6.111 Project Proposal
Partners: Becca Arvanitez and Cristina Domnisoru

Our project is a combination of live image processing, digital effects and interactive mini- games.

The general architecture of the project consists of two sets of modules, the effect modules and the game modules. Each effect module will take video input and generate real- time video output with a particular effect applied to the input. Each effect module will operate independently from the others and it will be possible to apply multiple effects to video by using the output of one effect module as input to another. Each effect module will be able to store image frames in memory and use them as references for processing. The effect modules will be able to send output to the screen in order to allow users to check what the effect does by itself or to send output to the game modules. The game modules represent separate games, with the user selecting which game should be played by using the switches on the FPGA. When a game is selected it uses the memory and camera input and issues controls to the effect modules in order to apply the game- specific effects to the input image. Each game module will process the flow of the particular game and overwrite data written to memory by previously played games.

Our goal is to have interesting and entertaining games that make use of the image of the person playing the game and the environment arround them and add a virtual gaming world to the image. The games and the vissual effects will be developed in conjunction with each other. In general, every game module will have its own FSM and helping modules to display the images required. The design of each game will be done independently and each game will be tested independently. Similarly, the effect modules ' design will depend on the individual characteristics of that effect, and will be tested independently.

Following is a discussion of particular digital effects and particular games we intend to implement.

There are several appealing digital effects that we are interested in. We decided to implement effects that have different degrees of difficulty , starting with the easiest. The easiest effect is taking the input and dividing the image into a set of puzzle pieces arranged randomly on the screen. Implementing this should be fairly straightforward. Every time this effect is enabled, it processes the input once and writes the output to memory.

There are several effects that we want to implement that we consider of medium difficulty. This category of effects is based on distinguishing between new and old objects. The effect, upon starting, will remember the current image and then compare each subsequent image to the initial image. Everything that wasn't there in the initial image will be somehow deformed by the particular effect. This would allow us to add static noise to a person that's walking across the screen or invert the colors of a person that appears and wasn't there when the game was reset etc.

The category of effects that we consider most difficult involve some form of motion processing and frame to frame analysis of the image data. We want to make an effect where new stationary objects get blended into the environment. The effect will remember the initial setup of the scene and then if a person walks in and keeps moving they will be displayed, but if they slow down they will become blended into the environment and if they are completely stationary they will effectively disappear off the screen. An alternative version of this is to analyze the color of the

current pixel and compare it to the respective pixel in the environment that is remembered and if the pixels are close enough in value then the environment is the one displayed. This would result in behavior such as , if one is wearing a white Tshirt over a white background only the background would be displayed resulting in a camouflage- like effect.
The most challenging effect that we intend to tackle involves motion- driven liquid-like alteration of the image. We are looking for effects such as sending liquid- like circular shockwaves that propagate across the image and deform the pixels they encounter accordingly. We intend to detect direction and speed of limb motion in order to generate these effects. The goal is that if a user waves their hand at certain speeds and with certain amplitudes (wide range intended), they will cause a deformation of the screen in a fashion corresponding to their movement. This effect requires more processing than the previous ones and will require frame – to frame motion analysis to detect not just the existence of motion but the direction, which involves at least some rudimentary version of object parsing.

We plan to implement three games, each of which integrates a different video processing effect into the gameplay. The first game we will implement is a game modeled after two- dimensional sliding tile puzzles. The game divides the camera- input video of the user into squares which are displayed in a mixed up order on the screen. The user moves the squares around until the puzzle has been put together. There is one blank square and picture squares may only be moved by sliding one tile at a time into the blank square.

A slightly more complicated game has the user catching falling objects to avoid catching on fire. Gameplay consists of the user watching themselves on the video screen with game- generated falling objects superimposed on the image. The user must move around and catch the objects, and if he misses objects fire effects are added to him on the video output. The fire effect added to the player is varying so that when a certain level has been reached it is game over.

The next game is two- player, where the users compete in a fighting game by throwing shock waves at each other. When a user gestures with their hands towards the other player, their motion generates shock waves which appear in the video. The shock waves will have varying magnitude depending on the speed of the user's movement, and will deal damage to the other player by making them look like they are melting. The amount a person has been melted will limit the magnitude of shock waves they can throw, and if a certain maximum amount of melting is reached then it is game over and the other player wins.

The main two parts of this project are the game subsystem and the effect subsystem. Becca will be working mainly on the game dubsystem and Cristina will be working mainly on the effect subsystem. Developing the game modules includes the FSM that controls gameplay, sends signals to interface with the effect modules, and decides what to do with the video which has been processed by the effect modules and returned to the game module.
Developing the effect modules includes processing the video sent from the game module to apply the specified visual effect. This means dividing the video into scrambled squares for the puzzle game and applying shock waves and melted effects to the video frame for the two- player game.

Separating the game FSMs and the video effects into separate modules makes it easier to test smaller parts of the system. This way each game module and effect module can be tested individually to find problems before integrating the modules.