6.111 Checklist
Rebecca Arvanites and Cristina Domnisoru

- **Puzzle game module**
  - FSM assigns locations for puzzle pieces of video from puzzle effects module, takes user input through buttons and applies game rules such as only switching a puzzle piece when adjacent to the blank square
  - Test gameplay: without effect module, use color blobs- module should allow user to switch puzzle pieces when adjacent to the blank square, stop game when puzzle is assembled in correct order

- **Fire game module**
  - Communicates to effect module the signals game_over and fire_parameter, how much fire effect to add
  - Communicates to blob_catch and blob_bomb modules the positions where objects should be displayed, and updates the positions to make the objects move
  - Communicates to hit module position to check for intersection of the player and the game blobs, receives true or false hit signal from hit module
  - Test gameplay: player will be able to interact with game-generated moving blobs superimposed onto the camera video, which interact with the player upon contact with the player’s control object (colored rectangle detected by the hit module)

- **Blob_catch module**
  - Works with fire game module: generates pixels of the blobs which the user should catch
  - Sends blob_catch pixels to module which merges all fire game pixels to be displayed
  - Test: module can display blob objects onscreen

- **Blob_bomb module**
  - Works with fire game module: generates pixels of the bombs which the user should avoid
  - Sends blob_bomb pixels to module which merges all fire game pixels to be displayed
  - Test: module can display blob bombs onscreen

- **Hit module**
  - Computes whether a blob collided with a patch of a particular color range (the color is specified as input)

- **Puzzle effect**
  - Takes as input positions of the squares in the puzzle
  - Processes pixels from RAM and outputs to memory the new pixels
  - Each square should end up at the position specified as input

- **Fire effect**
  - Takes as input the amount of fire to be added
  - Processes pixels from memory and outputs to memory the new pixels with the added fire effect

- **Input manager**
- Takes data from camera, processes data stream, generates appropriate pixel signals (all pre-written modules)
  - Converts ycrcb to rgb and delays pixel signals
- Output manager
  - Generates pixel signals for display
  - Displays data from memory to monitor
- Memory manager
  - Takes rgb data and pixel signals
  - Provides previous and current pixel data for specific coordinates to Effect modules
  - Generates data stream to be written to memory
- Buffer swapper
  - Controls which buffer is being written by camera and read by effect modules
  - Controls which buffer is being written by effect modules and read to monitor
- Effect manager
  - Controls which of the effects’ pixels and addresses are actually communicated to memory. Effectively swaps between different effects.

Time Permitting:

- Shockwave game module
  - Specifies for effect module the location to apply shockwave and how much of the effect to apply, also the amount each player should be melted by the effect module
  - Test: testbench waveform to see if module is behaving as expected
- Shockwave effects
  - One effect for wave-like deformation
  - One effect for swirl-like deformation
  - One effect for droplet-in-water deformation