The Ninth Annual Game Design Think Tank
Project Horseshoe 2014

Group Report: Rapid Prototyping in the Now

by Thom Robertson of Incandescent Workshop LLC
with help from everyone at Project Horseshoe 2014

Problem Statement

Not enough has changed in the videogame industry. Giant teams with high turnover burn through millions creating safe projects. While they connect with and hire from the major game dev schools, industry players otherwise shoulder no responsibility for training new employees.

At the same time, the proliferation of cheap, powerful tools and the new, young, "indie" game devs allow many more small, bold games to be generated quickly. Since any young, unemployed game developer can call herself “indie”, the indie scene takes de facto charge of training new young developers.

As churn continues in the industry, these two worlds will come together, and the industry will not only not make good use of the skillz indie developers have cultivated. Big industry companies will fundamentally misunderstand and dismiss skillz of young developers, and force them to change.

Solution

This workgroup suggests that it is the game industry that must change. Big companies must stop relying on the “gut instinct” of their owners, and also refuse to be guided by their marketing team.

Instead, they should embrace a prototype-heavy development structure, and embrace the rapid-prototyping skills of the new generation of young devs.

A prototype heavy development structure is not new, and is fairly well understood. Many smaller development companies (especially those who subsist on business from larger companies) learn the value of rapid prototyping, and build small, ad-hoc teams and projects, tasked with quickly building prototypes that will land contracts.

But when the contract is finalized, that same company will get down to the multi-year slog of building a “real” game, leaving behind the specific skills and values that benefit rapid-prototyping. And the young developers will have to adapt or leave.

Other companies that make “free-to-play” games also embrace rapid prototyping, but their path is also alien to young indie developers. Instead they iterate on A-B tests and other detailed metrics, letting math, ROI, and player churn determine what they create and sell.

Game companies should hire young indies, and USE their rapid prototyping skillz. They should allow these new hires to do what they do best; wear multiple hats and quickly develop testable game prototypes in small teams.

Rapid prototypes are valuable in three major ways. First, rapid prototypes are the best way to separate good ideas from bad ones. Like the number of teeth in a horse's mouth, actually playing a game gives everyone the best understanding of whether that game is good and fun. And for game developers, the question is rarely "is it fun?". The most common question to answer is "How can we change it to achieve maximum fun?" Which leads to the second aspect of value.

Rapid prototypes serve as THE most powerful descriptor/example of a game idea, serving the whole team as a solid reference point. You can say "I have a great idea." You can WRITE "I have a great idea." You can provide diagrams and other content. But nothing beats actually having a functional game in front of you.

With a prototype, it's easy for every team member to understand you when you say "Make it twice as long" or "Make it not so frantic." A prototype beats all other forms of documentation. Its descriptive power, and its ability to anchor the imaginations of the entire team are unmatched.

Finally, rapid prototypes can grow into finished games, without a complete rewrite. Many disagree with this, and feel it's self-evident that prototype code and data must ultimately be thrown away, and replaced by the "real" game.

I don't agree. In my experience, it's all a matter of choosing the right tools and development processes. It's quite true that rapid prototyping often results in clumsy or poorly designed code. But code is always changing, especially in a larger team. Bad code has a tendency to stick around and accumulate in dark corners of the codebase. But at the same time, teams of coders have a tendency to kick over the rocks and attack bad code when they find it. This is a predictable behavior of certain types of coders, so team dynamics (and a balance of team personalities) is important and worth managing.

It's easy to find examples of early (prototyping) content making it into finished products, even in big games with no actual prototypes. And if that content is proprietary or requires licenses, it can be a disaster. But don't use this to tar prototypes. It's not hard to be careful with licensed content; Even prototyping developers can learn to steer away from improper content.

And one of the many strengths of a skilled rapid prototype-er is the ability to find and integrate valid game content quickly. For big game companies that have their own sound teams, this isn't so important. But for a mid-sized or small project, buying a music loop for $20 is perfectly normal, and (used appropriately, with proper attention to license and attribution) is just as likely to be used in the final product.

Example

This game (Danc's Endless Faller) was built during Horseshoe 2014, and serves as an example of how rapid-prototyping can help communicate game ideas in a short time frame.

download the ZIP file containing this Windows game