**METANACT – Project Evaluation**

*Seb Holzapfel*

**Addressing the Solution’s Initial High-level Objectives**

* **Employ a sparse range of content to keep the player interested**

I think the game has quite a reasonable amount of content within; about 6 weapons, 4 enemies; a range of background images; different types of scoring and loot: as well as the game’s obvious facilities for using your own filesystem’s structure in building its own levels. Also, it’s easy to modify the game if you’re a developer or enthusiast and add your own content to it.

* **Provide intuitive game mechanics that can be easily familiarised with**

I used a pretty standard control scheme, with WASD as movement, and mouse as aiming. There is also an option in the game’s configuration to use movement that is relative to the player’s mouse position for players that are more used to this control style

* **Operate in a stable manner on multiple operating systems**

As far as stability goes, I think I’ve met this goal; through testing on ~6 different computer systems with a range of CPU, RAM and graphics adapters. I also did some performance optimization to the program running as quickly as possible on these systems.

* **Educate players somewhat as to the underlying structure of filesystem**

Fundementally, the goal of the game in fact IS to discover the file system’s structure, in order to increase score – So, I think this goal has been met. ALTHOUGH, I was definitely throwing around ideas involving the display of information text when you were situated in important system directories (Sys32, My Documents etc).

**Major Problems**

I encountered a number of problems while developing this application: Most are detailed in the project log-book, and there are some more specific examples & their resolutions in the Testing & Evaluation section. I’ll re-state a few here in short for the sake of clarity:

* **In-game camera was crashing game about 10 seconds after execution;** turned out to be a low-level memory allocation error.
* **Game crashed on computers with Intel graphics cards**; turned out to be a semantic inconsistency between ATI, NVidia and Intel’s shader language compilers within their drivers.
* **Game was running extremely slowly**; turned out to be for 2 reasons. SFML wasn’t doing it’s own sprite-culling; so I hand-wrote a culling routine – and Gravity calculations were bottlenecking the CPU, so I staggered the calculations so they weren’t occurring every frame.

**The Good, The Not So Good & The Future**

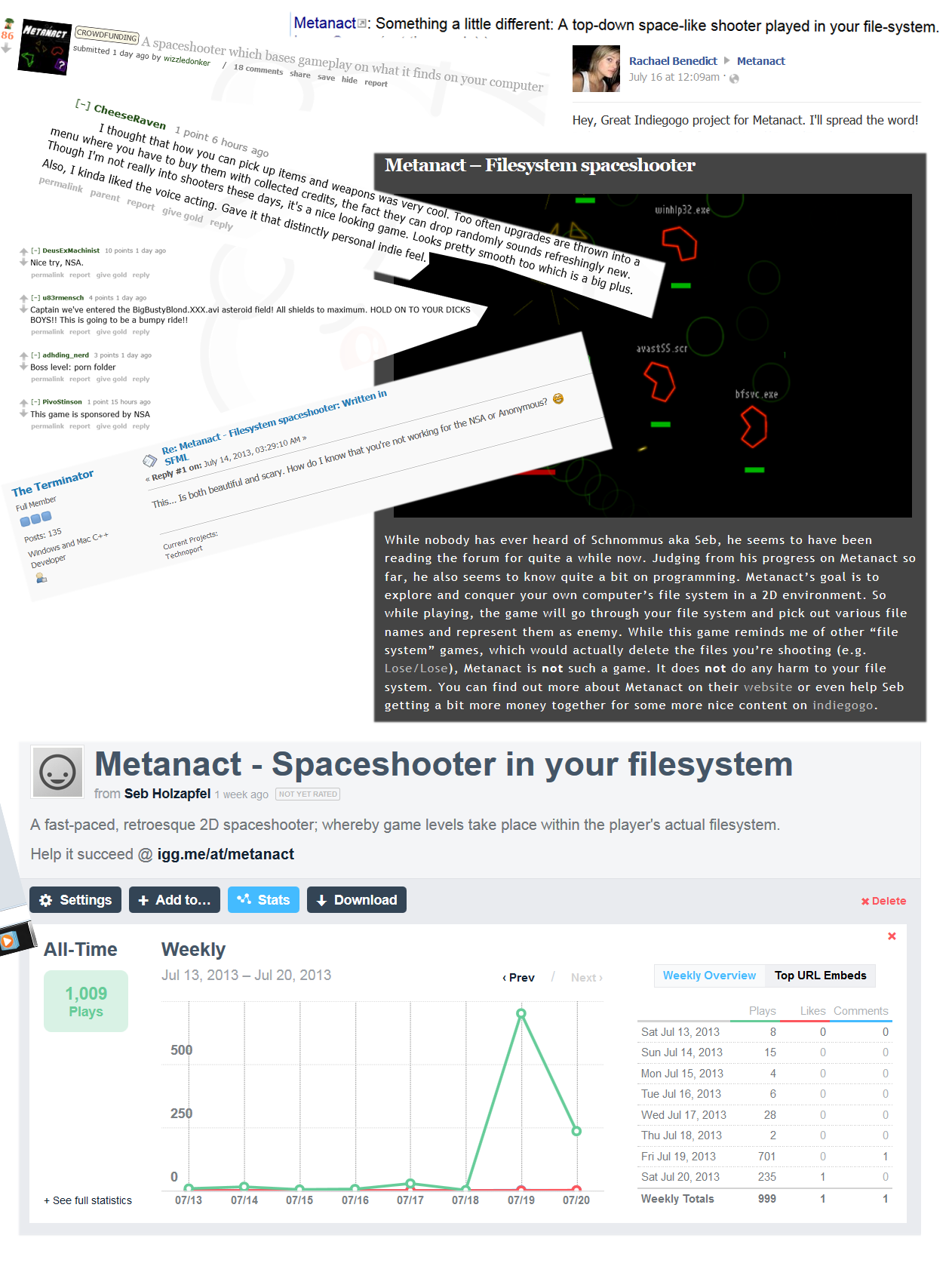
Although it’s my belief that the applications design objectives were met to a satisfactory extent; there still areas which could be improved – primarily in the areas of game content, cross-platform functionality, and assurance of data security. Metanact is currently Windows-only purely because of the difference in filesystem structures across operating systems (C++ & the libraries I used are cross-platform) – so, in the future it would not be too difficult to port Metanact to both Mac & Linux. In terms of content, Metanact is equipped; but unfinished – there is only a single beginning cinematic: the rest of the story is leftunexplained – So, in the future; more cinematics, enemies, weapons could be introduced. Finally, there’s the issue of clients perhaps believing that Metanact is spying on their data **–** an aspect of ethical concern. In the future, I could possibly obtain some third-party certification that guarantees to users that their data is not being spied upon.

**What I’ve Learned**

Speaking technically, I’ve learned some new programming libraries; more nuances of the C++ language, some more detailed insight into media packages such as Reaper and Photoshop, the OpenGL shader language, and how to structure a fast, realtime game engine. Perhaps more importantly though: this is the largest programming project I have ever undertaken [in a singlehanded sense; I’ve contributed to open-source stuff], let alone (basically) completed. I’ve learned how to structure working on projects of this magnitude through rigorous planning, version control and continuous testing. In the final stages of the project, I submitted it to a crowdfunding site and publicized it in a number of places – a learning procedure in itself!

Metanact, I believe; stretched the edges of my programming ability – forcing me to learn new techniques to solve problems as the application developed. It’s increased my ability to be organized & productive on long-term projects; and I’m just gonna say it: It’s.. made me a better human being. Well, perhaps at the expense of other areas ;)

**What some randoms from the Internet say about Metanact:**

****