

Category	Component / Feature	Description / Function
Motion Sensing	MPU6050 Gyroscope & Accelerometer	Provides motion tracking and orientation data (pitch, roll, yaw) for aiming detection.
Processing Unit	Arduino R4 Minima	Handles sensor input, button logic, and communication with other modules.
Wireless Communication	ESP32 Module	Enables Wi-Fi or Bluetooth connectivity for wireless data transmission to a host device (e.g., PC or console).
Input Controls	Pushbuttons (Multiple)	Used for trigger, reload, and auxiliary actions such as menu navigation or weapon switching.
Interconnects	Jumper Wires / Dupont Cables	Provide electrical connections between components on the prototype setup.

wii remote :

Category	Component / Feature	Description / Function
Main Processor	Broadcom BCM2042	Bluetooth + MCU core
Accelerometer	ADXL330	Motion sensing
IR Camera	PixArt PAW3305DB	Optical tracking
Memory	24LC08B EEPROM	Data storage
LEDs	4 × Blue LEDs	Player indicators
Vibration Motor	DC motor	Haptic feedback
Speaker	Piezo speaker	Local audio
Accessory Port	6-pin	I²C peripherals
Power Circuit	DC-DC converter	3.3V regulation

nes zapper(og):

Component	Function / Purpose
Photodiode Sensor	Detects light flashes from CRT screen targets during trigger pull.
Amplifier Circuit (IC + Resistors/Capacitors)	Boosts and conditions the photodiode signal for console input.
Trigger Switch	Mechanical switch that registers when the trigger is pulled.
Lens Assembly	Focuses light onto the photodiode and limits field of view.
Light Shield / Barrel Tube	Blocks ambient light interference for accurate detection.
Internal Wiring & PCB	Connects trigger, sensor, and amplifier to output connector.
Controller Connector Cable	Sends hit and trigger signals to NES controller port.
Metal/Brass Weights	Provides physical balance and realistic gun feel.

IvoryRubble_Lightgun:

DF Robot IR positioning camera,buttons for WASD,L-34F3C infrared LEDs ,(aurdino m4,m0,32u4(3v),32u4(5v)),100ohm resistors 2x,tall switch

Category	Component / Feature	Description / Function
Optical System	**IR Camera Module – DFRobot SKU SEN0188 (Wii IR Camera Breakout)**	Extracted from a Wii Remote sensor; detects up to four IR light points from the screen / sensor bar for aiming. Resolution ≈ 1024×768 @ 100 Hz.
Illumination System	**IR LEDs – Lite-On L-34F3C (940 nm Infrared Emitter)**	Two or more placed near the display to serve as infrared tracking beacons. Typical Vf = 1.2 V @ 20 mA.
Processing Unit	**Arduino Micro (Board Rev 3, ATmega32U4 16 MHz)**	5 V logic microcontroller with native USB support; reads the IR camera, buttons, and sends USB HID reports or wireless packets.
Connectivity – Wired	**Micro-USB Type-B Port (on Arduino Micro)**	Provides data and power for wired operation; enumerates as USB HID device.
Connectivity – Wireless Option	**nRF24L01+ 2.4 GHz Transceiver Module (PA/LNA variant optional)**	Handles wireless link between gun and USB dongle receiver; SPI interface to ATmega32U4.
Receiver Module	**Arduino Micro + nRF24L01+ (USB Dongle)**	Second board acting as receiver; converts incoming wireless data to USB HID mouse/joystick events for the host PC.
Input Controls	**Trigger – Momentary Tactile Switch (6×6 mm or Panel Mount)**	Primary fire control; connected to a digital input pin.
Input Controls	**Side Buttons × 4 – Tactile SPST Switches**	Assignable to secondary functions (reload, weapon change, crouch, etc.).
Indicators	**Status LED – 3 mm Red/Green**	Indicates power / wireless / fire status.

Working:

Sinden lightgun working:

- It has a high quality usb camera.mounted on the front of the lightgun.
- There are other hardware that captures user inputs such as trigger and also provides mouse inputs to the computer.
- Software is being programmed with knowledge of what is being displayed on the television. This is done by accessing the real time video output or knowing what will be in the output.
- The software recognizes the television display on the camera image. the center of the camera image is exactly where the camera is pointing.knowing where the television display corners are in relation to this gives the information required to calculate where on the television display the sinden lightgun is pointing.
- This is then communicated by the computer back to the lightgun and it uses additional hardware to act as a hardware mouse and move the cursor.
- A virtual mouse driver is also a possibility for just feeding the raw coordinates into the software being controlled.
- The television display is going to be distorted based on a angle to the tv and which area of the display we are pointing at. so the software has to use a clever algorithm to undistort the image and correctly work on where we would be pointing on a undistorted image.
- Sinden lightgun can hence be used from different angles and dosen't require calibration.

Image processing:

- the whole image processing algorithm runs very fast.it takes between 5-10 ms to process image frame depending on the camera quality settings.
- algorithm can be optimized. no gpu processing is being done. just sub frame processing is done.
- The camera provides frame every 33ms as it is a 30fps camera and on choosen camera quality settings the frame is processed to calculate position in about 7ms.therefore average lag =40ms or 25 frames/second which provides good playable experience.
- If 60fps camera is used the processing could be done faster.

Wii remote working:

- inside the remote there is is a speaker and a audio amplifier which allows for **audible feedback** from right inside the remote.
- It also has a standard memory chip capable of storing upto 16kb of information.
- It contains an **accelerometer** which records the direction and strength to the forces being applied on the wii remote.
- The **data converter** then converts these analog forces in a digital data that can be sent to wii via bluetooth.
- The accelerometer detects :
    1. front to back (z axis)
    2. side to side (xaxis)
    3. up and down (yaxis)
- it uses these three forces to calculate exactly which direction a force applied to it is travelling.
- It can also detect **Bank** and **Pitch.**
- **Pitch –** to find the pitch it finds the force of gravity which is always pointing down and compares with the angle it creates with the wiimote.
    1. If the angle is acute then it knows that the wii mote is tilting downwards
    2. if the angle is obtuse then it knows the wii-mote is tilting upwards
- **Bank-** same applies to the bank. It measures the angle of gravity versus the wiimote and calculates its tilt.
    1. By combining the two values we can see how the wiimote knows exactly which way it's being tilted.
- On the other side of the wiimote we can find a **rumble pack** .a rumble pack creates physical vibrations in the remote that corresponds with an event on the screen.
- It has a large chip in the center (bluetooth chip) responsible for transferring information to and from the wii-mote and the wii.
- The wiimote has a chip for translating analog data such as human speech into digital data stream(it acts as a microphone).
- On the bottom there is a port for plugging In the controller attachments.
- It is powered by 2 AA batteries
- it consists of an infrared sensor used to detect and translate signals sent to it by sensor bar.
- **Sensor bar:** the sensor bar dosen't sense anything .instead it has two sets of **5 infrared lights** that the wiimote uses to triangulate it's location onto the screen.

1. When the lights from the sensor bar moves to the bottom of the view. It knows that the cursor should be at the top of the screen.
2. When the lights are on the top of the view, the cursor is placed at the bottom of the screen.
3. Same thing applies to the left and right motion.by reversing the angle created by the lights the remote can display a slanted cursor on the screen.

Nintendo zapper working:

- inside the zapper there is a **shielded photo diode** that detects light.
- When we first pull the **trigger** on the zapper the game first blacks out the screen. the photo diode inside the gun first looks for the black screen .
    1. Suppose the diode detects the light instead of the black screen it determines that the player is not pointing the gun on the screen and the game records a miss.
    2. If the diode does see black.the screen flashes a **white hitbox** over the target for 1 frame. the photo diode now looks for the light if it detects the light the game registers a hit .
    3. If there are multiple targets. The screen will flash each hitbox **seperately** one after another (known as sequential targeting).the photodiode then determines if it hit any of them.

My implementation of the Lightgun:

- **Leaning towards IR technology due to bias towards wii controller.**

- **Also it is kind of an evolution of a photo sensor as seen in the nes zapper something optical in the sense.**

required components/working

Category	Component/feature	Description	Cost:
Optical system:	**IR Camera Module – DFRobot SKU SEN0188**	Will be used to detect the IR lights from the sensor bar on the screen	Rs.2,151
Emission	**IR led emitter**	Will serve as a means for tracking position	Rs.90
Connectivity wired	**Usb**	Usb wired connection	nil
Connectivity/ wireless	**nRF24L01 2.4 GHz**	Used for bluetooth connectivity.(ex-change weapon)	Rs.70 or nil
Main input	**Trigger**	Serves as the primary control for shooting targets.	nil
Secondary input	**pushbuttons/ switches**	Used for other functions	Rs.15
Secondary motion sensing	**MPU6050 Gyroscope and Accelerometer**	Will probably be using it for offscreen reload function.	Rs.200 - 400
Audio system	**Speaker and audio amplifier**	Will be used to provide auditory feedback to action	Rs.200
Interconnection	**Jumper wires**	To provide connections between components	Rs.40
Processing unit	**Stm32 bluepill/black pill**	Will serve as the main micro controller to handle the system.	Rs.300