

Delta Force Post Mortem

Post Mortem: Delta Force - Iain Dowling

Issues Faced:

We faced multiple issues in trying to develop a game for multiple platforms. Each member of our group was adequately experienced at developing decent games for the windows pc, however none of us had any experience at developing games for the PS Vita.

The first issue was attempting to learn the new IDE that came with the Vita, PSM studio. This did not take long, as PSM Studio was not too dissimilar from Visual Studio. However each team member had to feel as though they were comfortable working in a program other than Visual Studio, so we spent as much time familiarising ourselves as necessary.

Another issue we faced in cross-platform development was adapting controls to work on both platforms. All of our main development on the project was done in Visual studio, and so controls were adapted to work for the keyboard and mouse. However as part of the project we had to implement methods to determine whether the system the game was running on was Windows or the PS Vita and then alter the controls accordingly.

The last main issue we faced was making a game that worked with the restricted hardware of the PS Vita. The game we created worked smoothly at 60 fps on windows PC, however when we ported it onto the PS Vita we found that game was very laggy and slow. To solve this we had to cut back on texture and level sizes, as well as refactor code to create more efficient functions that would run much more smoothly on the PS Vita.

Awareness of cross-platform dev:

When working on this project, we had to take into consideration the hardware configurations for the two different platforms, windows PC and PS Vita. The Windows PC is a much more powerful device that can handle large texture and many objects at one time, however the PS Vita has much less memory and so can not handle as much.

The differences in hardware meant that we had to cut several features that are available on our PC version from the PS Vita version. Shaders and Particles were dropped because they were too memory intensive for the Vita. Also, when porting our game to the Vita, we lowered the texture resolutions for many basic shapes such as the rectangles and circles so that they would take up less space in memory.

As a team we were aware of the different controls schemes for the windows PC and the PS Vita, and so during the design phase we developed separate controls schemes for each that we thought would be intuitive and easy to use. In this way, we wouldn't find ourselves trying to adapt a computers control scheme to a PS Vita and finding the end result broken and difficult to use.

We also needed to take the screen Resolution into consideration. A computer's screen resolution can range from 800x600 to 1920x1080 and higher still. However a PS Vita's screen resolution is capped at 960x544. To account for this, we designed the game to run a 960x544 on both of the target architectures.

Reflection on dev process:

The first idea of our team when developing Gravitas was to create a game that would be fun and rewarding to play. Based off that original idea we created the original design for Gravitas, a 2D platforming game that required gravity manipulation to pass through each level. I feel as though my team and I spent an appropriate amount of time at this design stage, and we were quite productive while working on the design documents and technical documents.

Within the design stage, Delta team also spent time developing an art style to use in the game. The final design would use neon colours to create a bright and attractive game. I feel as though the time spent on this was wasted however, as ideas for art for some objects, such as crates and spikes, were never implemented.

Following the design process we whipped up a quick prototype to test out the gameplay mechanics. After playing with the prototype we felt the design of the game was solid enough to continue working on. This stage of development I felt lasted for too long. After we realised that the prototype was fun and would be enjoyable to create a full game based on it's idea, we continued to play with it. We spent too much time trying to implement more features into the prototype, such as animated player movement and particles. Next time I work in a team development project, I will definitely limit the time I work on the prototype to allow more time to spend on the real project.

After finally ditching the prototype, we started developing our main game. Regrettably our team was reluctant to let go of the prototype, and so some features from the prototype were cut and reused in the main project. Although this saved time early on, it was later realised the code was highly unoptimised and inefficient and so cost us later on. We ended up cutting all original prototype code from the final product. In future projects I will make sure to completely start again from scratch and to not reuse prototype code, because it ends up being more trouble than its worth.

When working on the game, we tried to follow milestones we had set up earlier during the design stage. Following these milestones really helped our team stay on track and it made sure we always had something to work on. I really like the idea of milestones, and after seeing how effective they were for this project, I would consider using them again for any future projects.

Towards the end of our development time period, we started to realise we had bitten off more than we could chew, and so started cutting back on features to make sure that the project would be finished in time and that it would fill all the criteria for the assessment. Among the features that we lost were; multiple enemy types, spikes, extra levels and other puzzle/physics objects such as crates. Although these features were cut, play testers unanimously agree that the game is fun and worth playing.