Final Project Proposal: Beat Gunner

By TungShen Chew, Stephanie Cheng, An Li

November 8, 2009

1. Introduction

Beat Gunner is a shooting & rhythm game where the player fires a gun at two moving targets on a screen and is rewarded for hitting them in time with the beat of the music. The targets change speed and direction in time with the music as well, making them harder to hit as the tempo of the music increases.

The music for the game is provided externally and the game performs real-time beat-detection to determine the behavior of the targets and the player's score. This allows the player to treat songs as different 'levels' where the beat content of each song dictates the game experience.

2. Game Mechanics

2.1. Detecting hits and misses

Beat Gunner determines if the player has hit the target by monitoring the light level observed by a photodiode inside the barrel of the gun. When the player pulls the trigger, the following actions are executed in order:

1. The game turns the whole screen black.
2. The game records the observed light level of the gun.
3. A white box is drawn at the location of one target.
4. The game checks if the observed light level has suddenly increased; if it has, it confirms that the target has been hit.

Steps 3 and 4 are repeated for each target, so a 'hit check' is performed in $(n_{targets} + 1)$ video frames.

2.2. Beat detection algorithm

The beat detection algorithm works on the assumption that the sub-bass region of music (below 100Hz) only contains deep drumbeats. It is thus possible to detect beats by passing the music through a low pass filter followed by a peak detector. A moving threshold can be placed at 90% of the highest observed peak, and everything above the threshold can be assumed to be a beat. Beats are detected in real time but the music playback is delayed by one second, allowing the player to 'see' incoming beats (displayed on top of the screen) one second before they arrive.

2.3. Scoring

Whenever the player successfully hits a target, the score counter increments by a value which decreases with the time difference between the moment the trigger was pulled and the most recent beat.

3. Block Diagram and Description of Modules

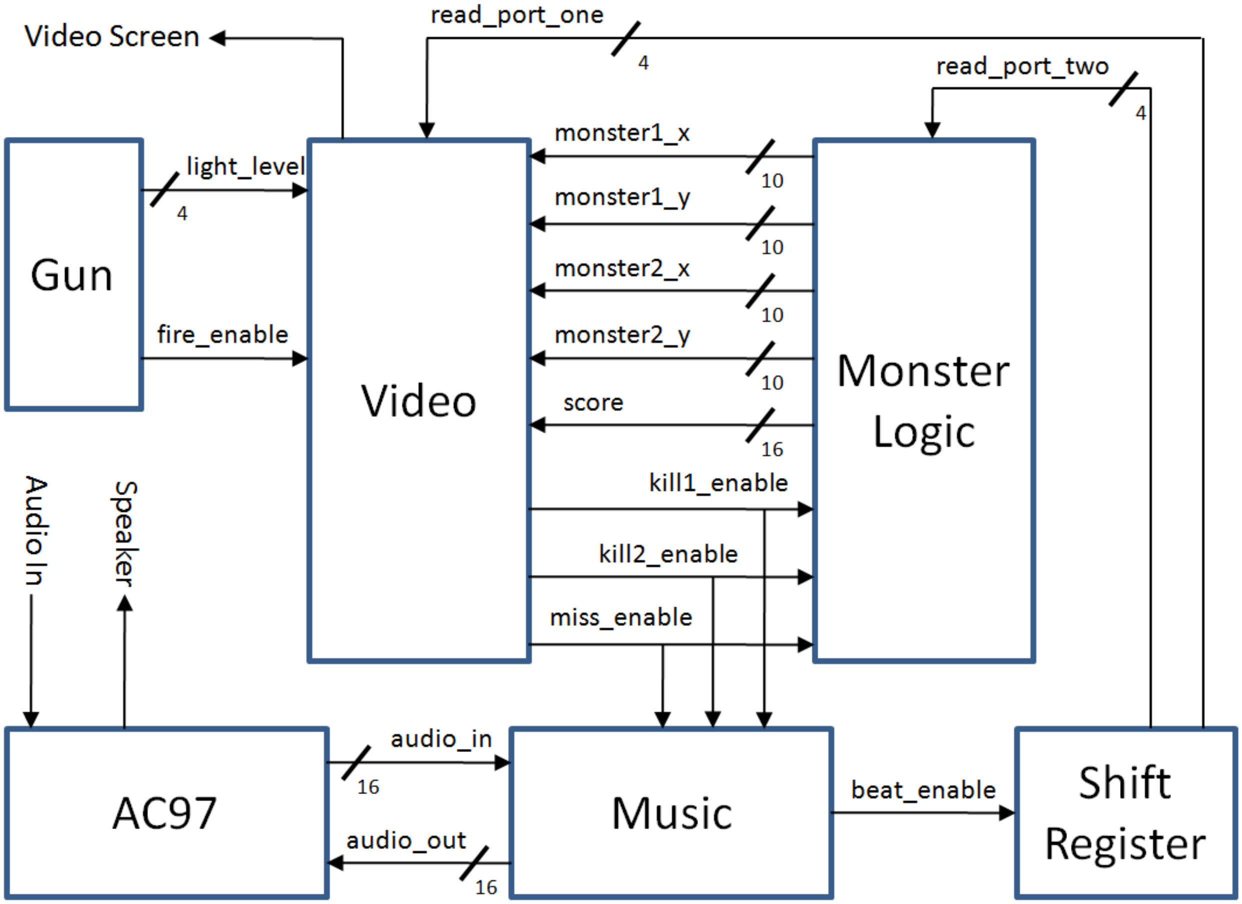

Figure 1: Block Diagram of Beat Gunner modules

The Beat Gunner hardware is divided into six modules as shown in Figure 1. They are described below:

3.1. AC97 Module

Inputs: Audio In (external), audio_out
Outputs: Speaker (external), audio_in
The AC97 module is the standard audio processing module which is built into the 6.111 labkit. In this application it is being used as an audio ADC/DAC to record the audio input and allow playback.

3.2. Music Module

Inputs: audio_in, miss_enable, kill1_enable, kill2_enable
Outputs: audio_out, beat_enable
The music module performs the beat-detection algorithm and contains half-second 15-bit 48 kHz sound effects for hitting or missing a target. It implements the one-second playback delay using 1Mbit of BRAM to store one second of 15-bit 48 kHz music data from the external audio source. Whenever a target is hit or missed, the music module adds the appropriate 15-bit sound effect to the 15-bit music data to form a 16-bit output signal to be sent to the AC97 module.

3.3. Shift Register Module
Inputs: beat_enable
Outputs: read_port_one, read_port_two
The shift register module is notified of confirmed beats by the music module and stores two seconds' worth of beat information in 96,000 registers. This information is made available to the video and monster logic modules via a pair of serial read ports.

3.4. Monster Logic Module
Inputs: read_port_two, miss_enable, kill1_enable, kill2_enable
Outputs: monster1_x, monster1_y, monster2_x, monster2_y, score
The monster logic module keeps track of the x-y position of the two targets and resets the position of a target whenever it is shot. It also correctly calculates the new score based on information from the shift register about the current proximity of the music beats.

3.5. Video Module
Inputs: read_port_one , monster1_x, monster1_y, monster2_x, monster2_y, score, light_level, fire_enable
Outputs: Video Screen (external), miss_enable, kill1_enable, kill2_enable
The video module draws the monsters on the screen at the correct x-y coordinates based on data from the monster logic module. It also displays the beat data from the shift register as a series of moving lines on top of the screen, such that the player can 'see' and anticipate approaching beats. Most importantly, the video module directly communicates with the gun module to determine if a monster has been hit whenever the gun trigger is pulled.

3.6. Gun Module
Inputs: None
Outputs: light_level, fire_enable
The gun module communicates with the gun to transmit a fire_enable signal whenever the trigger is pulled. It also contains an ADC to convert the output of the phototransistor in the gun to a digital value.

4. External Components

4.1. Gun
The gun will be constructed out of Lego and have a phototransistor inside a long, reflective barrel, a switch for the trigger and (potentially) a solenoid to simulate the recoil of the gun being fired.

4.2. Speaker
The powered speaker will be borrowed from a team member for the duration of the project.

4.3. Audio Source
The audio source could be anything at all – it is a major feature of this project that the player can use any audio source he or she pleases. For testing purposes a team member's laptop will be used.

5. Testing

The testing procedure of the modules can be adequately generalized as follows:

1. The music, video and monster logic modules have been designed to communicate mostly with single-bit enable signals. The effect of these inputs can be tested individually by linking them to debounced pushbutton designed to set the signal high for only one clock cycle.
2. The video and monster logic modules can be tested with a simulated version of the shift register which loops through a test beat pattern.
3. The shift register can be tested by reading the value of several registers along its length, sending it a single beat_enable signal and watching the signal propagate down the registers over the course of two seconds.

6. Team Responsibilities

TungShen Chew will design the Monster Logic and Gun modules and build the gun.

Stephanie Cheng will design the Music and Shift Register modules and configure the AC97 module.

An Li will design the Video module.