# 3D City

## Project Checklist

Grace Cassidy and Khalil Elbaggari

### Commitment:

- (Khalil) Rendering Block: This block will calculate the 2D location of a 3D point to be displayed on the screen. Tested by making sure the returned value read from the display is similar to what we'd expect to see on a screen.
    - Counter Module: A simple FSM that feeds all object's (edge or polygon) IDs and vertices to be transformed and passed to the frame buffer. There is a "vertex done" wire to keep processing speeds high. "Done" signals need to be passed through the transform module to keep things synchronized.
    - Cosine Module: Computes the cosine of a 8 bit number (actually 10 but symmetry's used returning a 12 bit number) with an error of ~0.1% using shifts, 3 multiplications, and two subtractions.
    - Sine Module: A shifted cosine module.
    - Transform Module: Apply the transformation matrix to the vertex to go from 3D to 2D.
- (Khalil) Input Block: This takes the inputs of the LabKit's buttons and adjusts the position of the user with time. Tested by outputting position values and triggered sound IDs to the display and LEDs respectively.
    - Movement Module: This changes the user's location over time based on the inputs. With the labkit, a small FSM will handle requests. There will also be a partial step counter (8 bits) so that movement angles are implemented correctly.
    - Sound Lookup Module: This will trigger the appropriate sounds if the distance to a set of vertices falls below a certain value.
- (Grace) Frame Buffer: This module will read and write to the ZBT SRAM. After reading from RAM, it will display the data on an XVGA display.
- (Grace) Line Drawing Module: This module will take the data points from RAM and draw the lines on the VGA display for the wireframes.

### Goal:

- (Grace) Sound Module: This module will store pre-recorded sounds in the ROM and will send the audio to the AC97 chip and then to the speakers or a headset.
- (Khalil) Polygonal Buildings: This would be an adjustment to both the rendering module and the frame buffer. It will display the filled-in buildings rather than the wireframes.

### Stretch:

- (Grace) Z-buffering: This would be an addition to the frame buffer and would allow some faces of buildings to be hidden if they are covered by other polygons. This would also enable shading to the buildings.
- (Khalil) Polygon shading: This module determines the color of the polygon based off angle the face forms with the light source. This requires vector manipulations so making it fast will be difficult.
- (Khalil) Polygon culling: This module would sit before the rendering block to check the two cases of when a polygon shouldn't be rendered. The first is when the polygon's "pointed" away from the user such that it's hidden from view and the other is when the polygon's sufficiently out of the field of view. This will save rendering time since these polygons don't change the scene.
- (Grace) Acceleration/Velocity Control: This would allow the user to change acceleration and velocity to simulate jumping and running in the virtual city.
- (Khalil) Distributing slower tasks: This will be difficult to keep things in order since processing times may be variable. For the 3D to 2D transform, keeping a vertex's absolute position in the list of vertices associated with its coordinates can keep them in order