6.111 Project Checklist
Asteroids
Richard Agbeyibor & Phillip Mercer

During our checkoff we will demonstrate the functionality of the following modules:

□ **Ship** (Richard)

The ship module controls the movement of the space ship in the game. The player can move the ship using the left rotate, right rotate, and thrust buttons. This module outputs the ship's position, its heading and the position of the gun turret. The movements of the ship are modeled by motion equations that determine its acceleration and velocity in space.

During the checkoff, we will demonstrate the movement of the ship across the screen in response to user input. The ship will accelerate in the direction of the current heading, and wrap around space when it reaches the edge of the screen.

We are approaching design of the ship in incremental steps. To test functionality of the movement logic, the ship will initially be in the shape of a circle. Once that is completed, I will implement the original triangular shape which has proven more complex than anticipated.

If there is additional time in our schedule, we will demonstrate hyperspace movement, which is when the ship disappears and reappears in a new random location on screen.

□ **Bullets** (Mercer)

The bullets module controls the behavior of bullets shot by the space ship. Bullets emanate from the ship and move across the screen until they collide with an asteroid or reach the edge of the screen. The user controls the direction and timing of bullets using the rotate and fire buttons.

During Checkoff, we will demonstrate the ship firing a round of magazines with various headings. If all 10 bullets in the magazine are on screen, the user is no longer able to fire a shot.

□ **Asteroids** (Mercer)

The asteroids module controls the basics of the asteroid such as initial position, speed and size as well as the movement of the asteroid. Each asteroid will be controlled by its own instantiation of this module.

This module will demonstrate a "random" selection of initial location and size, controlled by an internal variable local to the specific instantiation. The range of speeds will be determined based on the size of the asteroid such that smaller asteroids are able to move faster than bigger asteroids. Each asteroid will move independently of the other asteroids on screen and will be terminated if it should move off the screen.

If time permits, the asteroids will be able to splinter into two asteroids of the next smallest size as opposed to splintering on relevant collisions. We may also use stored images to create more interesting asteroid shapes, though this may make the splintering difficult or not feasible.

□ **Game Logic** (Mercer)

The Game Logic module will be responsible for determining when relevant collisions occur. Relevant collisions will be considered any collision between asteroids and bullets or asteroids and the ship.

This module will demonstrate recognition of all relevant collisions. After such collision, the involved objects will disappear from the screen.

□ **Sound** (Richard)

The Sound module will be responsible for loading sounds from BRAM and sending them through the onboard AC97 for playback. The most recent sound shall take precedence over the playing of any older sounds.

The demonstration of this module will be the playing of sounds that have been loaded to BRAM on collisions.

If time permits we will be able to incorporate more sounds into the gameplay making the game more interesting.

□ **VGA Display** (Richard)

The VGA module will be responsible for displaying the running game and score. The VGA will also be responsible for displaying the user’s score on screen. This will be accomplished by reading stored images of the ten digits and stringing them together as necessary. This will also be the demonstration of the functionality of the module. Should time permit this module will incorporate more interesting animation on relevant collisions.