Application Pseudo-Code

* Initialize module
  + - Initialize registers and memory
      * + Starting ship object (info to memory)

Health

Ammo

Lives

* + - * + Make asteroids objects (info to memory)

Set MAX # of asteroids allowed on field

Set size = random (out of 3-4 different sizes)

Start size determines subsequent sizes after it is blown up

Start size determines strength (how many hits to blow up)

* + - * + Calculate I/O addresses
* Physics
  + - * + Up

Increment x value

* + - * + Down

Decrement x value

* + - * + Left

Decrement y value

* + - * + Right

Increment y value

* + - * + A

Shoot Bullet

* + - * + Asteroid

Random paths

* Bullet
  + - Create bullet
      * + Starting position is (x,y) of ship
        + End point is edge of virtual field (i.e. we can display a smaller portion of the actuall field on which we play . This allows for off-screen shooting and blowing up asteroids)
        + Sound byte?
        + Check for hit

Hit = (bullet.endpoint == asteroid.location)

* + - * + If(Hit)

Explode(asteroid @ bullet endpoint);

Return;

* + - * + Else

Return;

* Explode (asteroid)
  + - Sound byte?
    - Clear current asteroid
    - If (Asteroid.size == MIN)
      * + Score += 200;
    - Else
      * + Create (Size - 1) new asteroids of size -1
* Menu Loop
  + - Load menu to screen
    - Wait for User Input to START or QUIT, etc.
* Game Loop (Once START is asserted)
  + - Send ship to screen
    - Send asteroids to screen
    - Begin physics
    - Begin game
    - While(player)
      * + Updatescreen();
    - Final score screen
    - Wait for User input for RESTART, QUIT, etc.
    - return