Idea notepad

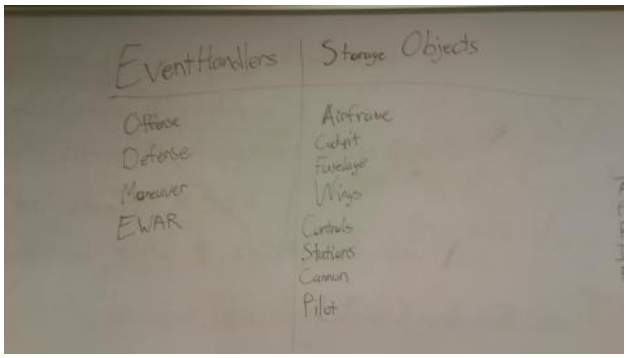
Wednesday, January 31, 2018 8:34 AM

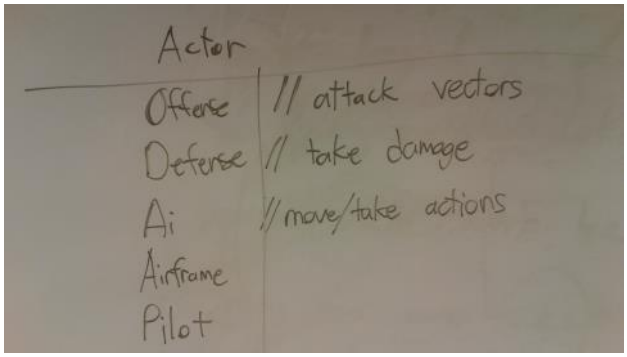
Camera that moves player?

make test map

2/3/2018

Saturday, February 3, 2018 5:39 PM





What Does Everything Do?

- Effects
  - Bonuses to Pilot or Airframe
  - Penalties against Pilot or Airframe
- Event Handlers
  - Handles the events called by players and NPCs
    - Offense
      - Make attack rolls
      - Expend weapon stores
      - If hits are successful, call Defense
    - Defense
      - Handles exchange of damage
    - Maneuver
      - Handles plane movement across
    - EWAR
      - Handles interference with other aircraft
      - Handles Effects against Pilot
      - Handles Effects against Airframe
    - Intercept
      - Handles movement target
      - Missile moving to NPC
      - NPC moving to patrol path
      - Player moving to steering path
- Storage Objects
  - Airframe
    - Cockpit
      - Armor
        - Hit Points
        - Once this number is exceeded, a part is damaged in the location. If the part receives damage again, that part is destroyed and no longer functions
      - Active Radar
        - Forward Radar
        - Long Range
        - Used to fire missiles
      - Passive Radar
        - Omni-Directional
        - Shows what is looking at you with their own radar
        - Relative Ranges (Far, Long, Short, Close)
      - Radio
        - Allows communication with friendlies
        - Can allow use of special abilities
      - Instruments
        - Allows plane to follow flight paths
        - Allows pilot to make certain maneuvers
        - Makes certain maneuvers not dangerous
      - Flight Control
        - Allows Pilot to control plane
        - If damaged, special maneuvers forbidden
        - If destroyed, Pilot must eject
    - Fuselage
      - Fuel Tank
        - Stores fuel for plane

        - May be reloaded mid-flight
        - If damaged, leak fuel from tank
        - If destroyed, airframe is also destroyed
    - Armor
        - Same as cockpit
    - Belly Stations
        - May hold certain munitions
        - Some munitions are Belly stations only
    - Fuel Stations
        - May hold a drop tank
        - Works like a fuel tank
        - Drop Tanks impose maneuvering penalties
    - Cannon
        - Dakka dakka
        - Allows direct engagement at point blank to close range (inside the arming range of close-range missiles)
        - If damaged, lowers number of possible hits
        - If destroyed, may not fire
- Wings
    - Armor
    - Wing Stations
        - Wing-tip stations
        - May take limited munitions, typically air to air or ECM related
        - If damaged, may not fire properly
        - If destroyed, may not fire at all
        - Does not impose flight penalties when loaded
    - Pylon Stations
        - Holds certain munitions
        - Imposes flight penalties while loaded
        - If damaged, same as wing stations
        - If destroyed, same as wing stations
    - Fuel Tanks
        - Same as fuselage fuel tanks
    - Cannon
        - Same as fuselage cannon
- Controls
    - Surfaces
        - Allow the plane to roll, pitch, and yaw
        - If damaged, penalties imposed
        - If destroyed, one of those is lost (significant penalties)
    - Engines
        - Allows the plane to go fast
        - May exceed parameters temporarily with afterburner
        - If damaged, no AB and increased fuel use
        - If destroyed, plane must glide
    - Canards
        - Improves maneuvering
        - If damaged, imposes speed penalty
        - If destroyed, imposes speed and maneuvering penalties
        - May substitute for damaged surfaces if present on the plane
    - Armor
    - Landing Gear
        - Allows plane to take off and land
        - Determines the types of airfields to land on (including carriers)
        - If damaged, landing is a crash based on Pilot skill
        - If destroyed, most severe crash is the most likely event (bad crashes can kill the pilot)
- Effects

        - Bonuses
            - Burn less fuel
            - Gain more altitude
            - Temporarily raise altitude and speed limits
        - Penalties
            - Burn more fuel
            - Shorten radar range
            - Temporarily add altitude or speed restrictions
        - Environmental
            - Cannot be removed during missions outside of certain circumstances
            - Not applied by player or NPC
    - Pilot
        - Skills
        - Talents
        - Effects

PicoArmor Punchlist

Monday, February 5, 2018 1:43 PM

- A-10 x 2
- A-1H x 2
- A-4F x 2
- A-6A x 2
- A-7 x 2
- AV-8B x 8
- C-130 x 2
- F-100D x 2
- F-105 x 2
- F-106 x 2
- F-117A x 2
- F-14A x 6
- F-15A x 6
- F-16 x 6
- F-22 x 2
- F-35A / C x 2
- F-4B x 6
- F-4E x 6
- F-5E x 6
- F-8 x 4
- F/A-18 A / C / E x 6
- O-1 & O-2 x 1
- Harrier GR 3 x 2
- Hunter F 6 x 2
- Lightning F 6 x 2
- Sea Harrier FRS 1 x 6
- Q-5 Fantan x 6
- JH-7A x 6
- J-10A x 4
- J-20 x 2
- Jaguar A x 2
- Mirage 2000 x 2
- Mirage F1C x 6
- Mirage IIIC x 6
- Rafale M x 6
- Tornado IDS x 2
- IAI Dagger x 6
- Kfir C2 x 6
- Eurofighter Typhoon x 2
- Saab AJ37 x 2
- Saab J35F x 6
- Saab JAS39 x 6
- MiG-15 x 2
- MiG-17 x 2
- MiG-19 x 2
- MiG-21MF x 12
- MiG-23 x 2
- MiG-25 x 4

- MiG-27 x 2
- MiG-29 x 6
- Su-15TM x 2
- Su-17 x 2
- Su-24 x 2
- Su-25 x 2
- Su-27 x 6
- Su-30 x 6
- Su-33 x 6
- Su-34 x 2
- Su-7 KBL x 2
- WARPAC Helicopter x 1
- US Cold War Helicopter x 1

Classes

Saturday, February 10, 2018 12:27 PM

Location
- int x
  - x-axis location on the grid
- int y
  - y-axis location on the grid
- int alt
  - altitude of the object
- int h
  - heading of the object (if applicable)

Render
- TCODColor col (if applicable)
  - color to render the ASCII character as
- int ch
  - ASCII character to be rendered

AI
- int AITYPE
  - what kind of AI does this Actor have, if any

Engine
- Map * map
  - holds our map attributes

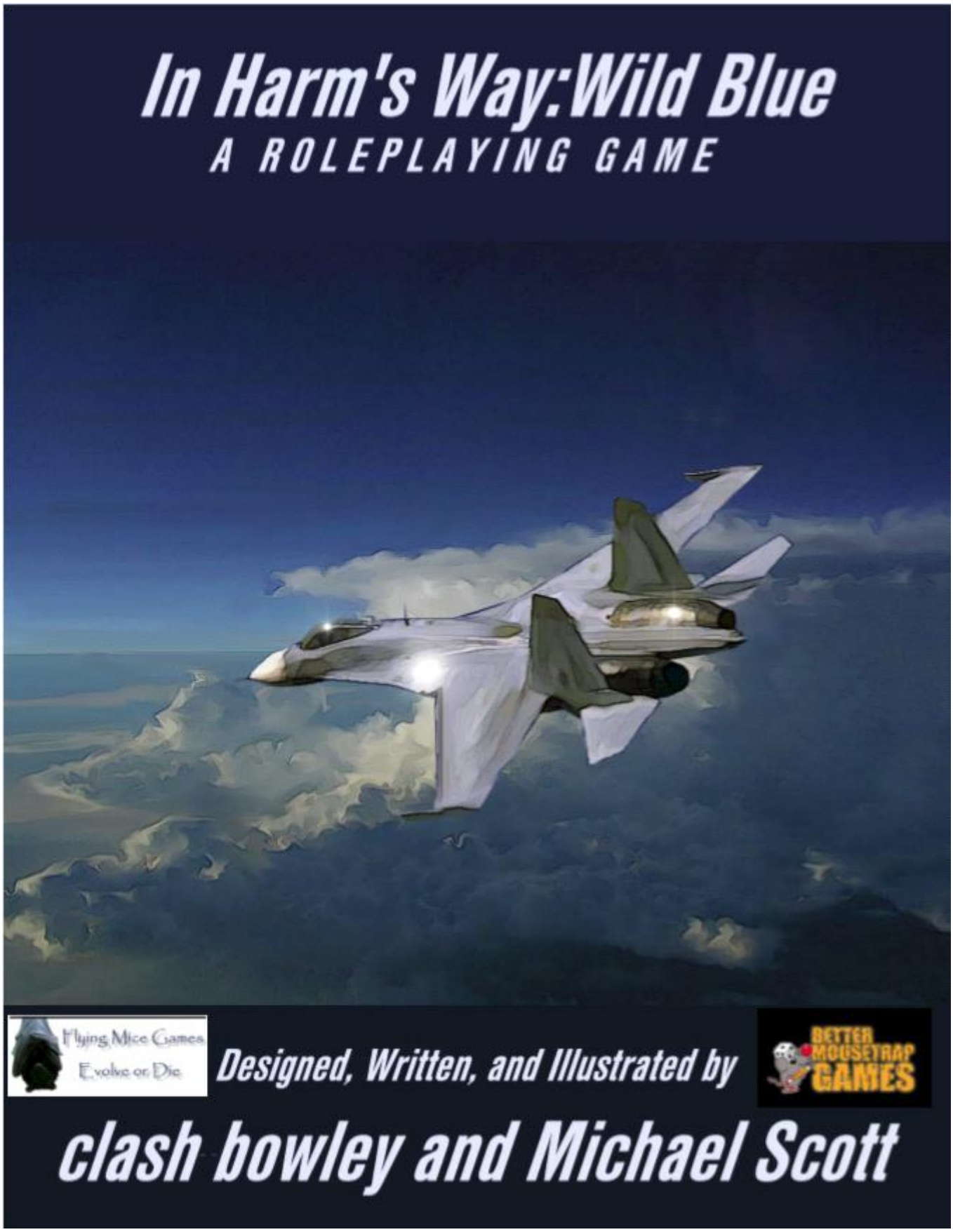
Map
- Tile ** tiles
  - 2d array that holds the tiles

Tile
- Location * loc
  - holds location information of the tile, may be redundant?
- Render * rend
  - Info needed to render the tile
- bool bounds
  - is this tile a boundary, and therefore not walkable?
- bool guidestone
  - is this tile a waypoint Guidestone?

Actor
- Location loc
  - Where is the actor
- Render rend
  - What does it look like onscreen
- Pilot p

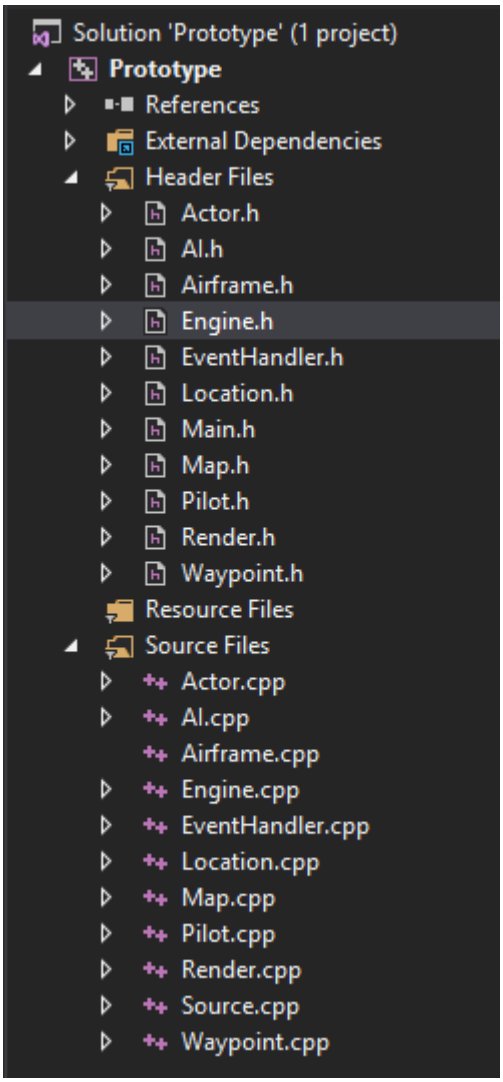
    - Who is driving it (if applicable)
    - holds stats for attack rolls
    - holds bonuses and penalties to these rolls
- Airframe a
    - What is being driven
    - Holds all the flying parts
- AI ai
    - Holds type of AI
        - Player
        - Friendly
        - Neutral
        - Hostile
        - Missile

Wednesday, March 7, 2018 1:18 PM



3/7/2018 1:18 PM - Screen Clipping

Wednesday, March 7, 2018 1:24 PM



3/7/2018 1:24 PM - Screen Clipping

Wednesday, March 7, 2018 1:24 PM

Actor.h* Main.h Map.h EventHandler.h Engine.h

Prototype (Global Scope)

```
1   #pragma once
2
3   #include <stdio.h>
4   #include <string>
5   #include <string.h>
6   #include <vector>
7
8   enum Status
9   {
10      Defensive,
11      // NOTHING HIGHER THAN THIS I SWEAR TO GOD ANDREW
12
13      // NOTHING LOWER THAN THIS I SWEAR TO GOD ANDREW
14      Patrol
15  };
16  class Actor;
17
18  #include "libtcod.hpp"
19
20  #include "Location.h"
21  #include "Render.h"
22
23  #include "Map.h"
24
25  #include "AI.h"
26  #include "Airframe.h"
27  #include "Pilot.h"
28  #include "Waypoint.h"
29
30  #include "Actor.h"
31  #include "EventHandler.h"
32  #include "Engine.h"
```

3/7/2018 1:24 PM - Screen Clipping

Actor.h* Main.h Map.h EventHandler.h Engine.h

Prototype Map

```
7   class Tile
8   {
9
10  private:
11
12  public:
13      /*
14          Tile holds properties of the particular square in any given map. This includes;
15          - Location of the Tile
16          - List of pointers to the actors on the Tile
17          - if the Tile is a map boundary or not
18          - if the Tile is a Guidestone waypoint or not
19          - ASCII character of the background (should there be one)
20          - TCODColor code to render
21      */
22      Location loc; // location of tile, may not need?
23      // std::vector<Actor *> contains;
24      bool bounds; // is this tile a boundary or not?
25      Render rend; // render info for the tile
26      bool guidestone; // is the tile a guidestone waypoint or not
27
28
29      Tile(int x, int y, bool edge);
30      Tile();
31      ~Tile();
32
33      void RenderTile() const;
34      void tRenderVals(Render * rend);
35      void setTile(int x, int y, bool edge);
36
37  };
38
39  class Map
40  {
41  private:
42
43  public:
44      Tile ** tiles;
45      std::vector<Tile> guidestones;
46      int width, height;
47      TCODMap * pMap;
48
49      Map(int width, int height);
50      Map();
51      ~Map();
52      void mRender();
53      void makeGuidestones(int num);
54  };
```

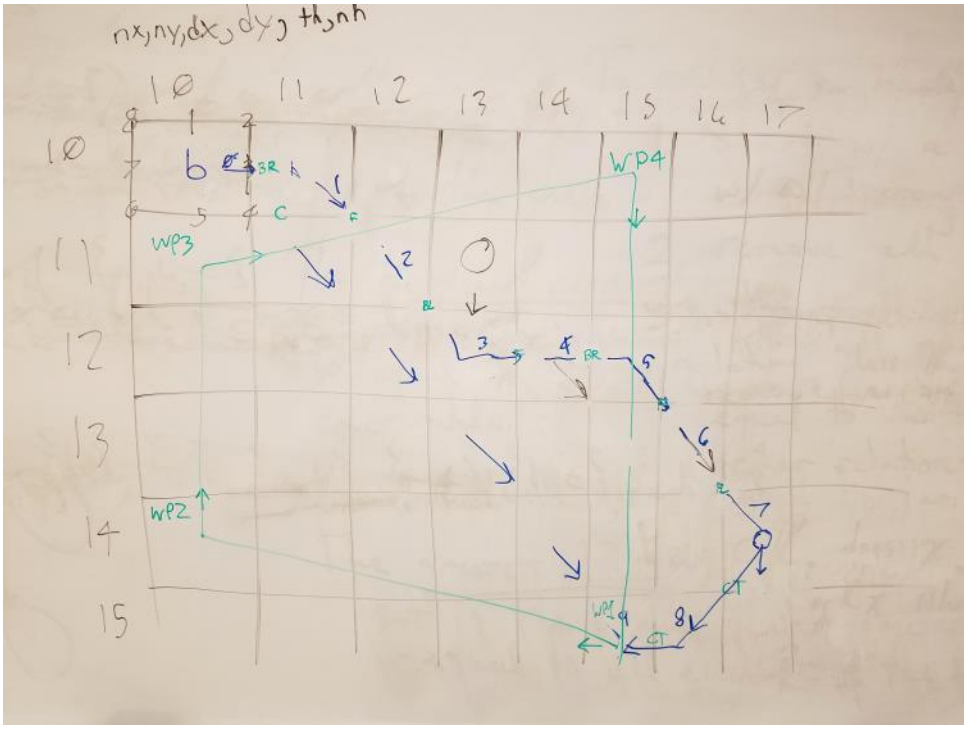
Engine.cpp | Waypoint.cpp | Location.cpp | Airframe.cpp | Actor.cpp | Pilot.cpp

Prototype → Engine → Engine()

```cpp
#pragma once
#include "Main.h"

Engine::Engine()
{
    // init game window
    TCODConsole::initRoot(80, 50, "Prototype", false);

    // init map
    map = new Map(80,50);
    // add guidestones here

    // init player
    player.loc = new Location(40, 25, 10, 1);
    player.ai.AITYPE = 1;
    player.rend.ch = '@';
    player.rend.col = TCODColor::white;
    // player.tp = &map->tiles[player.loc->x][player.loc->y];

    /*
        MISSION STUFF HERE
        -   eventually, all of these will have nice clean functions so it doesn't look like
            I hacked this nonsense together so I could go home early to pet my dog and punch rats.
        -   Mission consists of:
            - Objective
            - Guidestone Waypoints
            - NPCs (Hostile, Friendly, Neutral, etc.)
            - Weather? (I mean, a cloud is an actor I can't kill, right?)
    */

    // make guidestones
    TCODRandom * grand = TCODRandom::getInstance();
    map->makeGuidestones(grand->getInt(4, 8, 4));

    // make pods
    Actor tempBandit;
    tempBandit.loc = new Location(10, 10, 10, 3);
    tempBandit.ai.AITYPE = 3;
    for (int x = 0; x < map->guidestones.size(); x++)
    {
        tempBandit.ai.waypoints.push_back(map->guidestones[x].loc);
    }
    tempBandit.ai.path = new TCODPath(map->pMap, 1.60f);
    /*tempBandit.ai.path->compute
    (
        tempBandit.loc->x,
        tempBandit.loc->y,
        tempBandit.ai.waypoints[tempBandit.ai.wpt].x,
        tempBandit.ai.waypoints[tempBandit.ai.wpt].y
    );*/
    tempBandit.rend.ch = 'b';
    tempBandit.rend.col = TCODColor::lightRed;

    aList.push_back(tempBandit);
}
```

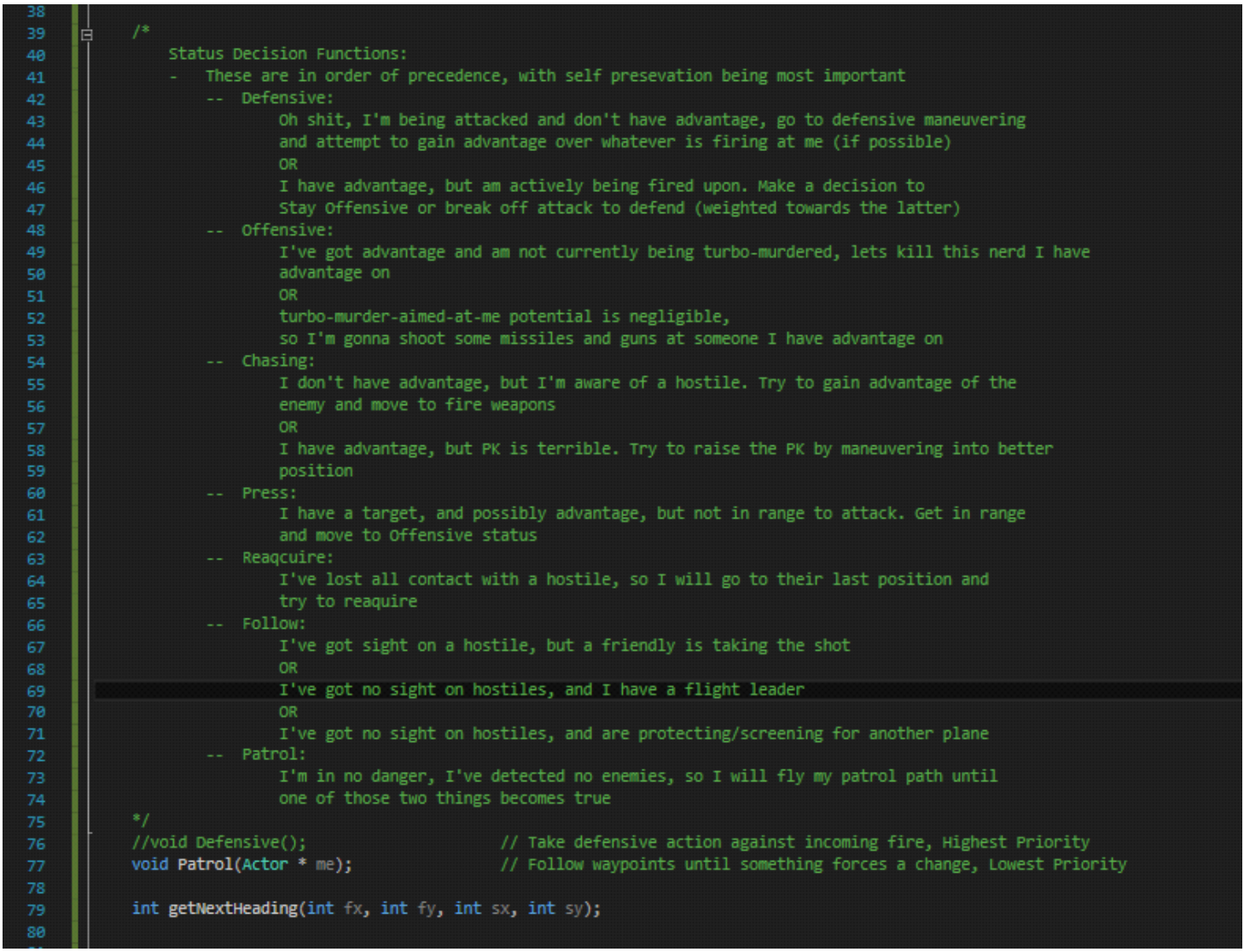
3/7/2018 1:26 PM - Screen Clipping

Wednesday, March 7, 2018 1:54 PM



Wednesday, March 7, 2018 1:55 PM

Wednesday, March 7, 2018 1:59 PM



```
/*
    Status Decision Functions:
    -   These are in order of precedence, with self presevation being most important
        --  Defensive:
                Oh shit, I'm being attacked and don't have advantage, go to defensive maneuvering
                and attempt to gain advantage over whatever is firing at me (if possible)
                OR
                I have advantage, but am actively being fired upon. Make a decision to
                Stay Offensive or break off attack to defend (weighted towards the latter)
        --  Offensive:
                I've got advantage and am not currently being turbo-murdered, lets kill this nerd I have
                advantage on
                OR
                turbo-murder-aimed-at-me potential is negligible,
                so I'm gonna shoot some missiles and guns at someone I have advantage on
        --  Chasing:
                I don't have advantage, but I'm aware of a hostile. Try to gain advantage of the
                enemy and move to fire weapons
                OR
                I have advantage, but PK is terrible. Try to raise the PK by maneuvering into better
                position
        --  Press:
                I have a target, and possibly advantage, but not in range to attack. Get in range
                and move to Offensive status
        --  Reaqcuire:
                I've lost all contact with a hostile, so I will go to their last position and
                try to reaquire
        --  Follow:
                I've got sight on a hostile, but a friendly is taking the shot
                OR
                I've got no sight on hostiles, and I have a flight leader
                OR
                I've got no sight on hostiles, and are protecting/screening for another plane
        --  Patrol:
                I'm in no danger, I've detected no enemies, so I will fly my patrol path until
                one of those two things becomes true
*/
//void Defensive();                     // Take defensive action against incoming fire, Highest Priority
void Patrol(Actor * me);                // Follow waypoints until something forces a change, Lowest Priority

int getNextHeading(int fx, int fy, int sx, int sy);
```

3/7/2018 1:59 PM - Screen Clipping