Delta Force Post Mortem

Post Mortem: Delta Force - Andrew Giannopoulos

Reflection on dev process

I believe that the project was both underestimated and overestimated in terms of what could be accomplished in the time that was provided. The original goals that we set out to accomplish seemed reasonable and realistic, but by the end of it only the most basic elements actually ended up making it into the game itself. While the game was originally going to feature walking monsters, death platforms, and more story-driven elements, many features were cut in order to allow for a finished product to be released on time. I'm sure every game developer in history has had this thought at some point, but had we better utilized our time management and had more of a laser focus on what NEEDED to be done at the time, much more could have been accomplished. The original vision wasn't that far out there and was mostly realistic, had there been a stricter focus on project management techniques and keeping on track.

Issues Faced

Regarding cross-platform development, a multitude of issues were encountered. Early in development it was a struggle to simply set up the PSM environment, causing us to overlook some aspects of cross-platform development until late in the development process. While things like integration of the control scheme and PSM shader technology were discussed throughout development, nothing was actually tested or implemented until much later on by which time it was really too late to do proper playtesting (in the case of controls) or just general troubleshooting in the case of shaders.

Additionally the hardware limitations caused massive frame rate drops in the game. While these were found to be in part caused by erroneous update calculations that had been introduced into some Draw code, the number of platforms being drawn in the levels still caused issues. Whether this was because the platforms were still being processed despite being offscreen or some other issue, the discovery should have come sooner so that a resolution could be found earlier and easier.

Awareness of cross-platform dev

As we discussed what to include in our "Options" menu we raised options like volume, brightness, resolution, and fullscreen. We then realized that since the PS vita deals itself with things like brightness and volume internally, and doesn't need to worry about differences in resolution or being fullscreen, we could essentially scrap the options menu that we had planned for the PS Vita version of the game (which, in a move that would make TotalBiscuit roll in his grave, led to it being scrapped altogether).

And so because we decided to omit an options menu, the PC version of Gravitas is locked to the same resolution as the PSVita version, and does not (at time of writing) allow for fullscreen. This creates an issue due to the control scheme on PC utilising mouse-click and drags as a method for gravity control. If the player were to accidentally click outside the game area this interrupts gameplay and causes the draw

function to stop, while the update function continues. This became an issue in playtesting and has since not yet been resolved.
Since the various control schemes for PS Vita and PC were discussed at length during the inception and brainstorming phases of development, these turned out to be relatively simply and quickly addressed, with the only real issues being balancing of gravity strength and player movement speed, and making sure that the player can only really move perpendicular to the direction of gravity (with of course the exception of jumping).

Suggestions for future improvements

In regards to the development process, less time spent prototyping could have helped. Initially we spent a while familiarizing ourselves with the Farseer Physics Engine. Had this time been better spent we might have been able to accomplish more. That having been said, the prototypes that we did produce were extremely simplistic due to our knowledge of the physics engine and how to create objects in it, and had we instead sought to separate the concepts of creating a prototype and messing around with the engine, more could have been done on both fronts. As it was, our prototype ended up being both an extremely limited gameplay test (that was also very broken and not at all representative of the final product) and a demo and test of the Farseer Physics Engine. Even just doing a prototype level on paper to establish core gameplay mechanics could have been more useful than what we had passing as a prototype. I would agree with the Hearthstone Dev's GDC talk regarding prototypes, and how useful it is to code the prototype in a different language so that you're not tempted into code reuse. What we saw at points was prototype code that was unsuited for use in the full game being included anyway.
This separation could have also allowed us to earlier focus on some issues that were hiding within the engine, such as bugs in collision detection causing issues when the player tried to jump (which came out as quite a large bug in the full game which, as of now, still hasn't been resolved).

Final Thoughts

Ultimately I am very proud of what we managed to accomplish together. I'd like to thank Jackson Luff and Iain Dowling for being such good sports, and humoring me through what was truly a rollercoaster ride of emotion. Feedback has been overall positive, and what initially set out to be a puzzle-platformer looks now more like a game akin to Super Meat Boy or I Wanna Be The Guy in terms of gameplay pace and intensity. Good job guys.

I'd like to finish with the Retrospective Prime Directive:
"Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand"