

## Proposal

We have decided to make a Vegas-style slot machine, given its popularity in casinos and the popularity of online gambling. Our slot machine will rely on camera-controlled inputs from the user, instead of physical touch, like the traditional slot machine uses. In order to implement this, we are dividing our project into two parts: the inputs from the camera and buttons on the FPGA and the outputs to the screen. In the end, we will deliver a fun, interactive, and visually-appealing slot machine.

### Las Vegas Slot Machines



Figure 1: Slot Machine [1]

Slot machines are the most popular casino game in Las Vegas. The machine has three identical reels in a row with pictures on them that rotate when a lever is pulled (see Figure 1). It is a one-player game in which the player wins money based on the pattern of pictures shown when the reels stop. A player inserts coins or tokens, which represent money, in order to place a bet. After the bet is made, the player can then pull the lever down to make the reels spin. The reels spin faster or slower, in a random fashion, based on the force of the pull-down of the lever, and the payout is directly proportional to the amount of the bet placed. The ultimate goal of the game is to make all three reels show the same pictures, because then the player wins a large amount of money.

### Our 6.111 Slot Machine

Our 6.111 slot machine, unlike traditional slot machines, will have all of a slot machine's elements on a computer screen. The reels and the lever will be animated, and there will be two

on-screen buttons. One button allows the player to place a bet (the “Place a Bet” button), and the other button allows the player to quit the game and take his/her remaining money (the “Payout” button). There will be two text displays: one showing the amount of cash in the player's pot and the other showing the value of the current bet. See Figure 2 below for a visual representation.

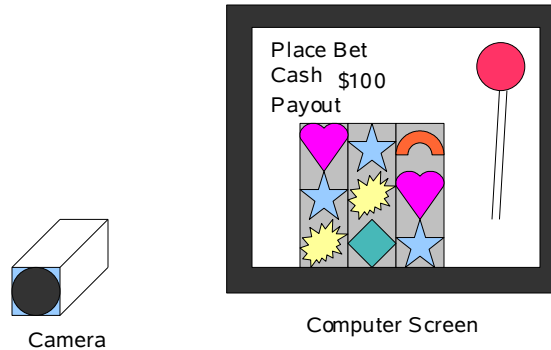


Figure 2: Our 6.111 Slot Machine

### Description of Our 6.111 Slot Machine

Each player will start out with a fixed amount of money to gamble with, and the amount will be displayed on the screen. From this pot, the player will be able to select the amount of money he/she wishes to bet from a range of \$5 to the total amount in the pot. The player will confirm this amount by “pressing” the “Place a Bet” on-screen button. In our game, pressing means standing in front of a camera with a red glove on one hand, moving the hand until the on-screen cursor is on top of the button, and letting it rest there for two seconds. After placing the bet, the player will be prompted to “pull the lever.” This means using the red glove and on-screen cursor to find the knob of the on-screen lever and making a downward motion, mimicking that of pulling a lever.

When the player runs out of money, the game is over. Also, hovering over the “Payout” button for two seconds lets the player walk away with his/her remaining money. Finally, if a player wins (according to the traditional rules of slot machines), their cash pot will be increased, and there will be fanfare to celebrate the win.

### Implementation of Our 6.111 Slot Machine

The project will be divided into two major components, and these will be broken up into modules. The first component is the game input component. This component is responsible for detecting the position of the red glove, which the player wears during game-play, from the camera. This component is also responsible for the bet input, which is placed using the switches on the FPGA. The second component is the game output component. This component handles the processing of the input and is responsible for game functionality. Lastly, there is one-way communication from the game input component to the game output component.

The modules of the first component are:

- Camera handler
- Switch handler (handled by labkit.v)

The modules of the second component are:

- RGB-to-More-Understandable-Format converter
- Camera Frame buffer
- Red Color Detector
- Center of Mass Detector
- ROM for Sprites
- VGA Displayer of Sprites From ROM
- Game Logic
- Random number generator

## The Modules

Here, we will describe each of the significant modules.

### Camera Handler:

**Inputs:** clock, video in

**Outputs:** RGB signal

**Complexity and Level of Performance:** We will have to look at the specifications of the camera before we can determine this.

**How it will be tested:** We'll hook up a camera and output the image to a screen and make sure that it's correct.

**Who Will Do It:** Daneaya

### RGB-to-More-Understandable-Format converter:

**Inputs:** RGB signal

**Outputs:** More-Understandable-Format

**Complexity and Level of Performance:** Should be fairly straightforward.

**How it will be tested:** We will feed it RGB signals and make sure we get out the correct signals.

**Who Will Do It:** Laura

### Camera Frame buffer:

**Inputs:** clock, More-Understandable-Format signals

**Outputs:** access to a memory containing the above signals

**Complexity and Level of Performance:** We will use a ZBT, giving us a high level of performance.

**How it will be tested:** We will make sure that we get out what we expect and test boundary conditions.

**Who Will Do It:** Laura

**Red Color Detector:**

**Inputs:** Signals from the frame buffer

**Outputs:** positions of the red color pixels that constitute the greatest area

**Complexity and Level of Performance:** This is complex and could potentially be slow.

**How it will be tested:** We feed it chunks of red among other colors to see if it picks out the correct spots, and we will test boundary conditions.

**Who Will Do It:** Daneaya

**Center of Mass Detector:**

**Inputs:** Takes in the output of the Red Color Detector

**Outputs:** the position of the center of mass of the red color

**Complexity and Level of Performance:** This should not be difficult, and it should be quick.

**How it will be tested:** We will give the module red blobs and make sure that it returns the center of mass correctly.

**Who Will Do It:** Laura

**ROM for Sprites:**

**Inputs:** No inputs, the sprites will be hard-coded into the ROM, clock, address

**Outputs:** Sprites

**Complexity and Level of Performance:** Making nice-looking graphics will be complex.

**How it will be tested:** We will test it in conjunction with the VGA Displayer of Sprites from ROM.

**Who Will Do It:** Daneaya

**VGA Displayer of Sprites From ROM:**

**Inputs:** Sprites from ROM

**Outputs:** VGA signal

**Complexity and Level of Performance:** We will need to make sure that the animation is fluid, requiring high performance.

**How it will be tested:** We will examine this module's behavior by looking at the screen.

**Who Will Do It:** Laura

**Game Logic:**

**Inputs:** clock, random number, bet amount, position of center of mass of the red color, value of bet, amount money in pot

**Outputs:** states of the reels, lever position, amount of money in pot, value of bet

**Complexity and Level of Performance:** This can be a finite state machine since each reel has a finite state.

**How it will be tested:** We can output the values of the states and money amounts to the 64-bit hex display.

**Who Will Do It:** Daneaya

**Random number generator:**

**Inputs:** clock, enable

**Outputs:** random number, done

**Complexity and Level of Performance:** We will look up some random number generators online, and the performance should be reasonable.

**How it will be tested:** We will use the 64-bit hex display to see if we're getting out random numbers.

**Who Will Do It:** Laura

**External Components**

We will need to borrow a camera and a speaker from the lab. We will plug them into the board.

**Optional Functionalities**

We could make the game user-friendly by having it give the player on-screen directions. We can make the reels spin faster or slower in relation to the force of the pull-down of the lever. We could also make our game so that a player can place a bet by using on-screen buttons, instead of having to use the switches on the FPGA.

**Bibliography**

[1] [http://ivegot2haveone.co.uk/index.php?main\\_page=product\\_info&cPath=8&products\\_id=51](http://ivegot2haveone.co.uk/index.php?main_page=product_info&cPath=8&products_id=51)