Programming meeting minutes - October 3, 1996

❖ October tasks ❖

We will meet every week on Monday @ 11:00 am in the war room.

General notes & issues:

- ALL tasks given for each programmer should add up to 1 month (~ 20 days) unless otherwise noted.

|| This indicates a task that is going on in parallel with another task (See Andy's section for an example.)

General issues that need to be addressed:

- 3dfx cards - J.G. believes that it would be helpful if we each had one for debugging reasons. * Tony, could you investigate?
- Pentium Pro 200 - J.G. needs some time on one for testing of David Wu's rasterizer. * Dave or Billy, help?
- David Wu's rasterizer - still has bugs, including a crasher - what is time estimate for final sign off of David's stuff - Jeff? (see J.G. issues for more info)

Pete:

task name: ship systems and Quick & Dirty (Q&D) hud displays for debugging.
time scheduled: 10 days
time spent: 5 days

technical description (what it is/means):

- ship systems: make repair system more robust. Rework gun and missile systems to implement design changes. Complete implementation of remaining systems.
- Q&D HUD: put temp code in cockpit file to display analog gauges of ship systems for testing

visual description (what we'll see and how we will test completion):

- ship systems: Powerplant distributes power to energy pools. Guns and missiles will fire/select. Systems charge up and convert powerplant energy to system energy. Engines effect dynamics. Power changes effect system efficiencies. Damage to systems decrease system efficiency. Repair system repairs according to preset priorities. I may have missed some things in the description...
- Q&D HUD: Should see analog gauges (variable length lines) representing all ships systems.

task name: rough collision system.
time scheduled: 10 days
time spent: 0 days

technical description (what it is/means):

This should be a very primitive collision detect algorithm. This task is currently up in the air while we finalize our constant model update paradigm.

visual description (what we'll see and how we will test completion):

Objects look like they collide with spheres.

Axel:

---

task name: VRAM memory manager.
time scheduled: 2 days.
time spent: 2 days.

technical description (what it is/means):

visual description (what we'll see and how we will test completion):

task name: Rough gameflow system.
time scheduled: 20 days.
time spent: 0 days.

technical description (what it is/means):

visual description (what we'll see and how we will test completion):

Jeff G:

---

issues:

- Sorting objects - How will we do this? BSP objects & depth sort them in world? Depth sort faces of object when bounding spheres intersect? Other options?
- Native support for 3d accelerator cards - which ones will we support natively? - GL being reworked to support 3DFX.
- Occlusion model - this needs to be discussed more and researched.
- Still doing tests with cap. ships from Dean and Chris D..
- David Wu rasterizer:
    - Crasher
    - Needs to be tested on Ppro.
    - Test transparent textures. This is done and works.
    - Need to make sure we have a final version "check list" to make sure hand off is complete.

task name: reworking engine to support accelerator card DLLs
time scheduled: 5 days.
time spent: 2 days.

technical description (what it is/means):

re-architect the win95 interface to the graphics engine to allow the graphics system to be loaded as a DLL, facilitating support for 3d accelerator graphics cards.

visual description (what we'll see and how we will test completion):

the game will be able to run and load graphics layers for David Wu's rasterizer, direct 3d, and the 3dfx, with more cards to be supported in the near future.

task name: File system API and wrapper class.
time scheduled: 5 days.
time spent: 0 days.

technical description (what it is/means):

design a file system interface that is efficient on all platforms, allowing seamless access to data files stored on a variety of storage mediums.

visual description (what we'll see and how we will test completion):

the game will be able to load data files that are stored in various formats on various source mediums, without the programmer needing to know where the data actually was.

Hugh:

---

file: hugh\sw\font\wc5.doc

wc5 work, hugh david

issues:

- Still waiting for info. from MS about TrueType fonts.
    - have name of engineer at microsoft who i will email when TrueType again becomes an issue...
- Going to 1 day PSX conference in San Mateo.
    - excellent conference, especially for GTE ... we have copies of documentation ...

task name: Font system
time scheduled: 15 days.
time spent: 10 days.

technical description (what it is/means):

NOTE: all this information will be in the wc5 database after i have tested the font code in the game

- uses dos-based tool to convert dpaint "brush" to proprietry data format
- support for multiple fonts
- characters can be any size (x,y)
- realtime functions allow:
    - PutChar ( ) and PutStr ( )
    - proportional spacing on/off
    - transparency on/off
    - absolute / relative cursor movement
    - left margin for indentation
    - setting of width of "space" character

- pc, stand-alone testbed program (not yet tested within wc5 databased version):
    - supports 1,2,4,8 bit data
    - supports "base colour" change
- psx:
    - supports 4,8,16 bit data

visual description (what we'll see and how we will test completion):

NOTES:
- flowchart underway
- all this information will be in the wc5 database after i have tested the font code in the game

- make:
    1. use, e.g.: 'lbm_rgb -l -p -q -8 testfont.bbm' to prepare "rgb" and "pal" files
- init:
    2. use 'Font F("rgbfile", "palfile", xcell, ycell);', where xcell & ycell refer to the spacing of the characters in the dpaint brush file
    3. use 'F.Load ( );' to get the font data into memory
    4. use 'F.PutStr ("hello world\n");' to see results on screen

task name: Cockpit systems.
time scheduled: 20 days.
time spent: 2 days.

technical description (what it is/means):

i have looked at the code already in existance, written by pete shelus & jeff grills.
as i see things so far, this is a simplified explanation of what currently happens

- cockpit render:
    - dustcamera render:
        - display starfield
    - gauges render:
        - power systems
            - display background / limiters
            - display engines' power
            - display guns' power
            - display shields' power
        - radar map
            - draw radar 'circle'
            - for all objects, if object not owner of radar, display as radar point
        - targetting
            - display 'circles' and 'crosshairs'
        - damage info.
            - display backgrounds / limiters
            - for all power systems, display system as normal / damaged / critical
        - guns
            - for all power systems, if gun, display

visual description (what we'll see and how we will test completion):

NOTE:

- flowchart under development, see technical description above

Jason H.

issues:

- Still designing out systems with Andy & Pete.

task name: Rough AI
time scheduled: 20 days.
time spent: 15.

technical description (what it is/means):

- Completed rough flowchart.
- wrote AI scheduler.
- Spec'ed out formation system.

visual description (what we'll see and how we will test completion):

- Non-visible system. Has been thoroughly tested with a test driver program through over 20 million data inserts with 10,000 objects on the scheduler at one time. Performance has been tweaked for maximum speed.

task name: Rough AI Design
time scheduled: 5 days
time spent: 5

technical description (what it is/means):

- basic component design for AI subsystems and data flow
- definition of tools required for the AI system to be flexible and dynamic

visual description (what we'll see and how we will test completion):

- AI subsystem flowchart

Andy

Issues:

- Still getting familiar with all code and systems (Wing 5).
- Still getting familiar with all code and systems (Mission Editor).
- Still getting familiar with all code and systems (Visual C++).

Task name: || Become familiar with code.
Time scheduled: 10 days.
Time spent: 5.

Technical description (what it is/means):

- Added several camera views.
- Added a mock cockpit gauge.
- See other tasks.

Task name: || Continued mission editor work.
Time scheduled: 15 days.
Time spent: 3 days.

Technical description (what it is/means):

- Added the ability to load in a MED created mission that placed ships [objects] in the world
- Added the ability to place ambient and directional lights in a MED created mission.

Visual description (what we'll see and how we will test completion):

- Create a mission with MED. Use the -m[filename] parameter in _debug.cfg to load in the mission. Enjoy the stunning objects and dazzling lights.

Task name: || Rough mission system.
Time scheduled: 15 days.
Time spent: 3 days.

Technical description (what it is/means):

- Discussed creation and implementation of mission system :
    - Rough MCP (Master Control Program) that would interpret the master series program and determine whether to run gameflow or spaceflight.
    - Would like to have by the end of the October spaceflight objects in the world running minimal programs [at least GOTO commands].

Visual description (what we'll see and how we will test completion):

- Flowcharts & design notes.
- At the end of the month, missions created in MED should be able to place objects and go to specific points in space.

ACE to-do list:

---

issues:

Tony writes, "These are what I consider to be my top three priorities right now. Let me know if you agree/disagree with any. I'll be collecting and sharing information before doing anything else."

- *Source Control Research*
    - Find info and opinions on MKS Source Integrity.
    - Find out if OSI owns a copy or buy one copy.
    - Install and test with Wing 5.

- *Object Editor Research*
    - Find info and specs on MultiGen.
    - Arrange MultiGen demo if possible.
    - Plan for ACE solution if necessary.

- *Language Freedom*
    - Standardize language freedom for all of Maverick.

- Document file formats and API.
- Research support of Asian languages.
- Plan for custom translator tool ??
- Write game module/library ??