The game file also contains the following folders and files:

TEAMFOLDER = "./saveFiles/teams/"

TEAMLIST = "./saveFiles/teams/teamList.txt"

GUNREF = "./data/guns.txt"

ARMORREF = "./data/armors.txt"

DATAFILE = "./data"

SAVEFILE = “./saveFiles”

SPACELOCATION = "./data/images/Space.bmp"

UPDATE INFO:

9:43 AM, DEC 24: Added augments to gear

10:11 AM, DEC 24: Added nameCheck to fileSystem (not ‘plugged in’)

Added a main file and a test module.

10:43 AM, DEC 24: Debugged fileSystem

3:19 PM, DEC 24: added augmentFunctions, modified some stuff in gear

1:08 AM, CHRISTMAS!!!: I figured out how sprites are going to work...

11:36 AM, DEC 26: Added some drawing functionality, such as drawing along a path and

changing what direction a sprite is facing along said path. It’s pretty neato.

9:28 PM, DEC 26: added method adjacentTileCoords to board

2:01 PM, DEC 27: debugged drawing and calibrated frame speed stuff.

12:12 PM, DEC 30: debugged a little, made some code cleaner, added some blank functions

1:13 PM, JAN 2: added some sprite functionality into fighters. Messing with groups.

4:00 PM JAN 10: some work was done on being able to draw a board.

8:41 PM JAN 10: Updated the trunk with all the junk I’ve got... Junk in the Trunk.

Sprite Skins:

There will be “hot points” on the armor typically colored FF00FF (ugly pink) that determine color. These are to determine team. There will be other hot points of other colors (presumably FFFF00 and 00FFFF) respectively) for guns and armor. They will represent extensions of the armor.

Ammo Types: (guns have one set of behavior, determined by the ammo they carry)

poison, fire, freeze, acid, time distortion, projectile, beam, armorbane, fleshbane, concussive, blast, stream, health, sonic, spray, nano, spike, persistent, cloud, proximity, explosive, unstable, mind control,

Armor Types: (ammo also have one set of behavior, determined by... it’s generation)

normal, jet, stealth, unstable, powered, dragon, healing, regenerating, combat drug, mind control, agility, reflective,

Possible Names:

Starbuckler, Spacebuckler, Cosmic Privateers, Spacecraft, Cosmocraft, Astrocraft, Pirate Space, Space Pirate, Space Raiders, Cosmic Raiders, Astrobuccaneers, Warlord, Pirate Lord

A GIF to give an idea of how the animation should look; this is preliminary and still has some issues, but it should be solvable.

<http://makeagif.com/uzG3RH>

<http://code.google.com/hosting/createProject>

Some info about loading images in pygame: <http://www.pygame.org/docs/ref/image.html>

Thank god i found this: <http://shinylittlething.com/2009/07/21/pygame-and-animated-sprites/>

Tom,

So I tried out Wesnoth. I do not think it is quite what we are going for, but it’s close. Certainly download it and try the tutorial. More than anything else, our UI can easily be better than Wesnoth’s. Our sprites have will have full animation among other things, and their upgrade system is pretty simplistic compared to what we’re envisioning. I think its biggest problem is that I have a ton of trouble just telling what unit I have selected, and the whole thing works with left clicks (as opposed to right).

That said, there are some things that Wesnoth does that I would like to steal, like the way villages work. I don’t want to have “settlements” per say, but have at least strategic points on that map that need to be captured. I was thinking things along the line of computer control (eg. turns off self destruct, allows control of automated defenses, security cameras), med bay(s), control panels that open doors, doors, etc. I will make a list of possible special tiles.

Their attack system is pretty simplistic too; it’s just percentages based on tile type. So I think my point is that we can do better. Try it out; it’s actually kinda fun, but It’s also really simplistic. When you play, build archers. They seem a lot better than melee troops.

Oh, also, I think i can do better walls.

Some notes on the file system for tiles:

For every skin set, put it into one file with the name starting with “tile” then the skin type, no spaces. (eg, tileCement) The rows of the file will be different kinds of tiles, and the columns will be different frames of animation (these can be whatever size you like)

For walls, put them into one file with the name starting with “wall,” then skin type.

There will be several sections of equal size (for every file) for different kinds of walls in the skin set. For each section, the rows of this file will be the direction the wall is facing, starting from the top and going counterclockwise. The columns will be different phases in the wall’s destruction.

The sections themselves will be placed into rows and columns. The rows designate a difference in functionality (Eg, tall walls vs. short walls), and the columns will designate different designs. (Eg, a tall wall that’s blank vs. a tall wall with a sign on it), Each drawn wall should be put into a rectangle the size of a tile. Please only draw the bottom 3 walls, and order the columns left, middle, right.

We will have separate files for raised tiles (or tiles with objects on them), since they will be a different size than a typical tile, so parsing will be hard. (we might want unique designs as well.) Those images will be in a file with the name starting with “raisedTile” followed by the skin type. The rows and columns will function the same way as the regular tiles.

Ted: On the tiles you drew, frame six has a slight difference from the rest along the middle of the top left edge (it’s a single pixel, but it flickers noticeably)  
Also, tile type 7 (indexing from 1, black and yellow stripes) has differences in different frames.

Another thing: Let’s change the format for tiles, separating them into animated tiles and stationary tiles. Since we are doing that, we need a new convention for putting all of them in the same file. I believe that the easiest thing would be to say that the first column is all stationary tiles, and then the columns after that will be reserved for animated tiles and their animation. Don’t worry about the width and height of the file, as long as for each tile set the animated tiles animate over the same number of frames (meaning, if we want, different tile sets can have more or fewer frames). I can handle the rest. (A side note: we can also make it so that different tiles have different number of frames within the same skin set. If we want to do that, we need to talk about conventions for making this possible (e.g. an all blank square with a “magic pixel” or an all ugly blue square (the easier option) or something along those lines to mark sections that should be ignored.))