**Metanact – Informal Programming TODO**

* ~~Remember how the heck I put together my older 2D engine~~
* ~~Write a console demo that traverses files+folders programmatically~~
* ~~Strip out unneeded functionality from the old engine~~
  + ~~Networking~~
    - ~~Remove Client/Server specific routines, replace with generic~~
    - ~~Get projectiles working again~~
    - ~~Remove awareness of other players~~
    - ~~Remove network library dependency~~
* ~~Get directory traversal working in new engine~~
* ~~Fiddle with randomness constants~~
* ~~Add determination of spawned entities based on underlying file size~~
* ~~Add logging functionality to the video window called from a universal function in App2D~~
* ~~Modify Schnommus logo to suit the game~~
* ~~Create Metanact logo~~
* ~~Start moving away from vector shape definitions~~
  + ~~Add an IMAGE tag for entities~~
  + ~~Create new player sprite~~
* ~~Allow player to move around more than the initial directory~~
  + ~~Create a special wormhole allowing the player to move up a directory~~
* ~~Add a fading splash screen~~
* ~~Create some resource-handling classes for external loading and to prevent memory leaks~~
  + ~~Font management~~
  + ~~Sound management~~
  + ~~Music management (specialised sound as only 1 can play at a time)~~
  + ~~Image management~~
* ~~Impose a limit on the number of files + folders that can be loaded in a level at once~~
  + ~~Pick randomly form a list?~~
* ~~Add a loading screen in between levels~~
* ~~Add some sort of background effect~~
  + ~~‘Stars’ made of zeros and ones?~~
* ~~Overhaul the project file system so it’s nicer~~
* ~~Draw some storyboard diagrams for the menu system~~
* ~~Create an engine particle effect for the player’s ship~~
* ~~Create a simple enemy type (grunt)~~
* ~~Create a menu system~~
  + ~~Draw/implement architecture~~
    - ~~Item type with option attributes~~
    - ~~Item type without option attributes~~
    - ~~Drawing is completely generic; title an attribute of screen class~~
    - ~~[System->Screens->Items->Attributes]~~
  + ~~Feed into start-game condition~~
  + ~~Intercept ‘pause game’ condition~~
* ~~Compress the audio codec – Ogg vorbis maybe~~
* ~~Load options from a config file at startup (and save them)~~
* ~~Add nice-looking background images for parallax (random?)~~
* Get rid of that retarded invulnerability bug
* ~~Look at ways to increase performance~~
  + ~~Streamline gravity calculation (staggering?)~~
  + ~~Particle density option?~~
* ~~Scoring system~~
  + ~~Points for each type of enemy~~
  + Points subtracted for player deaths
  + ~~Scrap maybe worth nothing (Replace with crappy enemy?) or random loot~~
* ~~Big-ass messaging system for more important messages (scoring, loot)~~
  + ~~Layered scrolling, and implicit App2D functions~~
  + Possibly smaller messages for tiny point-gains
* ~~Aggro range on enemies~~
  + ~~Far, activate~~
  + ~~Near, retract~~
* Loot i.e
  + Lots of points
  + Double Ship (maybe)
  + ~~Shield [antigravity]~~
  + ~~Weapons i.e~~
    - ~~BackForth~~
    - ~~StripeLaser~~
    - ~~Spiralis~~
    - ~~Harbinger (Programatically similar to drones?)~~
    - ~~Getron~~
  + Right-click Teleport (short range direct)
  + Engine upgrade
  + ~~Antigravity~~
* On death = random loss of points + loot (only equipped?)
* Ship customisation screen
  + Involves either separating menu system into pause-time menu + main OR renew with a different keyboard shortcut.
  + Inventory structure
    - Categorized by loot type
  + Split player data into class
  + Add ‘tooltip’ help when selecting items in inventory
* ~~Game-saving system~~
  + ~~Each enemy destroyed is remembered~~
  + ~~Remember the player’s current directory~~
  + ~~Wipe old savefiles on new game~~
  + ~~Fix main menu with respect to the saving system~~
  + ~~Remember collected loot~~
  + ~~Currently equipped items~~
  + ~~Difficulty tier~~
  + ~~Cinematic level~~
* ‘The Universe Thus Far’ screen; a tree view of viewed folders + enemy/point stats associated with savefiles.
  + Probably build it into the savefile viewer
* ~~Folders with more in them are rendered as visibly larger wormholes~~
  + ~~Will have to implement a secondary traversal algorithm~~
* ~~Stick generic enemy properties in their own tag~~
* More enemies! E.g
  + ~~Grunt~~
  + ~~Ticker (random bullet within range, supercrap)~~
  + Popper (moves around, randomly emitting bunches of particles)
  + ~~LostData (Suicide ship, moves towards player and explodes)~~
  + Duo (Consists of a couple of files, spirals bullets outwards)
  + Quattro (similar to ^)
  + Impossibler (Frickin’ huge spiral, need to work on particle engine maybe for this)
  + ~~Bomber~~
* ~~Genericise the enemy classes~~
* ~~Write a story for a possible campaign mode~~
  + ~~Mine wasn’t too good, work on other stuff while Josh polishes it.~~
    - ~~Adapt Josh’s, with a little more humour~~
  + ~~Create artwork for~~
    - ~~Player’s ship before transformation~~
    - ~~Player’s ship after transformation~~
    - ~~The ‘warp device’~~
* Cinematics engine
  + ~~Support for dialogue and subtitles~~
    - ~~Automatic hiding/showing of subtitles correspond to dialogue length~~
  + ~~Genericise clearing/loading of levels; cinematics need a clean slate~~
  + ~~Create a new set of entity definitions with overridden control tags~~
  + ~~Create a module to read from ‘cinematic definition’ file:~~
    - ~~Timings~~
    - ~~Entity creation~~
    - ~~Position changes~~
    - ~~Sounds~~
    - ~~Dialogue~~
  + ~~And play these timings back in a ‘cinematic mode’ where no player interaction is allowed.~~
  + ~~‘Cinematic bars’~~
  + ~~Credits [at start]~~
  + ~~Faders~~
* Improve ease of modification
  + ~~Store dialogue subtitles in a file rather than in code~~
  + ~~Move enemy/size database into a file~~
* ~~Have a crack at using some OpenGL shaders~~
  + Distortion exploding effect
  + Instance particles?
  + ~~Warping of wormhole vicinity~~
* ~~Add ‘tiers’ of difficulty; escalating while game progresses~~
* ~~Move as much data as possible over to JSON~~
  + ~~Weapon definitions~~
  + ~~Savefiles~~