Gamasutra: Kevin Wong's Blog - Vanishing Point Postmortem

The following blog post, unless otherwise noted, was written by a member of Gamasutra's community.

The thoughts and opinions expressed are those of the writer and not Gamasutra or its parent company.

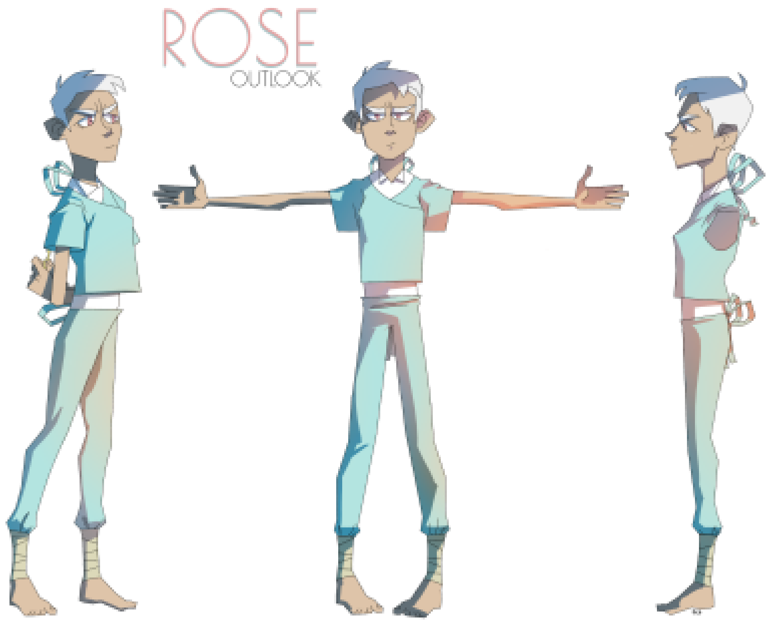
This article is ***long****, you can read a formatted, image-free PDF version here. You can also view the original post at kevinjameswong.com*

Through a chance encounter last year, I ended up working as the lead designer for the 2015 USC Advanced Games Project *Vanishing Point*. Aside from maybe Chambara, this was one of the larger projects I've worked on. It was a first-person puzzle platformer where players can manipulate the size and mass of objects in the environment to solve puzzles through a certain kind of forced-perspective, much in the same vein as *The Museum of Simulation Technology* and *Depth.*

It was challenging working with the game's theme and narrative, which attempted to subvert ableist tropes regarding the depiction of mental health issues in popular media, but tripped over itself and either simply averted or reinforced them.

Many games enforce a certain kind of normativity through their harmful depiction of the mentally ill, and use "insanity" as a way to exoticize a villain and vilify them as "criminally insane". Characters like *Far Cry 3's* Vaas and Batman's Joker reinforce that trope. Mental institutions as settings are not better off either, and often misuse the politically inflammatory implications of a legacy of mistreatment as a way to create an oppressive and hostile atmosphere without critically meditating on what that meant.

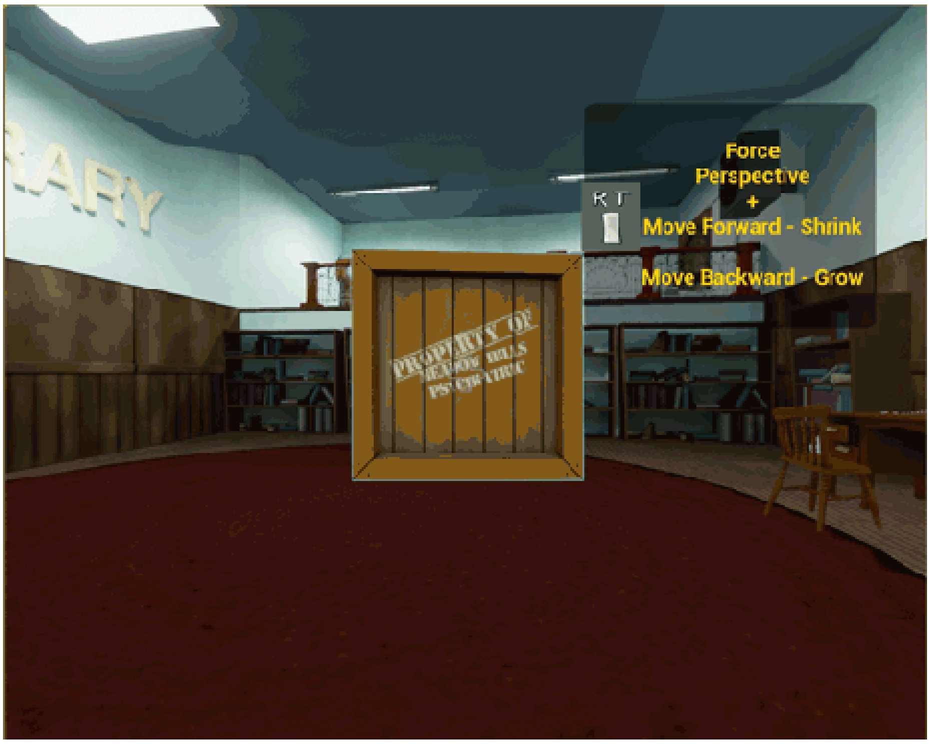
While some games might try to tackle mental illness in a more nuanced manner, like *Metal Gear Solid 4*'s handling of PTSD and *Papo y Yo*'s depiction of addiction, more often than not, lazy writing breeds monsters. The creative direction behind *Vanishing Point* very explicitly wanted to create a world that did not use problematic tropes with heroic characters that succeed with their conditions.



The game’s protagonist, Rose

The game’s protagonist was a woman suffering from a condition that affected her perception of scale who voluntarily seeks treatment at an institute called Meadow Hills. A doctor there creates a magical device that grants her the supernatural power to physically manifest that condition by scaling objects.

I think a lot of the focus of what *Vanishing Point* did not say detracted from the focus on what it could say. So the thematic core of what the game ended up doing lacked a powerful thesis, and ended up reinscribing ableist power structures in unintended ways by pursuing the fantastic rather than dealing with stuff closer to home that affects our own lives like PTSD, depression, and anxiety.



An early example of the scaling mechanic.

The game’s design was part of that. If a game has a very intentional thesis, its argument must be communicated through its every aspect, otherwise you end up with a flaccid ludonarrative dissonance, and *Vanishing Point* lacked the laser-precise direction to unite its constituent parts into something thematic, relevant, and positive.

Nonetheless, I speak only from the perspective of a lead designer who came onto the project later, after many of the core aspects of the game were settled. I was not in charge of the creative direction of the game, and my focus was spent only on exploring a mechanic. So I should speak mostly of what my experience was on the project and talk about what went right, what went wrong, and what I learned along the way.

WHAT WENT RIGHT 1. PERSONAL GROWTH FOR TEAM MEMBERS

One of the major goals for *Vanishing Point* was to help team members achieve personal goals and develop themselves as game makers through their participation on the project. To do this, much of recruitment focused on finding people who would benefit the most from participating in *Vanishing Point*. In the end, I think we were able to help a lot of the people on the team. Team members ended up with jobs they wanted, leadership on later projects, great friends and collaborators, and a clearer focus on what personally excites them about working on games.

2. DOWNSCOPING HELPED WITH FOCUS

Reducing the scope of the game by removing some player-controlled abilities and eliminating the levels that would use them allowed us to more fully explore a single mechanic. A lofty and ambitious scope would have spread the team too thin and forced us to rush levels without the critical thought and care needed to make those levels interesting, fun, and accessible. This downscoping also helped with morale, and building this game felt like less of an insurmountable and dehumanizing task that infiltrated our nightmares and more like a challenging project would push our skills as game developers and bring the best out of us.

A locker room in the game’s world

3. AGILE LEVEL DESIGN TOOLS

We created a certain kind of “tile editor” for *Vanishing Point* where levels were constructed of 400×400 square tiles which could have textures and special properties applied to them. Building a map for *Vanishing Point* was agile and did not rely on very specific modeling and architecting, and changes in level design can be tested and saved or reverted very fast. This allowed us to create better levels with consistent rules, as well as construct secret areas with hidden rewards. The downside to this workflow is that every map ended up with this very clinical, boxy feel that made the tiled construction of the world very obvious.

4. EVOLUTION IN RESPONSE TO PLAYTESTING

Playtesting is important for all games. For puzzle games like *Vanishing Point*, where all puzzles are heavily authored, emergent behavior can cause situations that could mislead players into false conclusions. Frequent playtests allowed us to adjust puzzles to better dispense essential information and tools, and allow progress in those puzzles to be more clearly demarcated.

That playtesting also allowed us to confidently make deeper changes to how our mechanic worked. Early in the project, we realized that everything about how the mechanic worked simply was not working, and revised it from the ground up. This forced us to throw out everything we built prior to that, but we ended up with a tighter game that was easier for players to learn and granted designers more room to be creative.

CAT MYNC BOLT
CAT

Stasis fields make objects placed in them static.

5. MECHANIC WAS EXPLORED WELL

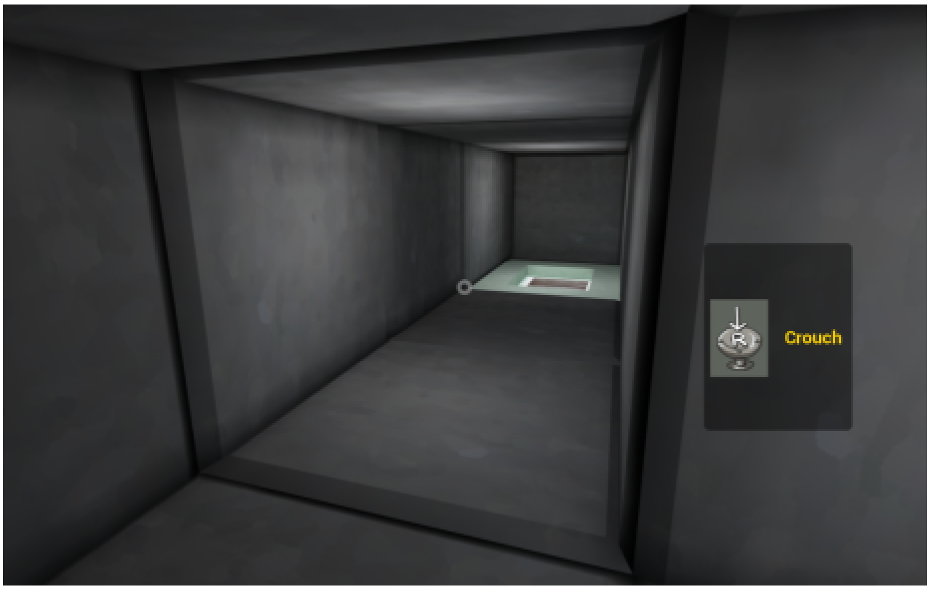
There is a total of 20 encounters in the entirety of *Vanishing Point*, and while the ability to manipulate objects' mass and scale does not immediately invite visions of colossal spaces for play like *Portal* or *Antichamber's* core mechanic, I think we were able to create some interesting play by continuously mixing upon that core mechanic. Objects like stasis fields, mass-dependent buttons, destructible meshes brought out the specific properties of the scaling mechanic. UI and usability changes helped make the game as understandable as possible to players playing the game in a gallery exhibition, and efforts were made to make the game accessible to disabled players.

WHAT WENT WRONG1. VAGUE CREATIVE DIRECTION

The creative direction for the project was unclear throughout its first half, aside from a high-level social goal; questions of tone, mood, theme, and purpose were left very ambiguous. *Vanishing Point* meandered from being a superhero origin story, to a gritty drama, to a magical-girl adventure. There's a well-wishing intentionality where you purposefully leave the creative direction for a project vague as to let others fill in the blanks and share creative ownership on a project, and that comes from a good place. The issue with that is, on the receiving end, it often feels like being handed a blank sheet of paper and asked to draw something that fulfills a set of vague parameters. An absence of constraints invites aimless meandering, and not vesting the power to have the final say in someone makes decisive action difficult.

2. EXPLORATORY PHASE LASTED TOO LONG

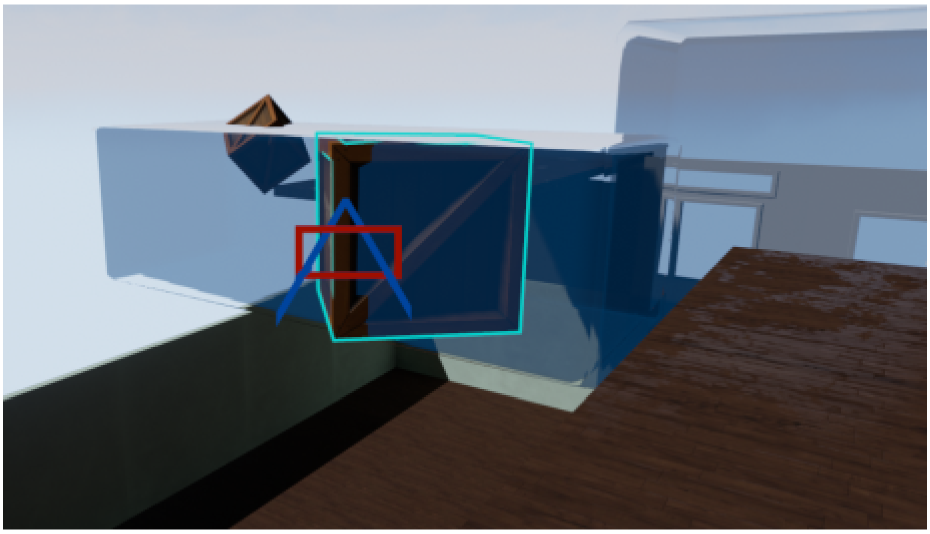
Development on *Vanishing Point* originated in Unity but switched to Unreal 4 to give exciting challenges to engineers and permit use of its powerful lighting engine and blueprint system. This was not the best decision for the game or its players because developers had to spend the first several months learning a new toolset and transitioning to an unfamiliar workflow. This slowed development greatly, and entire weeks were spent bumbling around the engine trying to figure out how to make the game with these tools. That delayed that early, exploratory development phase where you are figuring out how to bring out the fun in what you have, prototyping new mechanics and discovering what works lasted all the way to the end of the first semester. This exploratory phase is essential, and *Vanishing Point* would have failed spectacularly without it. Everything that was presented at Spring Demo Day last week was constructed in a single semester because all the work from the exploratory phase was not of a level of quality that we were comfortable exhibiting.



You can crawl through vents.

3. ART PIPELINE EFFICIENCY

One of the things that we struggled with was the dearth of 3D artists and animators local to USC. Much of *Vanishing Point*'s 3D art had to be outsourced to volunteer freelancers or non-artists. Thus, asset production was slow and fraught with errors. A single world prop needs to be modeled, unwrapped, textured, and cleaned in order to be ready to be shown inside a game engine. There are special parameters that have to be fulfilled in order for a 3D prop to be usable in *Vanishing Point,* the pivot point must be exactly at the center of an object and must be built to translate exactly from Maya units to Unreal units. If any of these parameters are not met, that asset must be sent back to whoever built it. With different individuals working at each step of this process, it could take a while for art assets to be ready for integration in the game.



The tile based design workflow is particularly obvious in this screenshot.

4. DIP IN TEAM MORALE

Losing morale sucks. Crunching under a tight deadline when you and the people around you lack it feels like a tragic death march towards an inevitable defeat. In our first semester, we ended up in this state of mind as our major Winter Demo Day deadline loomed by. Folks from the game industry were going to show up at this event, and lots of people were counting on an impressive exhibition to secure jobs. Back then, our game was simply not ready for exhibition, and morale was low and we struggled to want to work on the game. As a lead, I was responsible for perpetuating this atmosphere by feeling this way myself. Getting stuck in a rut where you and your team are consistently in this state of mind is an obvious sign that you need a break.

Furthermore, since most of our team was made out of volunteers who were not taking the AGP class for credit, we had no way to keep volunteer members accountable for completing their tasks on time to a high standard since there was no consequence for receiving a red on a task. When a deadline was missed on an important task, it ended up impacting morale by causing delays and blocks.

5. CORE MECHANIC NOT PROTOTYPED ENOUGH.

Early prototypes of the game were mechanically identical to *The Museum of Simulation Technology*, where the scale of an object was relative to the object's position in relation to the player. Over the summer, the mechanic was changed to work by the position of the player in relation to the object. Last summer, I was out in Dundee working on *Chambara*, and was not involved in the development of this new mechanic and did not have any input in how it function. When I had it in my hands at start of fall and was asked to build levels and new mechanics out of it, I never really felt the mechanic granted me enough room to be creative with design. It was a bit of a shallow foundation to build off of for such a substantial project, but we ended up doing the best we could.

VANISHING POINT
CREATED BY MAX COHEN & SAM SANDWEISS TEAM LEAD MAX COHEN LEAD PRODUCER SAM SANDWEISS LEAD DESIGNER KEVIN WONG ENGINEER NICK WONG LEAD ARTIST VICTORIA CUTHBERTSON
LEAD WRITER COOPER SURRETT LEAD USABILITY MATHEW SCHACHER FALL LEAD SOUND ELLE CALLAHAN SPRING LEAD SOUND AARON REHS PRODUCER JACK SEMDEN COMPOSER AARON REHS
DESIGNERS CATHY TRANG MATHEW SCHACHER BRITTANY BEDLEMAN ROBERT NEIR GEORGE NYLONAS MAX LAVOXA ENGINEERS BRIAN FABIAN FINTAN O'GRADY MICHAEL GOULIS JOSHUA LEE TONY CHUN
ARTISTS JOSHUA LEE FUNKSTER SCERBAK SEAN TROEHLER REMY EARG JANA "OTTO" MAIER MAX CALLAHAN VOICES BY CHRIS DORMAN CAROLINE DINGWALL CAITLIN LEON KATE CLARKE
COLE FINNEY KRISTIN MORRIS SOUND DAVID DEEDMANIA ADVISORS CHANEL SUMMERS JONATHAN COOPER REALM LOVEJOY
USC Games

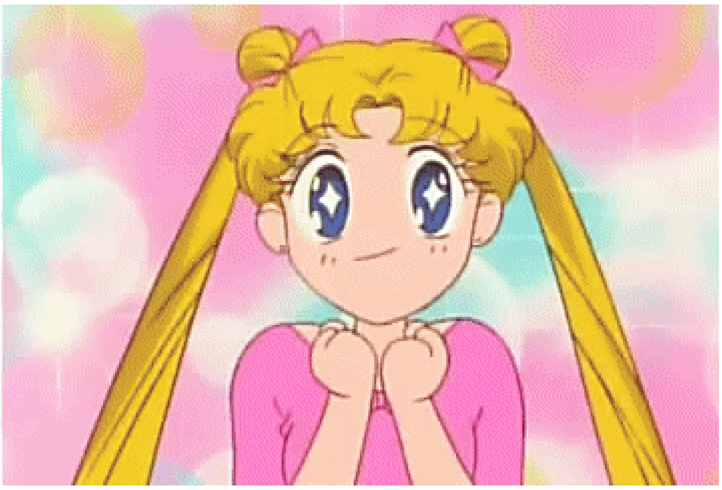
Game Poster

PERSONAL MEDITATIONS1. HOW TO SAY NO

What makes team members immediately happy and what's right for the project are not always the same thing. Sometimes you have to say no to proposals and ideas that would come at the expense of the overall project even if folks want to do things their way. It hurts to say no to someone who, say, wants to work on a complex photorealistic shader or texture objects with selfies, but saying yes to everything arbitrarily can detract from focus and cohesion, and balloon the timeline by having people focus on tasks of lesser importance.

2. PROJECT MANAGEMENT IS GREAT

Task lists and effective project management does not only make things operate smoothly and reduce blocks, but it also serves to motivate people by making progress visibly quantifiable and boost team morale. Just because you are intimately familiar with what needs to be done with a game and the priority of each individual task does not entail that everyone else is.



This was my emotional state working on the game.

3. BEING HONEST TO YOURSELF/OTHERS

Be honest and communicative about your emotional state. Its easy to get into a mindset where you hide how stressed out you are, but being open about that can save you from unneeded suffering and prevent you from behaving badly. As a team lead, your behavior sets the tone for the people working with you, and a crappy, toxic attitude spreads around and hurts people. There was a phase where we hated the very Platonic essence of the project, and that manifested in how other people behaved. Everyone has a different source of burnout, and being aware and accommodating will create more peaceful projects.

It is also important to be honest to yourself about who you are and what you can do. It is very easy to get into patterns of harmful overwork by refusing to admit to yourself that you do not have to always be the one to charge into every fight.

4. STAND UP FOR WHAT’S RIGHT

Doesn’t only apply to game development, but life too. It takes courage to speak out when something wrong is being done, but that initiative is important. Everyone wants to see the game succeed, and people welcome critical feedback about their decisions.

Sailor Moon is a failure.
Flat chested, a flighty idiot,
you're about the worst there is!

Same.

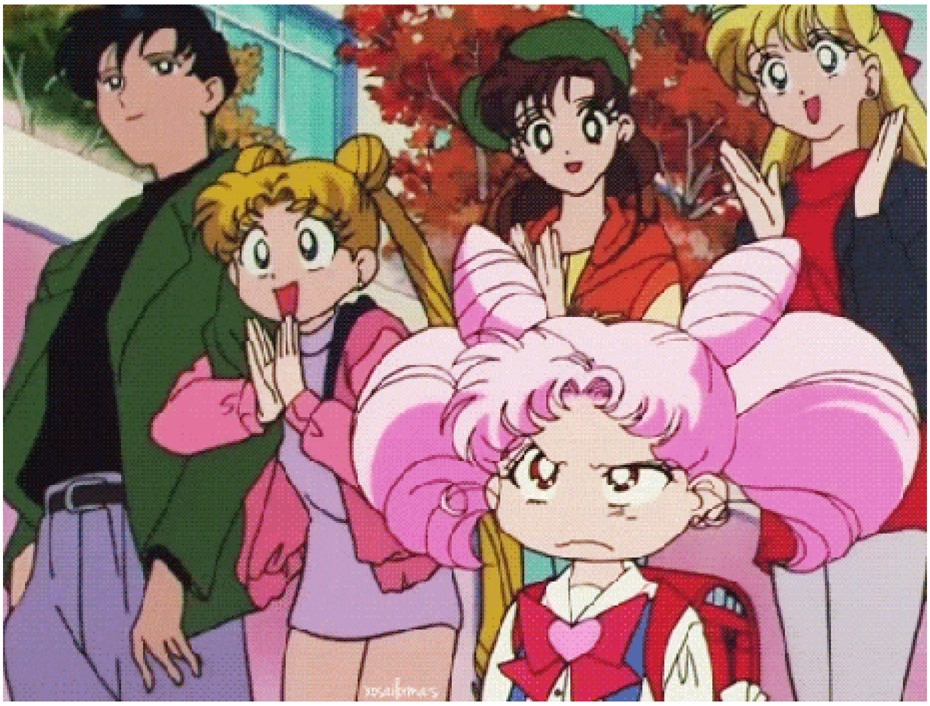
5. HAVE SELF-WORTH

It is useful to be your own worst critic, but don't forget to be a fan and advocate too. It is easy to get into a mindset of hypercritical self-skewering and make that fester into self-hatred and feelings of incompetency, failure, and worthlessness. I feel this way a lot. This is very harmful to creative people. There are things to be proud of, and a singular fixation on only the negative aspects of an experience breeds toxic attitudes.

There comes a point in a developer's career where they transition from feeling proud and accomplished by the fact that they were able to make a game to feeling dejected and despondent that the games they make are not the best they feel they could be. That's a tough transition to make, and I still feel its sting even as I near my fifth year as a game developer.

6. DON'T OVERASSIGN/SET LOW EXPECTATIONS

You're working with real people with rich lives outside of game development, and its important to not overtask people and be intrusive. At the same time, don't underassign work. People want to use their talents, be expressive, and feel valuable to the team, and doing simplistic, boring tasks is patronizing and wrong. The worst you can do is waste people's time, especially when they are specifically here because they want to work on a particular project.



Sometimes your players like your game more than you do.

7. YOU'RE HERE FOR THE PLAYER

I've talked mostly about what goes on with internal team dynamics here, but at the end of the day, your ultimate goal is to create the optimal play experience for your audience. There are self-serving approaches where developers build features or mechanics for the sole purpose of making the game more interesting for them, building alienating features for fun, or creating busywork just to keep people occupied. This approach is right for certain kinds of games, but not for ones that aspire to be welcoming to players and hold them in high regard. I've heard a metaphor that preparing the experience of a game is like preparing the experience of a meal, you have to make the experience a positive one for them by ensuring every aspect of the experience serve the pleasure of their experience. Make games as if you want people to have a good time playing, this informed every single decision I made on *Vanishing Point.*

8. DISTANCE CREATES PERSPECTIVE

Taking a break from a project or working on it less gives perspective and lets you come back with the focus to make better decisions. A bit of time where you don't touch the project at all does wonders.

This doesn't really capture exactly what I'm trying to say here, but its close.

9. I STILL LOVE MAKING GAMES

Game design is a delightfully stressful emotional rollercoaster. It is hard, often grueling with highest highs and lowest lows, and I love every moment of the process. I'm truly blessed to have had the opportunity to work on this and every other game I've been a part of, and I'm immensely proud of and impressed by the colossal talent of folks I've worked with.

And that's all I have to say for tonight. There is more that I could say, but the next game awaits.

Production Length: 11 months
Staff: 27 Overall, at greatest 23, at least 17
Software Used: Unreal 4, Autodesk Maya, MotionBuilder, Fraps, Microsoft Visual Studio, Blender, Audacity,
Budget: 600+ USD

[Play it here]

Related Jobs

ten4
recruitment agency

Ten4 — San Francisco, California, United States
[06.17.16]
[VR / Virtual Reality Creative Director]

Industrial Light & Magic — San Francisco, California, United States

[06.16.16]

ILMxLAB Environment Artist

Rainmaker Entertainment — Vancouver, British Columbia, Canada

[06.16.16]

Senior Gaming Engine Pipeline TD

Disruptor Beam — Framingham, Massachusetts, United States

[06.16.16]

Design Manager