Cain, Billy

From: McCurdy, Keith
Sent: Tuesday, February 18, 1997 12:15 PM
To: Mattrick, Don; Nakamoto, Rod; Downing, Dave; Roan, Frank; Shelus, Peter; Cain, Billy; Gordon, Bing
Cc: Wilson, Tim; Cronce, Scott; Hollis, Andy; Garriott, Richard; Grajeda, Mike; Sullivan, David; Everett, Jeff; Carloss, Alex; Hilleman, Richard
Subject: Wing:Prophecy TDR report (short version)

Wing:Prophecy TDR was held on monday Feb 17. In attendance in addition to the team was Keith McCurdy, David Sullivan (lead on Soviet Strike) and Jeff Everett (lead on Crusader). Tim Wilson cam down with the flu and could not make it. Still hoping on written feedback from Tim Wilson and Scott Cronce on the TDD.

General Impression

This team has already made considerable progress and has some key technologies fully under control and virtually finished. The team is staffed with quality engineers and team members from previous WC products that understand the game they are trying to make and the issues that have been learned in previous product cycles. Many of the production processes are in place (mission editor, renderer, rasterizer, mission system, game flow control) and do not need to significantly change to meet the product definition. The team contends that they have consistently meet their schedules (defined as monthly goals) for the last several months and are confidant that they schedules in the TDD will be accomplished. On the negative side the team is highly dependent on Jeff Grills to implement and maintain the key core technologies of the renderer, rasterizer, 3D art process, network play, client-server model, and optimizations. Also the AI, a key component, is not design on paper, and thus the schedule is fuzzy. The team *seems* to have a good understanding of what they need to accomplish when discussed verbally.

The schedule is ambitious. The good side is that there are 7 weeks from alpha to beta and 8 weeks from beta to final. The bad side is that there are still features being implemented during alpha (user controlled cameras, take offs, landings, install, effects, and many "final passes" on features). Its difficult to know how unfinal these features will be when the team wants to declare alpha or how many bugs the team will have to deal with when they are still implementing features during the alpha period (which would just delay beta). Also there are still art assets (all (?) the movies) being integrated after the beta date. I do not know how painful or difficult this has been in previous WC but the team seemed a little un easy about it. These are not killer issues and the fact that there are 15 weeks from alpha to final is very good.

Things Missing from the TDD (team did a great job getting together what they had in the time given)

- We just received at the start of the TDR the full list of detail descriptions of each engineering task on the schedule. I still need to review and give feedback to Pete.
- We also received during the TDR a schedule for the mission creation team from Billy Cain. I still need to compare to the technology schedule and give feedback.
- There are no traditional team milestones in the schedule. The team integrates on an almost daily basis and things come on line linearly rather that at milestones.
- The AI schedule and description of functionality is very light and does not communicate the amount of work to be done.
- There is no decision yet on the network communications model or the client server architecture. It discusses this in the TDD and outlines the options.

Major areas in good condition or accomplished

- The renderer and rasterizer are 80-90 complete and operational in the game. There are some potential optimizations to be done. They are fairly well understood and not deal breakers if they don't work out. It needs to be converted to support the new DPrimitve in D3D. Not a problem.
- The mission editor if functional and being used by the designers to create missions now. There is continued maintenance on the tool but no large features to be done.

- The run time mission system and game flow is working and running in the game.
- The ability to create "rooms" and branch through the game and missions is complete.

Major areas to be done

There are ~14 weeks to alpha.

- AI. Still the majority of work to be done here. 62 days scheduled between March 1 and June 5 (alpha) to one programmer. That's 62 days out of 72 work days available.
- Network communications. There is some basic ability to pass packets between game clients. There still needs to be a decision made whether to go synchronous or asynchronous (to be made in the next 2 weeks) in the overall architecture for the way the game handles network communications. This task is assigned to Jeff Grills. See risk section below.
- Client server. If we decide on a asynchronous communication model (most likely) then this will need to be implemented also.
- Audio and interactive music. Not started. See risks.
- Warfare management (guns, missiles, explosions, damage).
- Briefing and Debriefing system.
- Communications (in game messaging) and formation.

Major risks areas

- AI not being fully designed on paper and scheduled. I do not have good data on what the tasks and requirements are as most everything is in the programmers head. Luckily Frank Roan implemented the AI for WC IV and is working with the engineer on the AI. Also, AI is not finished (fist pass) until June 1 and June 5 is alpha. Mission will not be very will tuned with the AI at alpha.
- Jeff Grills being over scheduled and crucial to key technologies. In March (22 days), April (28 days), May (24 days) Jeff has more days assigned to him than are work days (mon-fri). This means nothing can take longer than he plans and he must work weekends or everything takes less time than he plans. This is not realistic scheduling no matter how good Jeff is. I would like to see the schedule written with realistic commitments from the programmers.
- Some of the game programmers are scheduled more than 20-22 days of programming tasks in a month. Again like the item above this means that they have to work weekends, can't get sick, nothing can take more time than the estimate, or they are all over estimating. There is no padding added to the schedule over all.
- Network architecture and client server model. No one on the team has done an asynchronous communication implementation. The team is concerned about the threading and Win95 and reliance of a server based model to resolve issues that are usually handled by the clients be synchronized (collisions, decisions, etc). I believe this has been solved by other teams at EA but it is a risk since the schedule is small for it and the team doesn't have direct experience with it. Potential help issue from outside the team.
- There are programming tasks on the schedule starting March 1st for hire that is not on board at this time. I believe Rod has him identified but not in house yet. Again he is booked for 21 days of programming in his first month.

Other areas of serious concern

- Mission creation. I still do not have a good handle on the schedule or risks associated with mission creation at this time. This is a TBD for me.
- Multi player capabilities will not come online until May 1 and June 5 is alpha.

Schedule

alpha - June 5
beta - August 28
final - October 23

Conclusion

A basic assumption of the team is that they have so little time that they must schedule themselves for 20-28 days of work a month. This is not how scheduling is supposed to be done. Its high risk that they are starting out here. There is no way I can defend this practice. The team's response is that that is the performance that they have been tracking to. This tells me either 1) They have been working 20-28 days a month for the last couple months and nothing has gone terribly wrong or 2) They accurately pad their schedule's and get things done faster than they plan. I hope it is #2. There is no question that they team is staffed with talented smart programmers. They have experience with the product line in the past and a good working relationship. The networking schedule currently assumes going with an asynchronous client server implementation, which is best for low latency (interent) and for highest frame rate possible on each client. Although this is the best decision for the product it is the highest risk and the most work. The other decision (synchronous) would mean that all machines would be limited to the slowest machine frame rate. This is hard pill to swallow on a game that we are going to sell as great 3D and multi-player.

Recommendations is a separate email.

/keith