

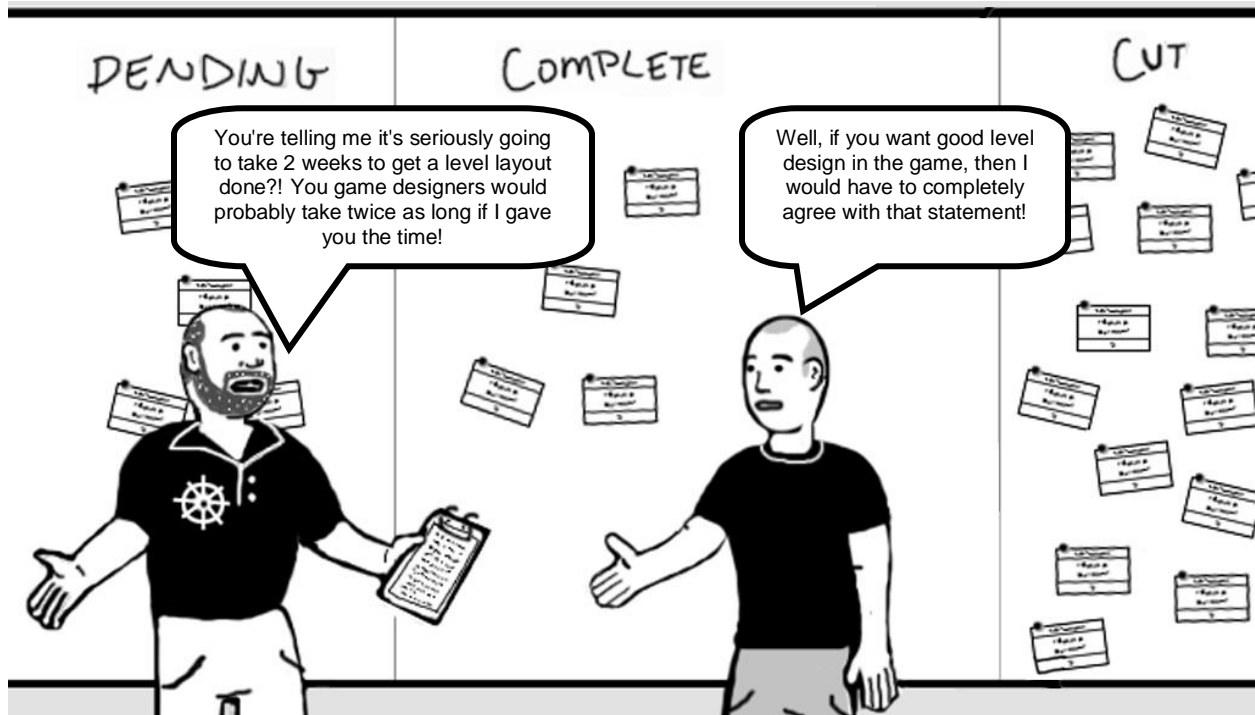
CHAPTER SIX: SCOPE

"Now the crowd breaks and a young boy appear; looks the old man in the eye as he spreads his wings and shouts at the crowd 'In the name of God my father I'll fly!'" - Bruce Dickinson & Iron Maiden "Flight of Icarus".

"The Knack of Flying is learning how to throw yourself at the ground and miss." - Douglas Adams.

One of the most heartbreaking things that can happen is project failure. There's the obvious investment of time and effort, the inevitable dissolution of a team, and the brunt of personal failure, but also the unrealized potential of an idea that never got to see the light of day. I've been on two big projects that both failed for the same reason: the name of the project. Well, ok, that's not really it, but it is generally a bad idea to call a game "Cancelled" or "Failure" or "Downfall" or "Damnation". You don't want the title of a game to be some weird harbinger of events to come! So, the real outstanding reason for failure is: **a lack of realistic scope**. In the last chapter we talked about brainstorming. Scope is how you start figuring out the viability of those brainstormed ideas in relation to actual game development! Ready?

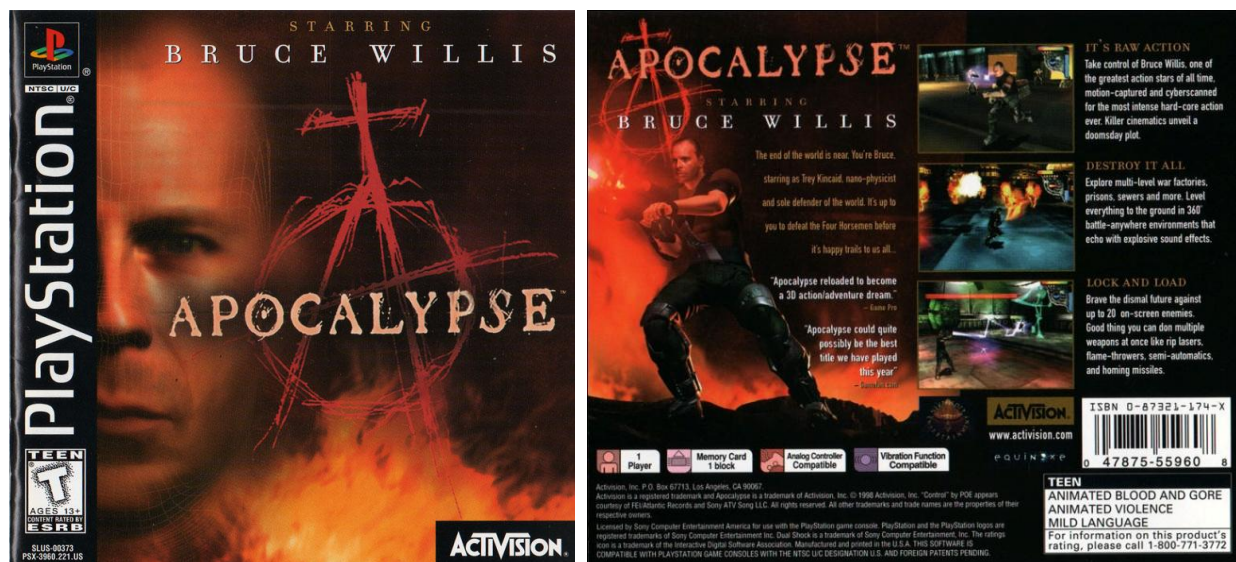
The simple definition of scope is "a measure of the breadth and width of a game in terms of assets, resources, time, and ambition." Assets are the individual pieces that are needed to make a game: models, textures, scripts, code libraries, sound files, etc. Resources are the people, software / tools, money, facilities, and equipment needed to produce the assets. Time is an estimation of the number of hours, days, weeks, months, and years it will take to create the assets with the given resources. Ambition is simply a measure of how big the game idea is!



What Happened to Apocalypse?

Alright, how about a good example of scope problems leading to a project's cancellation?

In 1997, I was hired by Activision to work on an in-house project called Apocalypse. This was my first next generation title (PlayStation and Saturn at the time), my first credited work as an actual game designer, my first exposure to 3D game design¹, and the first project I'd worked on that had a schedule longer than six or seven months! The Apocalypse design / art pit was a mish mash of homemade cube walls, couches, desks, and some bunk beds set in a premium corner office on the tenth floor of the Wells Fargo Center in Santa Monica, California. From our office windows, we had a beautiful view of everything east of highway 405 and north up into the Sepulveda Pass where the new Getty Center² museum was being built. I was one of four designers on the project and I had sole ownership of a factory-based level with massive machines and molten metal! To say I was excited would be a huge understatement!



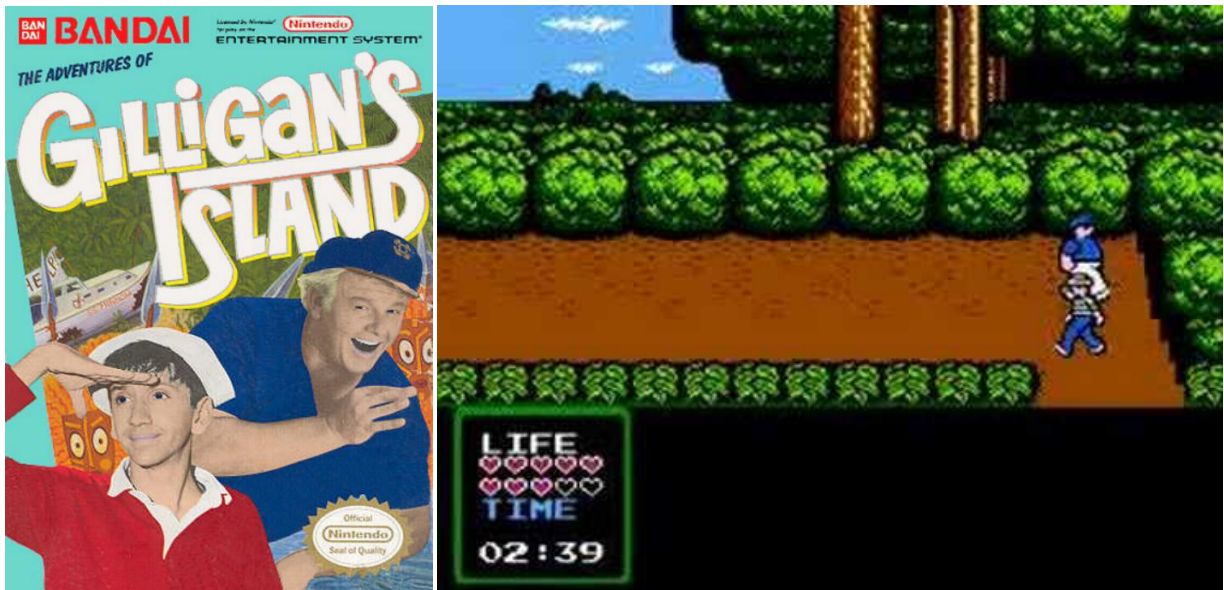
"The Apocalypse world is not a friendly one. You, as Trey Kincaid, must fight your way through sewers, factories, and rooftops (to name a few) in your journey to defeat the Reverend. Of course, to defeat this barbarian you must defeat his Four Horsemen: Death, Plague, War, and the Beast. The key to successfully completing each level is simple - kill everyone and everything that's out to get you." - Story synopsis from the Apocalypse game manual.

The original version of Apocalypse (not the one released) was a single player shooter with Robotron-style shooting mechanics (left stick moves, right stick shoots), sprawling apocalyptic levels, giant bosses (the Four Horsemen of the Apocalypse), an epic storyline (a battle between religion and technology brings on the end of the world!), and Bruce Willis starring as your AI companion! Bruce Willis? What?! One of the big selling points of the game was Hollywood action hero Bruce Willis. At the time, he was the biggest action star in the world with three successful Die Hard movies under his belt and this was going to be the first next generation title starring an actual major league actor! It was an ambitious and expensive selling point for the game which cost Activision more than \$1.2 million for the rights to use Bruce's voice and likeness!³

¹ I actually fibbed during my interview and said that I knew 3D Studio Max! Before I started the job, I managed to find a copy of the manual and read it cover to cover so that I at least knew my way around the software.

² The Getty Center is a beautiful art museum founded by oilman J. Paul Getty. The museum cost \$1.3 billion dollars to build!

³ "Playstation" by Geoff Keighley. Entertainment Weekly - March 5, 2004.



Do you remember a NES game called the Adventures of Gilligan's Island? Probably not. Bandai released the game in 1990; late in the life cycle of the NES and some 20 years after the cancellation of the television show the game was based on. One of the game's biggest features was an AI version of Gilligan who the player had to keep out of harm's way during each of the game's episodes. Unfortunately, this pixel-y little Bob Denver sprite had a bad habit of exhibiting unpredictable and very gameplay-unfriendly behavior like pathing into holes and getting stuck on collision!

Over the course of Apocalypse's development, leading up to its eventual cancellation, the team ran into three core problems related to scope: design, core technology, and artificial intelligence. Maybe if we'd taken a better look at games like Gilligan's Island we could have avoided some of these problems?

- **Design:** most of the preproduction cycle was spent creating a very rough vertical slice of gameplay as a proof of concept and writing the equivalent of a screenplay for the story instead of planning out the entire design of the game. This wasn't some heinous misstep on the part of the preproduction team⁴, but a necessity in getting the game green lit and funded for full development. Unfortunately, this left the team stranded with a bare bones prototype of the gameplay, but no structure for the rest of the game when actual production started. Instead of having a finite set of tasks needed to take the project to completion, the design department ended up pursuing a constantly moving target which was dictated from the top of the company down instead of from the development team up. Additionally, core gameplay mechanics were not prototyped and production started with key features like the A.I. either incomplete or missing altogether.
- **Core Technology and tools:** the original PlayStation was able to push polygons in the 10's of thousands and maybe a hundred thousand or so depending on the power of the game engine and which PlayStation resolution mode⁵ was being used. In the case of Apocalypse, the core engine technology (specifically the graphics rendering) wasn't powerful enough to push the polygons necessary to create realistic characters and large, vibrant environments. Keep in mind that games that involve multiple players usually require at least 50% more level layout space than strictly single player games. This is due to the fact that multiple

⁴ Preproduction teams are usually small groups consisting of the leads and production staff.

⁵ The original PlayStation had multiple interlaced and non-interlaced resolution modes that ranged from 256 X 240 pixels up to 640 X 480.

characters need more room to move around each other. Other limitations to gameplay were imposed by severely limited collision handling (only about 10 flat planes of collision were available at a time), a low cap on enemy spawns, and a lack of tools for quickly prototyping and iterating on any of the gameplay mechanics. These days, many developers opt to use third party solutions for a lot of these issues (Havok, NavPower, etc.) or go with complete third party engines like Unreal or Unity. At the time, Activision had a lot of experience with PC-based technology (used to create titles like Mech Warrior and Interstate 76), but newer consoles had to have largely unproven technology built from the ground up.

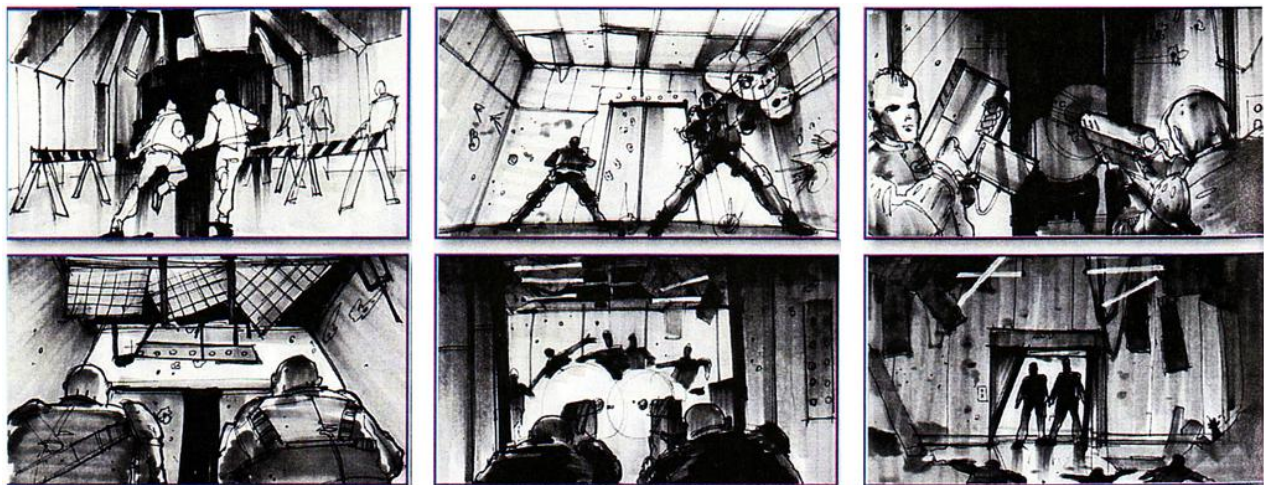


The original version with the player + Bruce.



The commercial release with just Bruce.

- Artificial Intelligence:** Trey Kincaid (Bruce Willis' character) and our entire rogues' gallery of enemies never progressed past simple point to point navigation and a handful of scripted behaviors (like pulling on levers to open doors) which more often than not resulted in lots of running infinitely into walls and spinning in place while yelling for cover. Enemies had very simple run and gun⁶ behaviors, but design was severely limited by only being able to have 3 or 4 on screen at a time. Workable solutions to these problems were never found despite many admirable attempts. Later on in the book when we cover AI and combat, you'll quickly see that it takes a considerable amount of work to make decent AI.



⁶ "Run and gun" describes games where players and enemies are constantly in motion with an emphasis on shooting.

As you can see, our scope problems were based on the fact that we didn't have a solid design or the technology to create the game no matter which direction it went! At the 1997 E3 in Atlanta, Georgia, we were in for an unforeseen surprise which marked the start of the end of this version of Apocalypse. In addition to the Activision booth, our game was featured as a third party title in the Sony PlayStation booth. However, next to our game was a strikingly similar game called One. ASCs⁷ One was basically the game we were trying to make, but with a lot more polish and no AI partner.



Apocalypse is on the left and One is on the right. At the 1997 E3, the Sony PlayStation booth had both games demoing right next to each other. Both demos took place on dense, urban rooftops with nearly identical gameplay. When Apocalypse was eventually released, the final version was nearly identical to One!

Apocalypse is a picture perfect example of scope problems eventually killing a project. Based on everything that went wrong with the project, you can see that scope is equal parts planning, execution, and anticipating problems that will require some adjustment of plans. When one or more of these things don't happen, scope can get wildly out of hand and result in too many features and too little time or not enough features and too little time!

In most cases, there's very little that a single team member can do about this situation unless you help effect change from within. The first thing to do is go around and talk to other people to help form a consensus on whether there actually is a problem or not. As a somewhat junior designer at the time, this was difficult for me to do since I'd never been in a situation like this one before, so I made sure I'd thought of some solutions to take around with me. I talked to my lead, my producer, and my creative director, but they were under a great deal of pressure from executives above them and most likely feared the repercussions of any large changes in plans (whether good or bad). I finally went all the way to the top with a plan for fixing the game....

⁷ ASC (American Softworks Corporation) was the publisher of the first Grand Theft Auto game in 1997!



The vice president of console development was a nice guy and I certainly appreciated him taking the time to listen to me and talk to me honestly about the project, but he also had a job to do whether I ultimately liked his answer or not. In retrospect, unless the team had been given another year to work on the game (which would not have made financial sense), there was no way my proposed changes would have come to fruition.

After the original internal team at Activision failed to deliver the game that was promised, the project was handed over to an external developer, Neversoft, who finished the game with Bruce as the only playable character and then built the successful Tony Hawk franchise on the same technology! According to developer Jamie Frstrom (in regards to the Dreamcast port of Tony Hawk), the *"code base was a hybrid of C and C++ code, originally written for Apocalypse, with a lot of Assembly for the renderer and some Assembly for the physics. Apocalypse was a Playstation game featuring Bruce Willis, which, we learned, is why in Tony Hawk the code for the classes of skaters is called "CBruce."*⁸

Formulation

Thinking about the scope problems with Apocalypse still causes me to lose sleep. I have recurring nightmares where I'm talking to the vice president and I'm only wearing a yellow thong or another one where I'm running in place shouting "ALRIGHT, LET'S GO!" while endless waves of enemies steadily drive the frame rate down to one frame per second!

Formulation is the process of transitioning an idea or group of ideas into more formal relationships and structures. It's one of the last, very important steps of the preproduction process before prototyping and development begin! In life, we formulate ideas every day. Where are we going for lunch? Which party are we going to this weekend? Where are we going on vacation next? Formulation is generally based on experience with a previous game's development or experiences with similar ideas from other games. There are basically three approaches to formulating an idea:

⁸ "Postmortem: Tony Hawk's Pro Skater (Dreamcast)" by Jamie Frstrom. Gamasutra - June 28, 2000.

- Creating a brand new idea - *"Hey, I have a great idea for a new player weapon that allows you to pick up large objects and throw them!"* OK, so Half Life 2 and Dead Space already did this, but perhaps it's a new idea within the context of your game idea? This is perhaps going to be a key feature of the gameplay, so it needs to be a primary scope concern especially if there are a lot of brand new ideas!
- Modifying an existing idea - *"That's a great idea! How about if we add the ability to charge those objects so that they explode when thrown?"* Again, not a new idea, but the scope is a little more reasonable because you're most likely taking an existing idea within the game and modifying it. In this case, exploding objects are probably going to be a class of object that will cover a variety of gameplay mechanics: exploding barrels, grenades, self-destructing robots, buildings blowing up, etc.
- Extending an idea as a creative solution - *"There's a puzzle that involves a switch on the far side of a pit that the player can't get to. We can use the weapon to launch an item at the switch."* An easy win when it comes to scope is leveraging one idea to work within a variety of different gameplay contexts. This cuts down on special cases which are gameplay mechanics that take a lot of time and resources to develop, but only appear once in a game.



In Tomb Raider Legend, a fully functional forklift ended up in King Arthur's Tomb and was definitely a special case gameplay mechanic. In effect, the forklift was basically just a different way to move crates around and open doors which could have been done much more cheaply with existing mechanics or less complicated puzzles. The forklift met with an unceremonious end when it fell into an abyss during a poorly done quick time event.

Scope Exercise

Let's look at a simple exercise in scoping and formulation. Here's how it works. Go through each category and pick the answer that most suits your idea. Each answer is assigned a numerical value. When you've finished answering all of the questions, add up all your numbers. If you don't necessarily understand some of the answers, come back to this exercise later once you've read through the rest of the book and it will totally make sense. Ready?

Originality: original ideas can be great because they are new and potentially innovative! Original ideas can also be incredibly difficult because there is no precedent set and they can be difficult to communicate and execute!

- + You discover that the idea actually has been done before, but poorly. -1
- + True inspiration strikes! You discover that the idea has never been done before. 0
- + You find that a similar idea has been done many times before and quite well. +1

Communication: do people understand your idea? If nobody understands the idea, then it needs refinement and you should go back to the formulation or even brainstorming stages. If a good group of people understand it, then the scope is probably reasonable!

- + You explain your idea, but nobody understands it. -1
- + Most everyone shakes their head in confusion, but one person seems to get your idea! 0
- + Your presentation goes fantastic and everybody understands your idea! +1

Time Estimate: for the sake of argument (and this exercise), let's say you ask various people on your team to give you a time estimate and they come back to you with a week to prototype your idea. It's a tight schedule, so time is a critical factor!

- + At the end of the week, you discover you're going to need more time. -1
- + At the end of the week, you have a working prototype to move forward with. 0
- + Miraculously, the prototype comes together in only a few days! +1

Overlap: does the idea overlap with any other idea in functionality, appearance, implementation (the way it gets into the game), etc.? Remember, it's always a great idea to find some overlap with other ideas!

- + The idea is completely unique and requires all new assets. -1
- + The idea has overlap with one other major gameplay mechanic or system. 0
- + The idea has considerable overlap with multiple gameplay mechanics and systems. +1

Resources: how many people will it take to see your idea to fruition? Can you do it by yourself? Do you need programmers or artists or other designers or outsourcing?

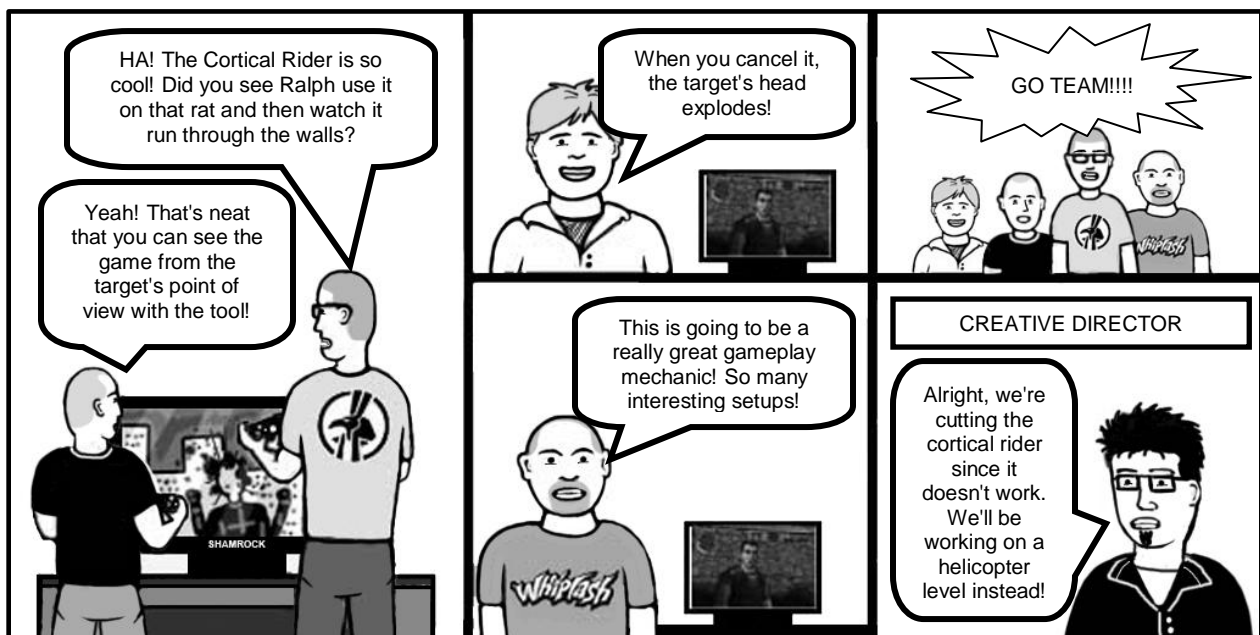
- + As the idea goes into development, you realize the entire team needs to get involved. -1
- + The prototype needs the lead programmer, gameplay programmer, a visual effects artist, and a couple of modelers to create. 0
- + The idea will work with only your gameplay programmer and modeler's involvement. +1

OK, time to add your numbers and see where you're at in relation to scope. A -5 is really bad due to a combination of all the factors above. Think about it. You have an idea that's never been done before and nobody seems to understand it (although you expect them to work on it) and it will take an unknown amount of time to do it with no benefit for any other gameplay elements. Ouch!! A 0 indicates an idea that has a good chance of succeeding with some minimal amount of risk. In other words, pretty much par for the course in the video game industry! A +5 is a near miracle that happens when the planets align and the team is all perfectly rested and all the technology for the game is perfect and bug-free....

Scope Reductions

Every game goes through some degree of scope reduction. There are times when it's absolutely crucial because the project is running out of time or resources or a feature just doesn't work. There are other times when it's a preemptive cut to account for an imbalance in the amount of content in the game. There are also times when cuts to scope are ill advised or even unnecessary. How about some examples of all of these?

Project Snowblind: some creative directors are inspirational leaders while others are like house guests who stay a little too long, cause arguments among the roommates, and leave the place a mess on their way out. The Project Snowblind creative director probably had the best of intentions, but his design decisions mostly seemed arbitrary because he rarely discussed any of them with the design team before acting on them. In relation to scope, he made two decisions that weren't great for the project. The first was cutting the number of different levels, but not the total number of levels in the game. Because of this, the player ended up having to revisit one of the least interesting levels 4 or 5 different times with very little to distinguish between each visit. This proved to be a huge design challenge even though there was some amount of savings on the art side. The other scope cut resulted in the loss of a single tool which happened to be one of the more interesting gameplay mechanics in the game:



Drawn to Life the Next Chapter: when Planet Moon Studios lost a big project, their Plan B project (a much smaller title in development with a separate team) became their Plan A project with a few more resources, but, unfortunately, no extra time on the schedule to get it done. The game had a very solid prototype and design plan, but a lot of tech was missing and the number and variety of levels that needed to be created was too far ambitious for the size of the design team. About three quarters of the way through development, the team's management had a 2 day summit where they went through every aspect of the game and made cuts to everything from individual gameplay features to entire levels. Some of the decisions were very difficult and resulted in some heated exchanges, but every cut was made in terms of the impact to both the game and the team. Oh, and there was also some really amazing catered sushi! Coming out of the meeting, the team still had a ton of work to get done, but the scope of the work was far more reasonable and resulted in the game getting finished on time and on budget.

Sly Cooper 4 Thieves in Time: Sly Cooper suffered from a somewhat rare situation where there was too much content and parts of the game were too long (gameplay-wise) as a result. This led to cases of uneven pacing where some levels felt longer than others and some missions felt padded. Cuts were made in order to balance out the overall experience: the Japan Geisha House was cut in half eliminating a secondary path, a Bentley helicopter mission in the Wild West was cut completely, and a Bentley boat mission in England was cut due to length and an overall lack of time to polish. The nice thing about all these scope changes were that they were the direct result of the designers, creative director, and Sony producers watching hours of focused play testing, so none of the cuts were a surprise and all of them were totally warranted.

There are many lessons to be pulled from all of these examples, but the most important takeaways are these:

- Team management should always try to involve the team in scope changes so that cuts and additions are not complete surprises.
- Cuts should be made with as little personal bias as possible. They should not come across as vindictive or petty.
- Cuts should benefit all aspects of production and not favor one department over another.
- Completed features should generally not be cut unless they just don't work.
- If the entire team thinks a particular cut is a bad idea, it's a good idea to reassess the cut.