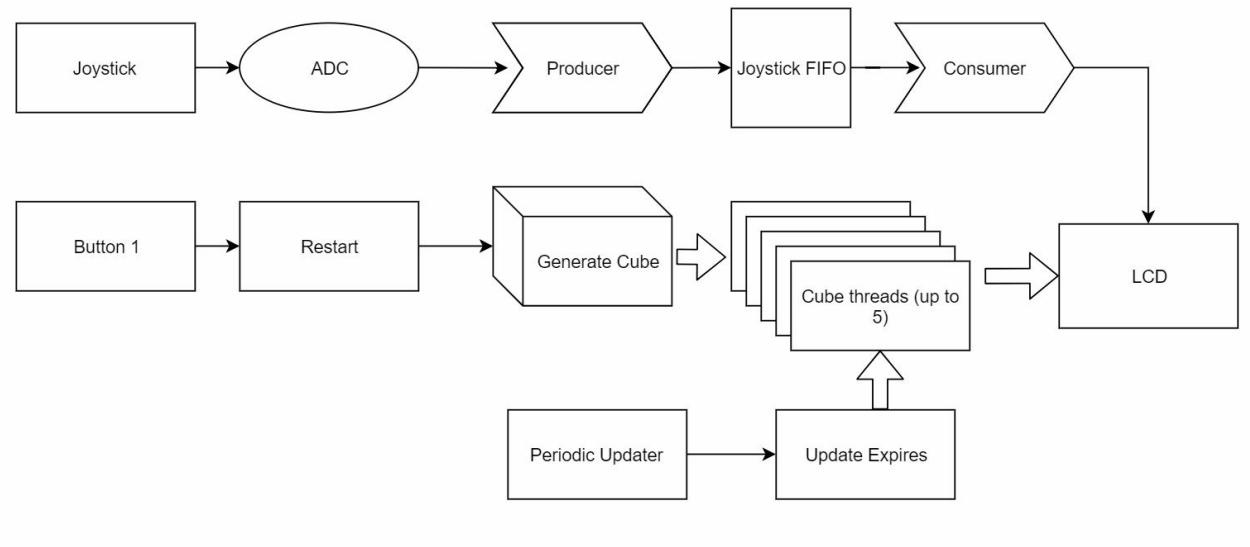
Part 1 Deliverable
Advanced Embedded Final Project Group Eta
Navid Ahmed, Talis Basham, Trey West, Max Poe



Data Flow Diagram

Deadlock Prevention

The deadlock prevention checks if the cube would move into the space of another cube and if it would randomly selects a different direction as it would next to a wall. This is accomplished by checking the value of the semaphore corresponding to the cube in that direction. An additional requirement is that another semaphore is needed while the cube is checking direction to ensure that the board doesn't change after a thread checks an adjacent cube but before it gets a lock on that space.

The movement code is blocked out by a semaphore. In between the following is the entirety of the movement code.

```
OS_bWait(&criticalTest); OS_bSignal(&criticalTest);
```

However, that alone is not enough to prevent deadlock, it additionally tests if the next space in its current direction is blocked, and if it is picks a random direction

```
else if (thisCube->dir == 1) {
  // If we've hit west wall, pick new direction
  if(thisCube->x == 0
     || !(BlockArray[thisCube->x-1][thisCube->y].BlockFree.Value)) {
    do {
      thisCube->dir = Random(4);
    } while(thisCube->dir == 1);
  }
```

Pseudo-Random Number Generator

A Lehmer Random Number Generator was used. It is a special case of linear congruential generator where the value is multiplied by a constant then the remainder of division by another

constant is taken to get the next value of the generator. To map this value into an arbitrary specified range, the remainder of division by the range is taken. Although this does not give a uniform result, it is “good enough” for use in a game. Here is the code including the unused additive parameter.

```
// Return random value from 0 to range-1
// using Linear Congruential Generator
unsigned int Random(unsigned short range)
{
  unsigned int m = 2147483647;
  unsigned int a = 16807;
  unsigned int c = 0;

  Xn = (a * Xn + c) % m;
  return Xn % range;
}
```

Team member contributions:

Navid:

- Setting up repo
- Struct for Cube
- Cube Thread method

Talis:

- Deadlock Prevention
- Pixel Perfect collision
- Restart
- Writeup of RNG and Deadlock Prevention

Trey:

- Pseudo-Random Number Generator
- Cube Movement
- Wall detection
- Cube detection

Max:

- Flow Chart
- Debugging
- Report
- Video