Virtual Rock Climbing: A video game using tracking and tactile feedback

Turner Bohlen Chris Lang

November 20, 2013

1 Introduction

This project aims to create a rock climbing video game in which the player can climb a virtual wall by making motions in the real world. The player will be able to reach up, down, left, and right in reality in order to find and grasp the virtual handholds. A force feedback system will provide tactile information about the location of holds, and a computer display will provide visual cues. If the player lets go, they will fall back to the beginning. The game is over when the player reaches the top of the virtual wall.

Hand tracking will be achieved using two colored gloves, one for each hand, and a video camera. Force feedback will be provided by one or more small vibrating motors. Finally, a button placed on the glove will register when the player has grabbed a virtual hold.

Stretch goals include creating a three dimensional environment, a map editor, and adding head tracking in order to create a virtual headlamp that allows the user to only see a small number of holds at a time.

2 High-Level Design

The system will have three major components: the tracking block, the gameplay block, and the glove block. See Figure 1 for their interactions.

2.1 Tracking Block

The tracking block will analyze video input from a camera and determine the position of the players hands on screen. It will take the frames and ancillary information as inputs and output the position of the players to the gameplay block. This section will be constructed by Turner Bohlen.

2.2 Gameplay Block

The gameplay block will handle the physics of the game, as well as outputting video information to the VGA display. The gameplay block will take as input the hand position from the camera block as well as grab information from the glove block and control information from the display. From this information, it will determine the location of the player on the rock wall, the objects to display on screen, whether or not the user is grabbing a handhold, and

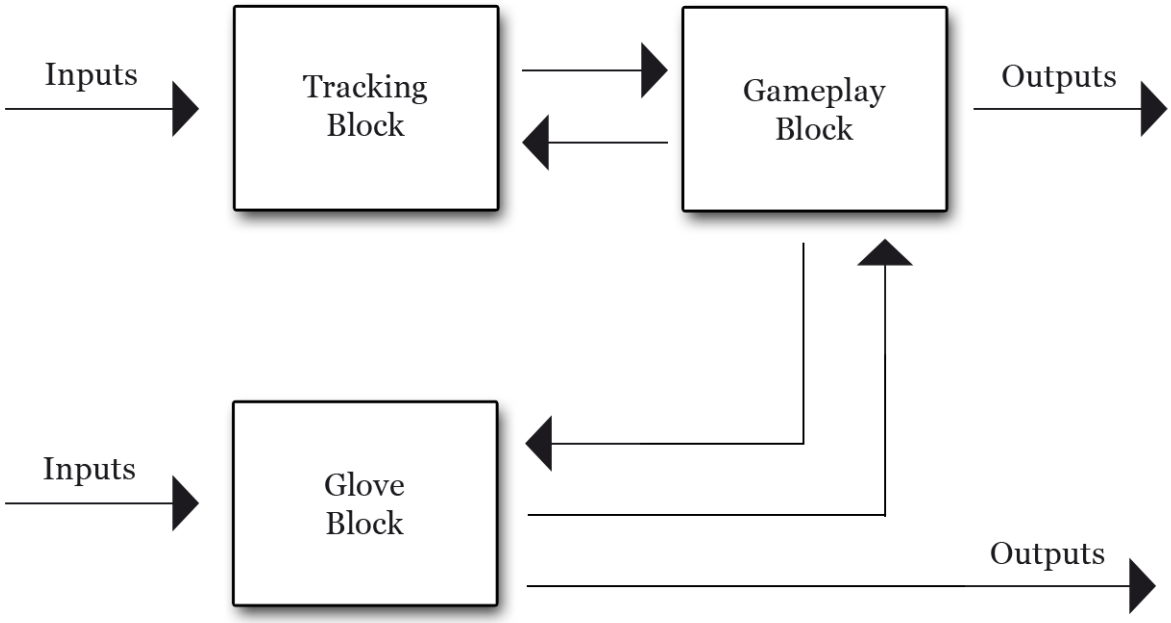

Figure 1: The high level block diagram shows the three major components of the system: the tracking block, the glove block, and the gameplay block. It also indicates information flow throughout the system

the pixel information to display in the next frame. This section will be constructed by Chris Lang.

2.3 Glove Block

The glove block will manage inputs to and outputs from the gloves. It will take information from the gameplay block describing if and how the user is touching objects on screen and output feedback signals to the glove. It will also debounce and output signals from the gloves' buttons. This section will be constructed by Turner Bohlen.

3 Hardware

This system requires a significant amount of additional hardware.

First, two gloves are required to allow for hand tracking and tactile feedback. Two colored gloves will be borrowed from the 6.111 lab supply, one red and one green.

Second, a number of vibration motors will be purchased. The 6.111 staff has confirmed that the cost of these components will be reimbursed. The following model sold by Adafruit will has been chosen for the project. : http://www.adafruit.com/products/1201?gclid=CPz6gYTAwroCFZGf4AodCi8AGw.

Third, two push-buttons will also be used. These will be borrowed from the 6.111 supply. We still need to confirm that these are available.

Forth, and finally, one video camera will be used to gather position information from the real world and insert it into the game world. This will be borrowed from the 6.111 lab supply.

If possible, a virtual headlamp will be added to the game, allowing the user to tilt their head in order to see different portions of the wall in front of them. The entire wall will never be visible. To accomplish this stretch goal, a styrofoam ball will be purchased online and painted blue to provide a trackable color mass.

Turner Bohlen will construct the hardware components of this system. Parts should be received by November 15, 2013 and construction of the first version of the hardware system should be completed by November 22, 2013. Additional modifications may be necessary after this date.

4 Tracking Block Design

The tracking block uses camera inputs to calculate hand positions. Four modules will comprise the block. These can be seen in Figure 2.

4.1 Camera Module

The camera module will process inputs from the camera and down-sample them. The basis for this module will be the already implemented 6.111 camera module provided by the course staff. Modifications on top of this base will allow for the functionality necessary for this game. Inputs will come directly from the camera. Outputs will be control signals and pixel data.

4.2 Frame Buffer

A frame buffer will use ZBT memory to store two down-sampled frames from the camera. It will take as input the control signals and pixel data from the camera module and a number of bits from the downstream center of mass module specifying which pixel to output next. It will output a single pixel of information and any necessary control signals.

The complexity of this module is low, although it will use a significant amount of ZBT memory and therefore require careful clock manipulation.

4.3 Center of Mass Module

The center of mass module calculates the center of mass of red, green, and blue color in each frame fed from the camera. It takes as input pixel data and any necessary control signals from the frame buffer and outputs the x and y coordinates of the center of mass of each color.

This module is somewhat complex. It must do a weighted sum over all pixels on screen. It will loop through all pixels on screen at 27 Mhz, storing in a set of registers the running value of each center of mass. Division will be necessary for the weighting calculation, significantly increasing the complexity of this module. An approximation may be used that rounds all values to powers of two if that is deemed accurate enough.

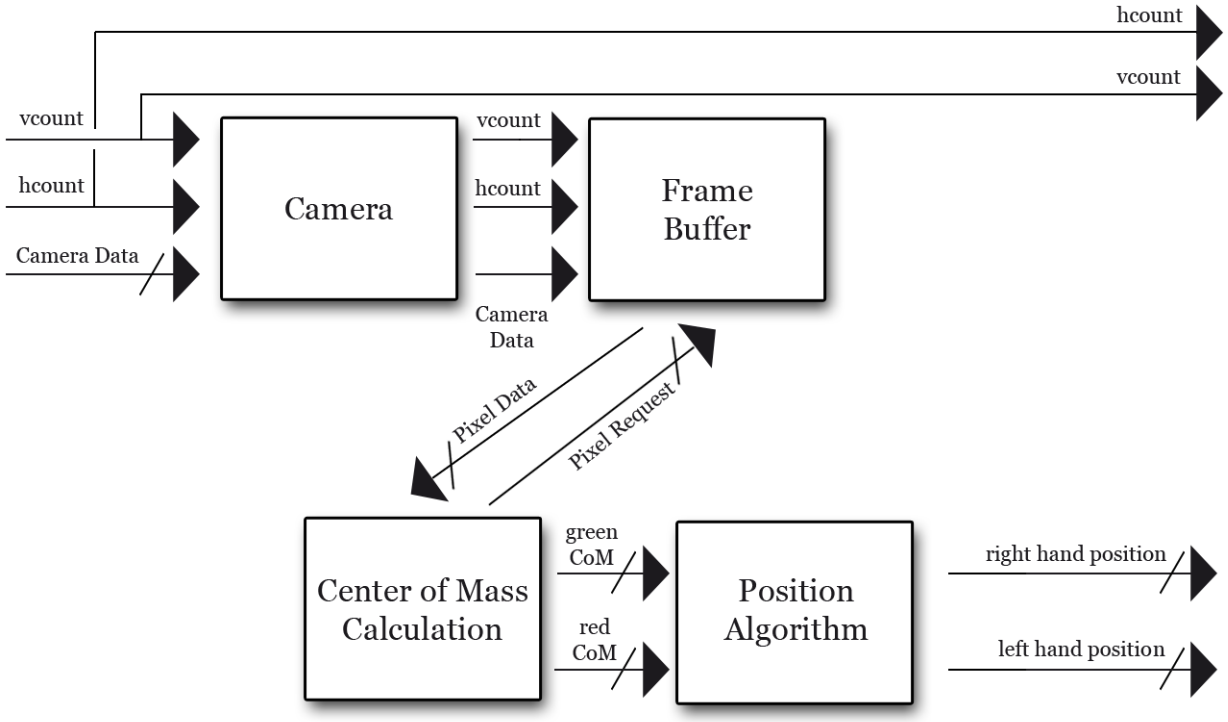

Figure 2: The tracking system has four primary steps. First information is processed from the raw camera input. Second, frames are stored for later use. Third, center of mass information is calculated for each color. Finally, a smoothing function using past velocity and position information to prevent errors. The result is accurate information describing the position of the player's hands.

4.4 Position Module

The position module will take as input the center of mass information from the center of mass module. It will output the position of each hand. This algorithm will store the last position and velocity of each hand and use that information to calculate the updated positions and velocities. This should allow for improved accuracy when compared with a simply center of mass position calculation.

This module will be fairly simple, using two registers to store past positions and velocities, and running short algorithm to calculate a next position and next velocity.

4.5 Timeline

This block will be the first goal for Turner Bohlen. Construction will proceed in the order shown above, starting with the camera module and proceeding until the position module is completed. A first working version of this block will be completed before Thanksgiving. The specifics of the timeline can be seen in figure 3.

5 Gameplay Block Design

The gameplay module will model the physics of the game, as well as output video information to be displayed on screen. Based on information from the grab button, player hand position, and hcount and vcount signals, it will see which of the player's hands are grabbing holds, and simulate their movement accordingly. The hcount and vcount signals indicate which pixel on screen is currently being evaluated, and will be the same as used in the tracking module. It will then send this information to the VGA out component of the labkit. Additionally, a feedback signal will be sent to the glove module to signal when a player is grabbing.

The gameplay module will be broken up into four submodules: hand-holds, grabbing, movement, and pixel information, which can be seen in figure 4. The hand-holds module will keep track of where the hand holds are on the game map, and which onscreen pixels are occupied by hand holds. The grabbing module will take information from the hand holds, player hand positions, and grab button, to decide whether the player is grabbing onto any of the holds. If he is, the corresponding glove will be sent a haptic

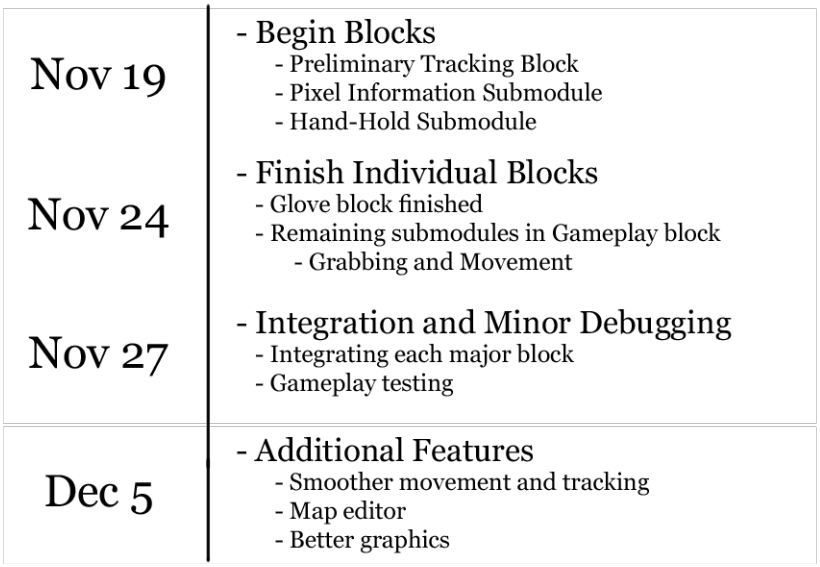

Figure 3: The projected timeline for the game. The first two weeks will be devoted to creating the major blocks. The third will be devoted to debugging and integration. The final will be used to add additional features to the game.

feedback signal. This information will also be sent to a movement module. The movement module will use the players hand positions, and whether each hand is grabbing onto a hold to simulate movement of the player. Finally, the pixel information module will take the pixel by pixel information from the hand holds, as well as of the positions of the players hands, to output pixel by pixel information, as well as its corresponding hcount and vcount signals, to the VGA out component of the labkit.

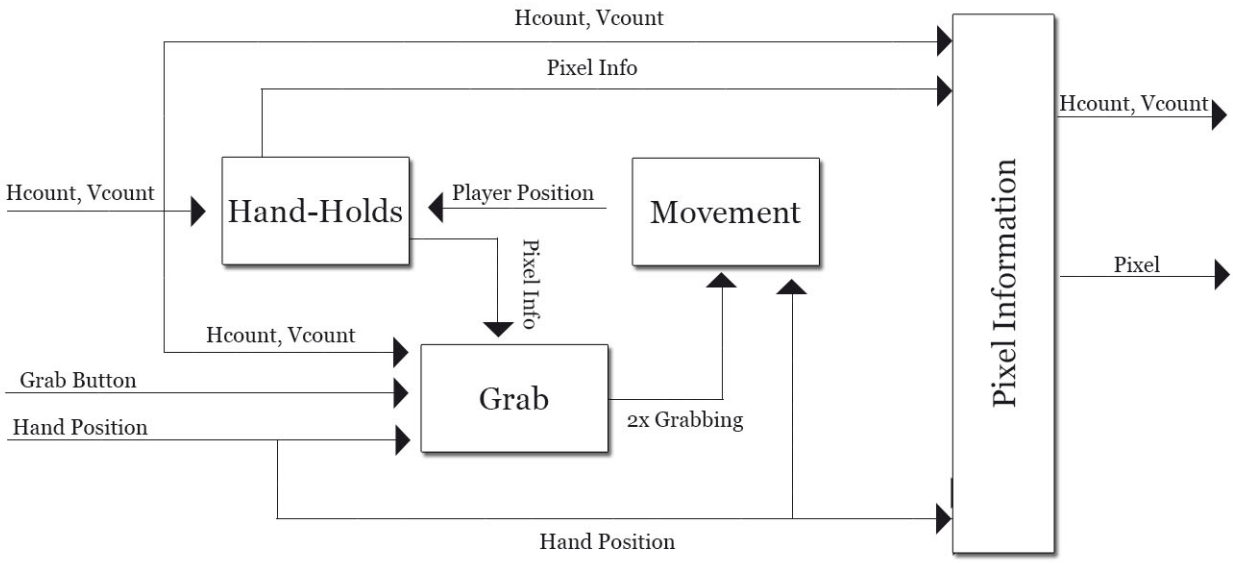

Figure 4: The gameplay block is comprised of four modules: hand-holds, grabbing, movement, and pixel information

The internal game map will be large, and only a portion of it, centered around the position of the player, will be shown on the screen. Because of this, there will be a mixing of relative and absolute positions during calculations. The absolute positions are in relation to the map as a whole, and relative positions are in relation to what is shown on the screen.

5.1 Hand-Holds

The hand-hold submodule will in fact be multiple submodules, however they will act as one. Each instance of the hand-hold module will represent a single hold in the map, and will have an x and y coordinate associated with it, indicating its absolute position. The holds will also have a width and a length, over which the holds are present.

Each hold sub-module will take in an hcount and vcount signal, as well as the screen's absolute position information and determine, pixel by pixel, whether a hand hold is present. This will be done by adding the screen x and y coordinates to the hcount and vcount coordinates, and seeing if they lie within the width and height of each modules x and y coordinates. If it does, it will output a one, associated with the corresponding hcount and vcount signal. If not it will output a zero. The outputs of all of the individual hand hold modules with then be anded together to signal if the current pixel, indicated by hcount and vcount, is occupied by any hand hold.

Initially, the placement of the hand holds will be predetermined. However, if time permits, a map editor module will also be implemented. By flipping a switch, the player can enter this editing state. While in this state, the virtual camera position will be controlled by onboard buttons, and a cursor will be controlled by the right hand position. Clicking the right hand grab button will create a hand hold, and clicking a left hand button will delete one. This is a lofty goal, and the details are not yet worked out, but could be implemented if there is time.

5.2 Grabbing

For each hand, a grabbing module will determine whether or not the player is currently grabbing onto a hand hold, updating its output at every vsync. It will do it in a similar, pixel by pixel, manner as done in the hand-hold module. Its inputs will be hcount and vcount signals, the pixel by pixel information coming from the holds, hand positions relative to the screen, and signals coming from both of the grab buttons. It will have one output registers, grab, for each of the two hands. When hcount and vcount are equal to one of the relative hand positions, it will check to see if the player is pressing the corresponding grab button and if there is a hand hold at that pixel. If these are all true, then it will signal to make the next output a one. Otherwise it will be made a zero. Additionally, this will be send to the glove

to signal haptic feedback, as well as to the movement module.

5.3 Movement

The movement module will be in charge of determining the players absolute position. It will use the player's relative hand positions, which of his hands are grabbing holds, and his previous velocity to do so. In actuality, the players position will always be fixed in relation to the camera, and will always be displayed as a fixed-position circle onscreen, so the movement module will actually be updating the screen position. The screen position is then fed back to the hand-hold modules at every vsync.

The movement module will have two different states: one for when the player is currently grabbing a hold, and one for free-fall. If the player is grabbing a hold, their position will be determined by an anchoring hand. This is to avoid complications due to the player grabbing holds with both hands, and moving them in a way that violates the game. Typically, the anchoring hand will be the hand that the player most recently grabbed a hold with. However, if the player lets go with the anchoring hand, the movement module will assign the other as the anchoring hand. Whenever a hand becomes the anchoring hand, its current relative position will be stored, as will the absolute position of the screen.

When a player is grabbing a hand hold, the movement module will update the camera position based on how far the anchoring hand has moved since it first anchored. It will subtract the current relative hand position from its relative anchored position and add it to the stored absolute screen position. If a player moves his anchored hand down, the screen will also shift down, allowing the player to reach a higher hand-hold with his non-anchored hand.

If the player is not grabbing any hand-hold, he will be in freefall, with a corresponding x and y velocity. When he first releases his hand, the gameplay module will take an average of his most recent velocities in the x and y directions, and set that to his freefall velocity. With each frame, the movement module will update the camera position based on this velocity, and the y velocity will be decremented in order to simulate gravity. This will allow the player to quickly pull down and release his grip, propelling him high into the air in order to grab a previously unreachable hold.

5.4 Pixel Information

The pixel information module will output a color, along with appropriately pipelined hcount, vcount, and vsync information to a VGA output. It will take hcount and vcount signals, pixel information from the hand-hold module, and player hand position to determine the value of the current pixel output. It will first check to see if the hcount and vcount signals are within a small radius from the hand positions. If so, it will set the pixel output to red, signifying player hand position. Otherwise it will check to see if the hcount and vcount signals are within a small radius from the center of the screen. If so it will set the pixel output to green, signifying the players position. Otherwise, it check to see if the pixel information from the hand-hold module is true. If so, it will set the pixel output to yellow, otherwise black.

5.5 Game Module Complexity

The game module is a relatively simple design. It does not rely on a large number of multipliers or memory, and all buses between modules are at most 16 bits wide. The grab, movement, and pixel information modules each use less than ten multiplexers, adders, and multipliers. The largest submodule is most likely the hand hold module. The hand-hold module will be created by many smaller modules, each representing a single hold on the map. Each of these will be comprised of four 16 bit adders, and four 16 bit logic gates. If there are 25 hand holds on the map, this will amount to 100 16 bits adders and logic gates, which is still within the capacity of the labkit.

5.6 Module Timeline and Testing

There are four gameplay sub-modules that must be completed, all of which will be done by Chris Lang. By the 19th, half of the submodules will be completed. By the 24th, the entire gameplay block will be completed. Integration with the other two blocks will then be finished by the 27th. The remainder of the time will then be dedicated to adding additional features. The timeline for which can be seen in figure 3.

Each sub-module will have to be tested individually. During testing, onboard buttons will be used to simulate hand position. The pixel calculator will be the first to be tested. Testing will ensure that the hand positions, as well as the player position, are correctly being displayed on screen, and that

overall video output works. Then the hand-hold module can be tested. The screen position will stay fixed, and a few hand-holds will be placed on the game map. If the hand-hold modules work correctly, they will be displayed on screen along with the hand and player positions. Next, the grabbing module will be tested by using the the onboard buttons to simulate the hand positions and the grab buttons. If the grab module detects that a hand is currently grabbing, it will light up an led, signifying that it is working correctly. Finally, the movement module will be tested by using the onboard buttons to grab a hold, and move the player position. If this works, then the entire gameplay module is functioning correctly, and it is ready to be integrated with the tracking module.

6 Glove Block Design

The glove block will manage inputs from and outputs to the two gloves the player wears while using the game. Two modules comprise the entire block.

6.1 Tactile Feedback

The tactile feedback module takes as input a short sequence of bits from the gameplay block and outputs two signals to the gloves, one for each vibration motor. The complexity of this system is minimal. It will use a divider and a couple counting registers in order to control the duty cycles of the two motors.

A stretch goal would be do add additional vibration motors to provide more complex tactile feedback. This would require a simple expansion of the inputs and outputs of this module, allowing for the use of more complex information about the placement of hands and objects on screen.

6.2 Debounce

The debounce module will take the raw output of two buttons from the gloves and prevent dirty or asynchronous input. The 6.111-provided debounce module will be used.

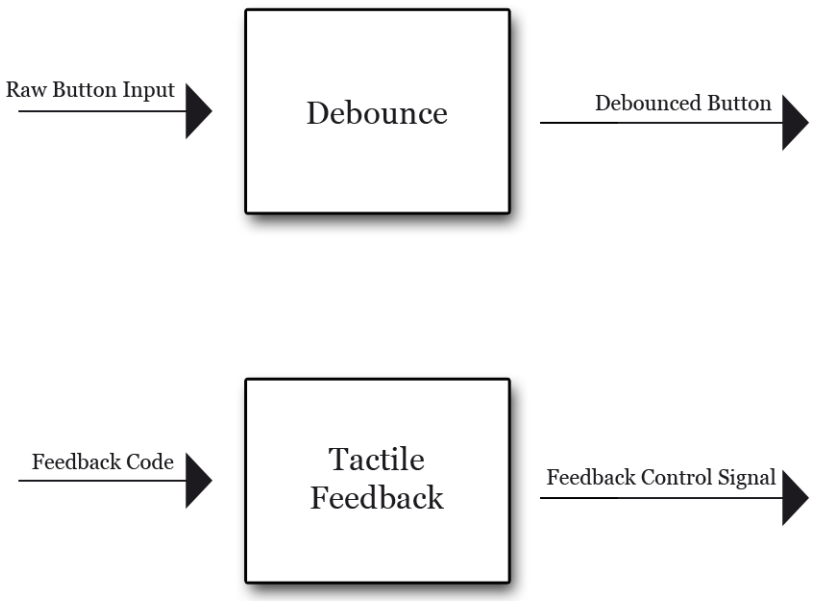

Figure 5: The glove system is comprised of two data processing steps, one for the debouncing of button input and the other for processing of tactile feedback duty cycles.

7 Conclusion

This project will be a 2D virtual reality rock climbing game. The player will wear two gloves, each with grab buttons and a haptic feedback system, to be tracked by a camera. The position of the players hands will be overlaid on-screen, allowing the player to climb a virtual wall. The game will be broken into two main blocks, in addition to the glove block, which simply controls feedback and the grab button. The first block is the tracking module. It will take in video information from a camera, store it in memory, then find the center of mass of both hands for every frame. It will then send this to a position module, which will be used to smooth out the hand position movement before sending it to the gameplay module. The gameplay module will take the hand position, to create and send video information to the VGA output in the labkit. It will do this with four submodules. The handhold submodule will keep track of the handholds on the map, the grab submodule will see if a player is grabbing a handhold, the movement submodule will control the position of the camera, and the pixel information submodule will output video information. Finally, a glove block will control the haptic feedback system, as well send information from the grab button to the gameplay module. When all three major modules are interfaced, the game will be ready to play.