Hi all,

I'm a local Application Developer with career experience in programming and databases. I'm coming off of a very long-term training period for learning new technology and expanding my skill sets to upgrade to the newest programming languages, such as Rust, as well as learn technologies relevant to the entertainment industry such as numerous aspects of 3D development including programming and content creation, VR/AR/XR, music theory, music composition, sound effects, AI/NN, and video editing. I wasn't expecting my training period to last more than a year, but technology raced, and I was transitioning to a new area of computers - entertainment, which is a huge transition, in and of itself. On top of that, Rust was a massive train. I'll be the first to admit, there's a lot I haven't touched yet in 3D, and at the same point in time, my mind is so full of technology that it's hard to keep it in. This was a 6-year train for me with only 2 months off total. I plan on offering solutions directly to small-to-medium sized businesses.

To get a digital copy of this document online, go to the following url: http://github.com/JeffGillin/Pitch . On that page, you will find a link to a PDF document. Click into that link. It will open up a page with a viewer where you can read the document. However, the document became too long to read the full thing in the github viewer. They only allow X number of pages before it stops loading more. So you need to download the PDF which you can do using a download icon/button on the top right of the viewer page. It looks like a down arrow. Remember that on github, you have to click the pdf to open the viewer page.

I'm seeking to develop projects for businesses who are interested in VR headset entertainment solutions in order to increase your customer base, as well as maintain existing customers. I'm also willing to create more typical computer gaming or "3d sim/experiences", meaning ones that can be played without a VR headset (including installed applications or web applications to desktop computers, laptops, phone, tablet), because it still is early in the days of the VR headsets. It doesn't necessarily hurt to be ahead of the next generation of devices. Your game or experience will automatically improve with the next device as the passthrough improves. There are also options on the passthrough which I'll discuss throughout this document, but part of that is getting the Quest Pro so that you get a better camera passthrough experience, though it does cost twice as much. That said, passthrough only matters for AR/XR solutions.

Initially, I planned on keeping this pitch as simple as possible, however, it's the

type of thing that brings forth lots of questions, so I decided to attempt to answer most questions I felt you'd likely be thinking if this was only a 1 page pitch. This VR/AR/XR technology is new, and most people are just going to wonder why I even came to them. I believe that it's relevant for most businesses, like any other entertainment venue - to get people in, or talking, or viewing your weekly discounts. Most still don't see the types of entertainment that could be created at this point. As more start coming out, more ideas will pop into your mind as far as what could be done for your business.

The technology is pretty decent now, but the next version is where things really start to get more interesting. That's just where the technology is at this point in time. On the other hand, the VR headset solution could be very good at this point in time if performance considerations are kept in mind while developing. Some trade-offs have to be made, but if you do so, then you can have an excellent, fun game, right now.

Most of this document is written up to be geared towards a VR headset project, whether Virtual Reality or Mixed Reality. If you want to do a standard computer game, most of this document will still be valid in answering a lot of your questions including costs, which will be the same.

What are you offering up, buddy?

My current and discounted offer is a 3-month contract for $24k - $30k for a VR/AR/XR entertainment solution, being either a game, or experience. After this initial offer, my rate will likely be closer to $40k per 3-month period going forward, so this is quite a discount actually, and likely 5x less than you'd get from any company. It's still really early for this technology even though the parts and pieces are mostly there from a development standpoint. From a device standpoint, it's okay at this point, but they will get much better. That said, the Meta Quest 3 and the Meta Quest Pro are fairly decent at this point. The new budget friendly Meta Quest 3S is also an option ($299). They aren't near as good as the one from Apple, but not many are buying that due to its price.

The difference in price simply depends on whether or not you received this offer directly from me, or someone else. I will likely be giving a few people the opportunity to go out and sell my services if they want to. If you receive this pitch from a sales partner, then the offer will be for $30k as I will be giving them $6k for the sales work, or you could directly pay them that sales commission if that works out better for you for any reason.

If you yourself are not interested, but think you might know someone who would be, feel free to become that sales partner and make the $6k for the hookup (after 2 payments into the project).

Can you give me a technology / terminology primer?

Yes, and it's a little bit of a confusing story. They like it that way. It keeps your head spinning… The confusion they laid down is adding in this term "Extended Reality", which is said to be a term that encompasses all the various top level technologies used in VR headsets - being Virtual Reality, Augmented Reality, and Mixed Reality. The term Extended Reality started taking the acronym XR which was previously used for Mixed Reality. Some people keep saying XR for mixed reality and others started saying and writing MR for Mixed Reality.

So, let's get to the terms:

Virtual Reality (VR) - this means that you're immersed inside the game and there is no reality coming through. In other words, the passthrough camera is not used. There's no problem with going with this type of game at all. It's more geared for indoor gaming where a person is sitting down or standing mostly in place. If you plan on having the people play a game that they would play from home, then this is the type of game that you should choose. Virtual Reality adds a lot of enhancements as compared with a regular computer game, because of how you interact with things much more realistically than a standard game controller, allowing you to swing your arms to hit something, for instance. It's not quite as cool as Mixed Reality in some ways, but it has the advantage that you can play from anywhere, and that you are totally immersed in the experience.

Note that when in VR mode, you can have the camera (real world) view display as a small portion of your view, like the "picture-in-picture" technology in televisions, such that you can still see what's going on around you even if most of your screen is virtual. We could also design the app to allow the user to "touch" that portion of their virtual screen, expanding it to a large size that allows them to quickly interact with those around them without having to take off the headset. This will be important for using VR at a restaurant, in a waiting room, or similar types of locations where you might be still interacting with those around you.

Augmented Reality (AR) - this technology is pretty much like "Pokemon Go". There are things being displayed on top of the camera video on your screen, and

they show up based on your location, either inside a structure, or outside in the real world. The items displayed are often based on your GPS coordinates, and they might display additional information about the buildings and historic sites around you. This can be done on a VR headset or even just a phone. But the phone experience isn't so great, so it's mostly going to be a headset thing going forward. The AR for phones is really kind of a starter market to allow developers to learn the technology while the headsets are still being developed. I'm not really doing AR development right now because I'm focusing on entertainment solutions. It's mostly for cities, tourist attractions, and stuff like that. It's not really for entertainment, but for displaying information about real world things, and/or interacting in some way like booking a night's stay at a hotel you're driving by simply by clicking a button that appears on the hotel, which also shows you a price.

Mixed Reality (XR or MR) - mostly for outdoor gaming, though it can be for indoor gaming as well if there is enough space. This allows reality (camera passthrough) to display in your game, but, of course, also allows you to add in additional rendered 3d objects and shadows that really appear to be part of the reality. The game objects don't just display on top of the camera feed, but rather, things in your game, such as a character can run and hide behind your real world couch (passthrough), for instance. When done properly, it will be close to hallucinating… If the lighting and shadows are done correctly - it will be hard to tell what is real and what isn't. When I use the term XR throughout this document, I'm referring to Mixed Reality. Think of users running through a field playing a video game and you'll have a better idea. Chasing zombies out in the field for instance, and maybe hitting them with a cucumber (digital). Or skateboarding while playing a video game. It's only limited to one's imagination. There truly is an endless array of types of things you can do with Mixed Reality.

Extended Reality (XR or it doesn't really exist) - this is an "all encompassing term" meant to mean that you're basically wearing a VR headset or glasses and that you are having some experience that at least enhances reality (AR, and Mixed Reality) or totally destroys reality (VR).

But Extended sure sounds like Augmented, doesn't it. And it sure seems to have the feel of both Augmented Reality and Mixed Reality. We're extending reality. But yet, they made it an all encompassing term that they also threw in as part of the VR/AR/XR set of acronyms… Clearly intentional. As someone that

works in this field, let me tell you, this is the confusion they lay down.

Here's the thing. They use the set of acronyms listed as VR/AR/XR all over the place. If extended reality is an encompassing term, then it doesn't make sense to list it out in the set, does it? It should be VR/AR/MR if they're going to list all three types.

Why go with VR/XR when only 20-25% of US households have a VR headset?

Because it can be a much greater experience, and some might go out and buy one in order to be part of the fun. And because, if you limit the amount of guests that use it at once, you can simply buy the devices needed. It will be a good intro for them anyways into this new world of using VR headsets regularly.

That said, non-VR headset devices have a much bigger reach right now, so I wouldn't blame you if you wanted to go that route if you are interested in some game or experience, if even just a lowly computer game. In that case, I would definitely use the Bevy game engine which is really good, and has excellent web support for browser based games, which have a big advantage of being available to everyone without needing any installation.

Which one is right for our business:

1) **VR** is good for storefront type businesses because your customers won't be there long and can play at home - likely at designated times in order to have teams play against each other. The downside is that only 20-25% of US households have a VR headset (typically the Quest). Note that VR isn't a bad option for people in a waiting room, but that's true of XR as well. See the section on using disposable inserts for maintaining cleanliness of the headsets between uses. In addition to the inserts, there could be alcohol wipes for the headset straps - just ensure they don't use them on the screen.

2) **Mixed reality** is good for businesses where there is a decent amount of property to create an outdoor game. For example, every Sunday at 6PM we play this game on our property if you want to come play. There might be prizes (or not). However you want to do it. That said, restaurants could certainly do **inside** XR games as well. Lastly, people who have to sit in a waiting room really might enjoy a good Mixed Reality game right there, such as changing other guests' faces, or maybe throwing virtual paper balls at some virtual roaches crawling on the real walls. The entire wall or ceiling could appear to be completely gone and

replaced with some 3d environment that you could be interacting with. This is an XR technique that will likely be very common, and involves rendering a 3d world to a texture, then displaying that texture on a virtual plane (that gets placed in front of the wall or ceiling you are virtualizing/replacing).

3) **Non-VR headset games** - Any type of game that does not involve wearing a headset, such as desktop, laptop, phone, tablet. This would be good for any business. It has the largest reach because most households have a computer of some sort - either a desktop computer, a laptop, a tablet, or a phone. Note that you would have to choose one of the 3 scenarios below (because they are each different).

a) web deployment - could target all devices, but depending on the type of game might only really be playable on large screens.
b) installable application on desktop/laptop (desktop operating system)
c) installable application on tablet/phone (mobile operating system)

In other words, I'm not going to be attempting to target all these devices. We're going to go with the one that works best for your solution. If it's for the web, I may even limit the browser type to a particular browser, which would likely be either Chrome or Firefox. Firefox is a lot better these days than it used to be - it's right up there with Chrome now. For the most part, I would likely ensure it works in both Chrome and Firefox, but there could be cases where it's limited to only one of those if we use functionality that's available in one and not the other.

Note that we can include the use of a gamepad for laptop or desktop computer games. It could either be required or optional. This would likely be very useful for some games as the mouse and keyboard do not offer the best gaming experience.

The downside to this standard computer game is that they don't get the same experience that they would in a VR headset game where you are immersed in it, or in a mixed reality game. I didn't like playing standard computer games much, but I do feel like I would play the VR/XR games though I haven't had much time yet.

What about different screen sizes if we go with a non-VR solution?

Basically, the answer is that I'm going to require that any non-VR solution is either for phone or for big screen (desktop, laptop, tablet). For the most part, I'd say, let's NOT go with phone. There are just too many possible issues with phone deployment due to there being so many versions of the phones - differences in software versions, provider constraints, connection reliability.

I'm not setting out with a goal of customizing solutions per screen size. At this price range, we'll develop for a target screen size and have to accept how it resizes by default for different screen sizes and we'll develop for the most typical screen sizes (desktop / laptop / tablet). So, do expect that there will be some that it won't work for if they are playing at home, but we'll keep that number low - under 5% if they are willing to upgrade their operating system and/or browser if needed. Whether or not they would need to depends on the specific set of technologies chosen for a solution. This issue is mostly only specific to non-VR headset games - you can't count on everyone's computer working for them.

Can we start out as a computer game and move to VR later?

Yes, for some types of games, it would be possible to start as a regular 3D game and have it converted by myself or someone else to a VR game down the road. This would be on a case-by-case basis. That doesn't mean it needs to cost a lot. It means there are considerations unique to each game - for some games it could cost a lot, and for others, not much.

Tell me more about Mixed Reality technology.

When your game characters start running behind real world trees, things get interesting… Mixed Reality really opens up a whole other type of gaming experience where someone could be playing a real world soccer game, for instance, and have their VR headset on and doing additional things, such as if they score a goal AND the ball goes through something on the VR headset, they get additional points. I know that's not a great example, but I'm just trying to give you a feel of the technology. It's basically like hallucinating in the real world. You don't have to be limited to a certain "playing area" either like in VR. You can wander around anywhere seeing mostly the real world, so you have to think in terms of the "real world" with game objects mixed into the camera feed. These objects don't just show "on top of" other things. They can be partially or fully hidden behind them. The objects really integrate with the environment (or should if developed well) rather than simply "floating" in front like you may have been

accustomed to from Pokemon Go. As of the last time I checked, you couldn't get internet through the phone even when connected to the phone from the headset using it as a wireless access point.

Why do you think we would go with you over a development team?

It's likely pretty hard to see, but believe me, I will beat out a team of 5 in terms of quality for 5-10 times lower price. By "team of 5", I'm referring to a typical team which has 3 devs, 1 QA, and 1 manager, where the devs are likely a programmer, a digital 3D artist/animator, or audio producer. I'm not suggesting you can't get better than me. I am suggesting you won't get better for under $125,000, which is five times my asking price for this introductory pricing model, and likely not for under $250,000 because of costs associated with putting together a team (unless someone comes to you).

The indie developer in the 3D field brings something that the entire team lacks if they don't have any indie developer. Someone who understands the full picture. You have to think as a 3d film producer, not as a developer. This is what you're learning when you learn a 3d content creation and animation software application like Blender - you're learning art and film production. You're not just learning how to do 3d content creation. You're learning various art forms. You're learning lighting, animation, scene composition, compositing, and from a video/film production standpoint - one aspect of film production, anyways. It's the same thing with video editing tools like DaVinci Resolve, which is the video editing software that I chose to learn. Just like with learning Blender, you're learning not just a tool, but the techniques of film production. I'm not saying I know them all - I don't, but I have picked up quite a bit, and can definitely put together some good video content for games.

I've spent many years learning Blender, and understanding things from that side - film production. My DaVinci resolve skills are much lower than my Blender skills, but I can do a pretty decent amount at this point. I understand it fairly well at this point, and picked up various film production techniques during training. Sure, I'm at a beginner-level in video editing, but I really dug into it. I learned the ins and outs of video editing and how to use Resolve. I understand how to use the Fusion page for composing custom effects and transitions, and also how to use the Color Grading page, as well as the basics of color management. A professional video editor would certainly do much better than myself in video editing - no doubt, but you won't find one on a project at my price, and we won't need professional-level video editing. The reasons why I would do a much better

job as a video editor than you may be otherwise inclined to think are 1) I do already have a solid understanding of DaVinci Resolve at this point in time, and 2) I'll be purchasing Resolve plugins that are most relevant for the project, if video is used in the project, which is likely. And by most relevant, I'm just saying that there are lots of plugins available, and each is tailored to different types of video production. These plugins give you professional quality vfx, transitions, titling, sound effects, color grading, and more without having to do much work. So, making use of plugins alone will actually bring you to professional looking stuff pretty quickly. Will it be perfect? Maybe not. But for perfect, you are looking at 10x this price (and it probably won't be…). This will bring us a LOT further than you think even with me having only beginner-level video editing skills.

Now, someone who has been in the 3D field for a long time will have learned these concepts to some extent. But you can't just take any dev, and make them a 3D developer. It's a whole different thing. They won't understand a lick of it, even though the game engines are actually doing a lot for us. You still have to digest the field as a whole, which takes forever - for some reason. When you get there, it doesn't seem like it should have taken so long, but one side is that things are constantly evolving, which is why the training is so long. I've heard the same thing over and over - many give up at some point. There's just so many different aspects of it, you feel like the Milton Waddams character from Office Space trying to learn all of it. Once you get through the training, things start to clear up. Then you can tackle projects. Until that point, it's a very long train until you can hop into a 3D project and start working on it. Nobody in the field understands why it takes so long when they get there, but it seems to be that way for everyone.

Even though I'm not an expert at any of it, I do believe I will develop and deliver a solution better than a typical team of five. You almost **have to** have someone with the complete picture on the team, which is why you actually won't find many teams doing this type of work for small-to-medium sized businesses. Good luck if you go seek out one - it's a specialty field. The corporations doing game development would have a hard time finding an indie developer. That's why they became an indie developer - they don't want to work there. Most of the teams are completely dysfunctional, and that's not a hit to any company. It's just something that doesn't work. Too many people not understanding the others, and often not understanding what they are asking of them. Too much team frustration. The companies above want the teams to all work well together, of course, but that's not the reality of what actually seems to happen. It's not that it's a big fight or anything, it's just that it just doesn't work. Too many decisions need to be made.

Too many people are part of each of those decisions. Certainly does not work well for art…

The indie developer is thinking much more about usage of sound, about scene composition, about lighting, and about various effects more like an artist - a film producer, and not like a typical 3D video game developer. You see a MUCH different style. This is the style that I'll likely be inclined to bring unless you have a specific desire for another style. It works for some types of games, but not all. Most video game producers make the games look flashy, which doesn't really work. Part of that simply comes down to the types of techniques used in real-time rendering being different than in film production, but that is changing with advancements in computers. So, even a typical 3D developer doesn't necessarily do that good of a job putting a game together in a way that feels engaging in the same way that a movie can be. They aren't using lighting like an artist. They aren't composing, nor compositing like an artist. They aren't using sound like an artist. But that's what the artist and sound producer are on the team for, right? It would be amazing if they could truly climb up inside each other's heads - the programmer, the sound producer, and the animator/artist. But that's not what happens. The game developers drive the project, and they tell the artist what they need, and the audio person what to make. That's not art… I can imagine how much the artists feel strapped in this process. But the developers job is the one that is absolutely necessary otherwise there's no game created, so they are put centric in the development process, whereas the artist should be the center of the project. It's a challenge going either way, because it's also hard to put an artist in charge of a software project. But the animator/artist really needs to lead the design aspect, but they rarely get the chance to do so…

Some would say it would cost you $125,000 just to put together that team, and then another $125,000 in development costs. It depends on who you come across… Market forces will drive rates up (if we stay with capitalism anyways…), so expect that price to actually increase - crazy as that may sound. So, my offer is likely under 10x what you will actually pay down the road elsewhere.

How will your pricing change over time?

I don't know how long it will take me to ramp up to my eventual target pricing model of $40k per project, but likely within a year, with each project going to cost more than the others, as I will have proven to others that I can do quality work for this price. This $24k project price is an absolute minimum for this project. I do understand that it is costly and many won't be able to afford this. It's not exactly a

great economy out there right now for many. This will also be my last project at this rate. My next project will likely be above $30k.

What can we expect in terms of quality of the game?

I'm going for intermediate quality at a really good price. It doesn't mean I'm shooting for intermediate quality. It just means I can't be an expert at everything, nor get everything refined enough in a 3-month project to meet someone else's desires. I'm not actually seeking to tear my hair out in this project either, nor should I, and that doesn't imply laziness. You see my value immediately in the price difference. The value that I'm out here bringing to small-to-medium sized businesses, whereas nobody else is. This is why I'm taking over the management of it. It's not actually hard to develop solutions for people when done directly. When done indirectly, through a consulting agency, for instance, it's quite challenging actually, and not just for the developer, for the project as a whole. Most places make the very big mistake of "pressing against" their developers, thinking it's going to motivate them. It's a big mistake - not just for their business relationship - but for the project itself. That just ties the developers thoughts up on something other than the work. Now they're caught up in emotions, being pressured by someone driving the process who typically has a much easier role, and is simply pressuring them because they think it's going to better their own profits. It doesn't work for the developer - not at all. My way is cheap and works.

How is the VR quality at this point?

The "passthrough" quality (seeing the real world) in the Quest 3 is still coming along to be perfectly honest. There is a setting to get better passthrough, although that limits processing power for the game. It's a trade-off. Keep in mind that not all games have heavy calculations - many do not. But these are just the things that need to be worked out for a particular game - whether or not you'd want higher passthrough (camera) quality or you want to leave more compute power for the game itself. Not all games will need good quality camera passthrough. Think of an outdoor game intended for after sundown - then you could use a low quality passthrough setting and get high performance for game objects. Or for a daytime game, then you can use a higher passthrough quality setting but less game objects on the screen. These games are very much do-able right now if you manage it correctly and don't overdo it. You just have to make some consolidations as to full expectations. Some of the most popular

games didn't actually require lots of power. This is one of the reasons it's best to leave the development up to the developer because they're always thinking about the technical side of each aspect, and they know the path where they can meet expectations if left alone.

Could this be a drive-through experience?

The technology can be used on the road, yes, however, the quality of the connection is very poor when attempting to connect to the internet from the Quest 3 through the phone using the phone as a wireless access point. I can't seem to get it to work at all with my iphone. This might be a restriction of certain carriers, but I'd definitely say it's not worth using the phone's 4G/5G internet connection. If it does work at all, it's not likely to work with all phones or carriers.

That said, it doesn't prevent the GPS in a phone from being used by the Quest 3, but that does require using installed applications on the phone and headset device, and I wouldn't recommend it if this is intended to be run on the user's device versus one you purchase/own, and they wouldn't really want to install your application on their phone.

Possible Solutions:

1) Rent a bunch of wireless hotspots, which I believe should work. As an example, Sierra Wireless has some wireless hotspots that have GPS receivers and you can access that data through AT commands (Attention commands), or through an http interface. If you rent say 20 of these devices from a phone provider for a 2-month long event, that might be worth it for you. This is an area where I'd just have to go down the path and try to get it working. I'm pretty sure I could though, and wouldn't charge if I couldn't.

2) There are other possible non-GPS solutions simply using a marker-based approach, whereby a "marker" could simply be anything coming through in camera passthrough images that is known upfront and can be loaded into the application during **development** time using some photos. Thus we could place 3D content based on expected visuals in the drive-through experience. There are other more involved solutions on this front as well. I don't know how well the marker approach works while driving. It really just depends on the speed, the changes in lighting throughout different weather conditions, the items chosen as markers, and some other factors. In heavy rain, it just wouldn't really work anyways, whether GPS or marker-based object placement (placement of 3D generated characters and objects in the scene). A place where we'd just have to

test it out and give it a try. I'm always up for trying it out. Take a picture of that tree over there and let's see if we can use it as a marker. All sorts of possible markers out there.

A note regarding non-installed games/experiences

Note that when they load the app onto a Quest 3 that they own, it could be done via a website, in order to offer the safest experience for the user. In other words, we take the concern away that our application is going to be installing something malicious onto their headset since the browser itself is taking care of the security concerns by running everything in a sandboxed environment. Overall, I'd recommend going with ensuring consumer safety if this is intended to be run on their devices. If you own the devices, go with an installed application to get the best performance.

Tell me about the overall project process:

You'll get to pick a game type (computer/VR/XR - XR being mixed reality), a theme/plot, a rating, a location, a style, overall tempo, audio styles, whether it needs to be multiplayer, and if there's an "end result" (such as whether or not they win a prize/discount). I'll also want to discuss your motivations, of course. Why are you creating this game in the first place? That will play heavily into how I develop the game and/or experience.

After that, you'll just need to let me loose, to go do it - no further specifications. No project management. This is the way we get it done cheap. For budget level pricing, you have to give up some control. That's my niche. Nobody else would possibly do it for this price. Let me work something for you - managed only by myself, and you'll get it cheap.

Note that when I say "computer" above as a game type, I just mean anything that doesn't require a VR headset, such as desktop, laptop, tablet, or phone. I realize the headset is a computer btw, but you know… Note that a non-VR game is a valid choice though most of the rest of this document will be about VR headset development and actually mostly XR development, and when I say XR, I mean "Mixed Reality". I just need to reinforce these terms a few times before we move on (sorry).

Wow Jeff, how nice of you to let us select something in our own project?

Trying getting it done for this price the other way where you specify every little detail - impossible. My niche is providing solutions to those who are turning a

profit, but not necessarily doing great. They are seeking more customers. And to do projects for this cheap (and it is cheap) requires constraints. But, I'm out here offering up VR headset solutions to small businesses whereas very few others are doing so.

Tell me more of the theme:

A theme might be something like "save a bird with netting stuck in its beak on the other side of the canal by climbing boats that are passing through the canal". Note that Mixed Reality games mix reality with non-reality (rendered content), so, the players could be very active in the game physically. For instance, they might have to walk or run through a field and go out and find things or conquer various challenges. In the example above, the player might have to walk along the side of the canal (or fake canal) in order to see the game character which might be obscured from view by a real-world object. The theme/plot is only limited to your imagination. The length of the theme might be a sentence, or a long paragraph, but not a full page, and certainly not a 27-page document detailing every little thing. I do understand why some do it that way, but that's for bigger projects with $100k plus budgets. It's especially necessary for applications involving business processes, but entertainment is a different thing. You might want to use video in your game considering that it's so easy to generate pretty decent ones now using AI, and soon to get much better. There are many existing AI tools that make the process of videos where a character is speaking to you very easy. These would be especially good for murder mystery games, for instance, but so many other types of games as well.

You're going to manage the project you're going to work on???

That's the idea actually. You can turn on a dime. You don't have to wait for decisions to be made. Fully managing the project is just best for entertainment projects where things aren't absolute. I'm sure some are saying "this guy is gonna have a real hard time finding someone who doesn't want to run their own project". I can see it… But there are those out there who are just like, "if he wants to run it, let him, it's just easier". And you see, it is - for me, and for you, both. And you get a solution that is just way cheaper than where most would come in for complete game/3d experience solutions. And for me to be my own critic, well, that's actually best. Do you bother an artist while they are painting, and say, I'd like to paint the portion in the top right, or, I think you took the wrong approach? All the game stuff is just an interactive art of some type. You have to let the artist

who is working on it run with it. That's one of the many roles you have to accept as an indie developer, and I do accept that role. The artist has the vision, and you need to let them run with it. You don't need to worry that it will be something great by the end unless they aren't familiar with the field yet. Anyone who is will get some good work done if left alone. It just takes trusting in them and you get back way more than the other way, where it's a big managed project with a manager, QA, 3 devs, etc. Not only is it cheaper this way, but you typically end up with a better end product.

How do you expect us to come up with that money? We're barely in business.

I simply won't try to sway anyone in my direction if they don't feel that it would be either financially feasible at this point in time, or simply not give a good enough ROI over 3 years to be worth it. It's all going to be dependent on your current financial situation, mostly. If you don't have the money available, don't take the risk, because you never know for certain if this will benefit your business. You have to have the right overall plan as to whether or not this venture will allow you to start picking up new customers that keep coming back year after year, or increase existing customer satisfaction enough. And that certainly will be, in part, based on your overall strategy to gain or retain customers. It's not guaranteed to bring in new customers. So, if you don't have the money, don't take the chance.

On the other hand, if you do already have enough savings for spending on new customer acquisition, then gaming could really help bring in more customers. Gaming isn't the only way to increase customers though. This will be right for some, and simply not right for others. I don't really know which ones are going to get the most out of VR or gaming in general, so I just hand it out to lots of types of businesses.

Can you provide a few factors to consider?

1) The purpose of the game - is it just for fun, or is there a learning experience involved?
2) Where is it going to be played?
3) How often will it be played? (and how often will it be available to play?)
4) How to get people talking about it?
5) How do you plan on keeping people interested in it?
6) How do you plan on using this to increase your customer base (if that's the

intention)?
7) How will you use this game/experience to keep your business in their minds?
8) Will the game be “team based” where teams get together to play against each other online or onsite? They really might like playing as teams. Even just branch employees playing against each other might be of real value to some companies (team building).
9) Will it be for the customers, for the employees, or both?

Of course this is just a starter set of things to consider and each business will have their own considerations.

Will there be limitations?

There are 2 main limitations:
1) Conditional Ownership - You get the source code as long as you don’t distribute it to others, regardless if money is involved. This solution is for 1 business, including all of its branches, and that’s a fair deal. We can’t provide a solution to a business, and have every other business in the world go using that same game/experience.
2) I’m not going to be “storing users data”, sending them emails about their game score, or emailing them screen captures of their game/experience. All the little things that some might consider nice to have, but aren’t necessarily all that worthwhile will not be included. It’s simply a must to exclude all these unnecessary things like this to get a small business project done.

I don’t think people are going to want to put on a headset that others have worn.

They make disposable inserts that people can wear. If you deploy a VR headset solution at your business location, I would definitely invest in these disposable inserts.

If you want to see what’s available, do a web search for Disposable VR Mask Sanitary VR Eye Covers bulk. I found a product listing with 200 face covers for $27, which is pretty good imo. That’s less than 14 cents per face cover - definitely affordable. That’s $1 spent per every 7 guests that play. If only one in one hundred event guests converts into a regular customer, that could be a big win financially if you get plenty of guests. Even if it takes me standing out there

naked, we'll make sure you get enough guests/players. On second thought, that would likely backfire (no guests…). We'll figure something out.

Guarantees and Requested Payment Schedule

Before I accept the first payment, I would bring the solution out to show where it's at and how it's coming. If, at the end of the first month, I can't meet some technical challenge that I thought I would be able to, then we'll just scrap it - **no payment needed**. It would be my bad at that point, and I'd know by the end of the first month, so it wouldn't be a big deal for me to scrap it, and you'd be off the hook completely.

If I do meet the challenges that I need in order for this entertainment solution to be viable, and I do have something to show at the end of the first month, specifically that I was able to meet technology challenges, then the first payment is expected at that point in time. If you're new to technology and wondering why I wouldn't be able to, it's because the tech changes on a daily basis, and things are not always compatible… You can't always predict these things, but I won't charge if I run into issues. After that point, it's mostly going to be about actually developing the game which means all of the 3d content, animations, background music, sound effects, game/experience interactions, video content, etc.

The other 2 payments would continue as monthly payments - 60 days out, and 90 days out from the start of the project. Note that in order for the game/experience to be successful in drawing people in, there will need to be some amount of advertising, though not necessarily much. You might just advertise in your business, though I'd recommend other forms as well, including people out near the street with signs. You might consider having some free t-shirts for people that won a game or completed some experience so that other people in town take notice. I personally think that's well worth the cost of the t-shirts.

Tell me more of the development process.

The way I'll accomplish "getting it finished" depends on the type of game. But, let me just give you an example so that you'll have an idea. For some games, there might be a natural segmentation, such as a "scene" or a "level". When working on a project like this, you wouldn't generally predefine 10 scenes before you start on the first. You have an overall plot and you make the number of segments variable. In other words, if I'm able to develop 6 levels in those 3 months, then the game has 6 levels. If I make it to 10 levels, then the game has

10 levels. Note though that not all games have levels, but I do have ways of segmenting the development for any type of project.

These projects have no way of coming close to knowing how long something will take to develop. That's why the concept of the "agile" development process came about. Understanding agile is simple - it's like what I stated above. You get the inner workings done first. The things that are absolutely needed. Then you start building it out and add highest priority things first and you seek a "working solution" as quick as possible even if it's not ideal. Then you continue forth working on the highest priority items first, which in the example above would be completing the next scene. That way the project has a much higher chance of being completed. You don't want scrapped projects… In a typical agile project, you'd request feedback at a fairly early point in time. For my projects, I'll be doing my own feedback, because it will be fully managed. It's a loss of control, but it also makes it easier for you because you don't have to worry about it.

Every project is new, but I do work hard, and very rarely have a problem accomplishing what I sought out to do. If one solution to a challenge doesn't work, there's always other ways to accomplish the same thing, or close to it. I'm seeking to use each game I develop as a showcase, so I myself want it to be as good as possible too.

Will you be using AI / Neural Network based tooling?

Yes, I do plan on making heavy use of AI tooling for this endeavor, such as text-to-image, text-to-video, text-to-3d, text-to-material, text-to-audio, and others. This stuff is getting better and better on a daily basis. There's always something new coming out every single day. It's actually hard to keep up with as a full-time student.

I should be able to get a fair amount done in 3-months with modern AI tooling - it's night and day compared to what could have been accomplished before AI.

I'm not a 3d expert, but I am very capable - and feel 10x more so with ChatGpt, Github Copilot and other AI code assistants. I have very little concerns about meeting any challenges needed. I actually did very well in math in school having 3 semesters of calculus in college for my chemistry degree. It's not ALL fresh in my head, of course, but this is where ChatGPT and others help out BIG. You can just tell it what you need on the math front (and programming) and it will create a function for you. As far as content creation, I'd likely use AI for background objects. They don't need some exact detail. Foreground objects will be dependent on what I get back from Gen AI tools. If they're good enough, and

seem to be working out, I'll go with those. Otherwise, I'll make them, which was a big part of my training. I'm not a 3d wizard, but that doesn't mean I can't do better than the tools, and specifically for exactly how I was wanting it. If Gen AI tools are coming close enough to my vision, then I'll likely just use it. These tools are improving every month.

I know I can create XR solutions that would be very fun and engaging as long as I have space to work. We can easily generate good quality AI video using RunwayML Gen 3, or Luma Labs "Dream Machine", or even Kling AI, a new Chinese AI video generator, if video is part of your solution. There are quite a few more text-to-video generators to choose from as well. The text-to-3d, for easily creating 3d objects, is getting better very quickly and it's status is "decent" at the moment for most things, especially anything not right up in the foreground. That means 3d objects and scenes can be developed much much faster than in prior years, though you won't always get exactly what you want, but that is okay for background objects. Additionally, "scene building" LLMs are on the way. Nvidia has already demoed some stuff that is just beyond... Just tell it what you want in your scene, and the style, and either provide the application some assets or have them generate them for you, and it will build out the scene for you, and alter it as much as your little heart desires. I know for certain that my job is on the cutting block, or at least making these VR headset experiences will be done MUCH differently in the future. You still need a director though - someone to actually put everything together. Even right now, you can get WAY more for your money than just 1 year ago for a 3D application because of the AI tooling, and adding video into your game is also extremely easy with the AI video generators. Just a little video editing on top of an AI generated video can make for excellent game assets.

What costs would be needed for this solution to be viable?

1) <u>VR headset costs</u> (if going with VR/XR)

Right now, I'm only targeting the Quest 3 because it's affordable. You'd need to buy a headset for each of the people playing simultaneously regardless of whether the game is multiplayer or not.

The 2 versions of the Quest we would use are: The Meta Quest 3 and the Meta Quest Pro.

It depends on the particular game needs as to which one you would need.

The Meta Quest 3S is: $299 - more info in next section

The Meta Quest 3 is: $499 per headset

The Meta Quest Pro is: $999 per headset

I do want to note that the technology is young, and it will get much better over time, but it's also good enough to create some really fun interactive games and experiences right now. The solution that I would create will work for **future Quest headsets** and possibly other brands of VR headsets as well. Many will keep using their solutions for years, and it will simply get better automatically when the next headset comes out.

You might only need a solution where 6 people can play at once, and if we go with the Quest 3, then it's only $3,000 total cost for all the VR headsets. You might have a game/experience that's only 3 minutes long that let's people play one or two at a time, so headset costs are just dependent on how many simultaneous users. For a bigger venue, you may want up to 40 headsets, thus headset costs go up. At this point in the life cycle of VR headsets, you may even develop now, but hold off on deployment until the next gen of headsets is out, then launch your game/experience with the very latest headsets. You'll be one of the first on the next generation which should be quite decent because the Quest 3 is pretty good already (save the passthrough quality). Today is definitely the early years of VR still, but it really is coming along, and becoming very usable for entertainment venues.

2) Generative AI costs - I'll be using a lot of AI in this which will SAVE tons of money and allow much bigger more involved games than without using Generative AI. I'll be using it for as much as possible, but I will use the least expensive ones, and we will put a cap on the amount that can be spent here - you decide. Perhaps up to **$2,000** might be very well worth the money.

3) Multiplayer server costs - This depends on the type of solution chosen. If it's a game meant to be played only on site, there won't be any costs here.

On the other hand, if this is meant to be played by users back at home, this should be in the range of **$95/month for up to 500 concurrent users**. If you feel you want to allow more users than that, we would need to do further research.

Will this price include "**voice**"? The answer is definitely No. For online multiplayer games that need voice, we could use Photon Voice, and we could likely get away with the high-end "Plus" plan, which is $150/month. We could get away with this plan by using "voice channels" that are created programmatically when players are near the others. Note that the voice plan is in addition to the multiplayer fee, so $250/month for total monthly networking costs if voice is

needed. This isn't a fee that I'm charging you. It's the fee that you would pay a third party company for their multiplayer networking services.

For some on-site games, the players are often close enough to just hear each other (in the real world), but not necessarily.

Monthly fees aren't necessarily required for multiplayer games. If it's a game that's meant to be played on-site, I could write an aspect of the game that records the voice and broadcasts it to others on their team, or to everyone depending on needs. This would be free - just local networking and some "server" software that handles some various logic needs. In other words, there wouldn't be any monthly fees. They'd just press a button while recording, and when they let up that voice recording would get conditionally sent to the other players, based on being a team member or other condition. Just remember that every aspect of the game adds up, so we only want to add digital voice communication IF it's a big thing for that particular game, and it simply might not be. There are lots of things we CAN do, but we CAN'T do all of them…

4) Unity licensing costs (Optional)

If the Unity game engine is used, then there is a licensing fee of $2,200 per year of active game development. They recently canceled an additional runtime license for those deploying to the masses. So, that's a consideration if you want to get an online game to the masses. But getting a game to the masses requires support, so I wouldn't recommend it. Their license is really a developer license, and is needed for each developer during active development. If you don't have active development, there is no fee (to the best of my understanding). As soon as active development continues, then each developer needs a license. This has gone up substantially from where it used to be a while back, but I can understand this high price to some extent because there's a LOT more functionality these days. You do get a lot of value from using Unity, especially in Mixed Reality games. That said, using Unity is not a requirement.

There could be some other small costs like tshirts, headset sanitation inserts, etc. The four above are the big ones though.

Tell me about the Meta Quest 3S - the budget friendly version:

This might be a good option for some projects. You can certainly buy more of them for the same amount of money. There are several differences so don't just think it's a storage size difference. However, both headsets share the same powerful Snapdragon XR2 Gen 2 processor, so performance in terms of graphics and responsiveness remains comparable.

Here are the differences that I've been able to gather so far:

1. **Display Quality**: The Quest 3 has a higher resolution (2,064 x 2,208 per eye) compared to the Quest 3S, which uses the same 1,832 x 1,920 resolution found in the older Quest 2. This results in a slightly less sharp visual experience on the Quest 3S.

2. **Lenses**: While the Quest 3 uses advanced pancake lenses, which offer a slimmer design and better visual clarity, the Quest 3S uses bulkier Fresnel lenses, which are older technology. Fresnel lenses tend to create a "sweet spot" effect, reducing clarity at the edges of the field of view.

3. **Field of View**: The Quest 3S has a narrower field of view (96º horizontal) compared to the Quest 3's 110º horizontal field of view. This reduces the sense of immersion slightly.

4. **IPD Adjustment**: The Quest 3 offers a continuous IPD (interpupillary distance) adjustment, allowing for more precise tuning of the lenses for different users, while the Quest 3S returns to a more limited stepped IPD system with only three preset distances.

5. **Storage Options**: The Quest 3 is available in 512GB, while the Quest 3S only comes in 128GB or 256GB versions, meaning you'll have less storage for games and apps on the 3S.

6. **Body Design and Weight**: Although the Quest 3S is slightly thicker than the Quest 3, it weighs almost the same. However, due to its use of Fresnel lenses, the weight is distributed more toward the front, which may cause some discomfort during extended use.

What if they want to bring their own VR headset? (yes, but…)

It's going to depend on the game requirements, but this is possible. We can now load VR/AR/XR solutions from a website. For instance, when you visit a website in a 3d headset, you can launch a VR/AR/XR headset from clicking on a link on the webpage. Not just for VR but for AR/XR as well.

It's estimated that approximately 25 percent of US households have a VR headset, but I don't know how many are the Meta Quest 3 or the Meta Quest Pro. For now, I'd mostly say that it's likely best to plan on buying VR headsets.

Regarding installed VR/XR applications versus browser based VR/XR applications, there are some differences in the technology. For instance, if you launch from a website versus having the application directly installed on the device (side-loaded onto it - ie, not via app store), there will be a hit on performance. Not all games need high performance though, so it's just a consideration to make based on the particular type of game desired.

Installed applications will yield the best performance which might be beneficial, but not necessarily. It depends on the game. For XR games, where you have the camera passthrough taking up more processing power, it's likely best to go with an installed application.

While you could "side-load" the application on their headset if they bring their headset to your business, I'd recommend that this is not the solution chosen. It would take them time to install it, and you might not want any liabilities of them complaining about what they think the game did to their device, or installed on it.

I'm not ruling out being able to install applications on their device, but I'm also not recommending it overall, with a final statement that in certain situations it might be a fit. So, I'd just suggest buying all the headsets you'll plan on needing to use in most cases, and perhaps limiting that to 20 or less at a time to keep your headset expenses low (enough).

What if we want our solution at more than one location?

This would need to be looked at on a case by case basis.

I would generally say it's likely to be about $6k more per site, but it just really depends on the game, the location layout, etc.

I would suggest that as a price for a game that needs to have things positioned differently in a different space for each additional location. This of course requires not only work, but also lots of testing at the new location as well as determining which interactions need to be changed and how difficult it might be. It could be somewhat easy or very hard depending on the game/experience. It's not just

repositioning items in the scene, but rather that it's going to affect all the interactions and animations. A lot of changes in some games. I might just have to say no in certain situations, but typically it would be a yes, and for a reasonable cost. This would basically be done after the 3 month project of the initial location where it's deployed, and would likely be about a month per location to finish development, but only really willing to do if just 2 or 3 locations. Otherwise, for those with more locations, their employees would need to learn how to rework it for the other locations, or hire on someone which might not be a bad idea for a business that has enough locations.

Are these solutions better for "year round" entertainment, or an event?

Either. I want you to use your imagination as to how it could benefit your business. I mostly envision entertainment venues where it gets people talking about and remembering your business, or even enhances an existing entertainment venue, such as mini-golf. It could be used daily, weekly, monthly - however you envision it.

Are you going to bring us any possible ideas?

I'm certainly willing to, but not unless you ask. I want you to dig up something that is most relevant for your business and location. Look around. See what can be done outside in the vicinity around your business, or inside your business. Just get me to a theme. An overall plot. Spend some time thinking it over. Some random thoughts that are generic might be something like a rainforest experience, a haunted mansion, a "who done it" game, but there are so many possibilities. Just trying to get your mind rolling on ideas.

If you want, we can set up a day to brainstorm ideas. It will be considered one of the days of work towards the project, but could prove very beneficial.

What technologies will you use, and will we need to license anything?

First off, it depends on the type of game

If it's a standard computer game, I'll be using the Bevy game engine written in the Rust programming language.

On the other hand, if it's a VR/XR game, my options right now are mostly: Unity, Godot, Aframe/AR.js, Babylon.js, or some early level Rust (Bevy) stuff. Bevy is very good, but only has limited VR/XR support through some third-party open source projects that are pretty bare bones at this point. There are other options however. This isn't a complete set by any means.

If we side load (install) a VR/XR application, then I would use Unity, Godot or Rust since they can build installable applications that have performance benefits over WebXR (browser based) games/experiences.

But I'm not limited to this set of solutions either. I can also use an Android library called ARCore (programming in the Kotlin language), which is Google's platform for building Android based AR/XR experiences. It's basically using the core technology without any game engine functionality. Depending on the game/experience, it might be enough. Having a game engine is a real nice to have for making a game though. Note that the Quest uses Android as its underlying operating system.

There are additional solutions out there.. It just depends on the project's needs as to which solution I think will be the best fit for that project taking into account all considerations. Using any of these frameworks is within my existing skill set. I've studied up on all of them, and could easily use any one of them. Each has pros and cons, and I'll decide based on the project. I would take a basic understanding of the game requirements, then go make that determination guided in part by my existing knowledge, but with further research.

The only licensing that would be needed would be if I decide to use Unity which would just be a developer license while it's used by the business, and that's only approximately $1,800/year/developer during active game usage. It was cheaper a while back, but I guess they aren't profitable, so prices came up quite a bit. Still not profitable, but I guess that's their short… (they make the money through the companies making the games, not the engines). This is the licensing model as I currently understand it, and holds for any revenue amount that a business makes, however, I will conduct full research on anything that requires licensing. It's almost best to avoid licensing if possible, but that takes away VR/XR at current for the good solutions like Unity or Unreal Engine. I don't do Unreal Engine, by the way. I just didn't like having to use either C++ or their "blueprints" visual scripting language. Neither was a fit for me, so I've ignored Unreal Engine. That means that, if you use it next year, this $1,800 charge would be applicable again, and so on for subsequent years. For some, that's not a big deal, and for others, that might make you want to choose a web solution at the moment, or a non-involved VR solution, which we might be able to get away using Godot for. There are a few other VR creation development tools out there, and will be in consideration as well based on requirements.

I will be choosing the technology based on everything I envision, and I am actually not allowing clients to make choices regarding the chosen technology.

There are lots of things to consider. Choice of technology will simply depend on the needs of the project and what advantages that technology brings forth over the others for your particular project. That said, you can let me know if the technologies chosen need to be free and open source, like Bevy and Godot. I don't rule out using Godot. It's just not as far as Unity with VR/XR integration. It's pretty bare bones, but might work for some projects. It's completely free.

Will we own the solution afterwards - the code and everything?

As mentioned above, it's "Conditional Ownership". You get to keep the code and own it for yourself, but it will be in the contract that you cannot provide the code to others in any way, and not even after making changes to it to make additions, whether changes are made by myself or someone else.

What can be done inside the business?

This will typically be best as an XR game. Some examples might be 1.) some virtual objects hanging off the ceiling that you interact with in some way - perhaps throwing darts at them, or 2) have some videos with clues show up on the various walls inside a restaurant as part of some murder mystery game. In this way, you can still see and interact with everyone around you. You just get additional things in your view to interact with as well, but they don't block your view of the others. Rather, the others block your view of the game (if you let them).

Another example would be a trivia game that included some video content as part of the trivia. Everyone would be playing together. It would just be enhanced beyond typical trivia by including video content/clues. Perhaps the content is generated with AI, and represents something in history, and you have to answer questions on.

These are just some stupid examples to help you get started thinking if you happen to be interested in this at all. There are so many possible games.

What about future changes?

You'll be able to make subsequent changes as well, of course, via anyone you choose. For me to work on it in the future past someone else's work would require all their changes to the game/experience to have been tracked in the same source code repository. This simply means continuing to update the same repository and not create a new one with the current source into a different repository for whatever chosen reason. If you do, that's okay, as long as a full set

of changes is kept, and it's not "kinda" kept. It's there. If the history is there after someone else works on it, then I'll still work on it going forward. I'll have YOU create a Github account for the business and I'll add the code to a private repository once you grant me access to the account. You can have a free account for small code bases like this, so no worries on having to pay for storage of the project. I'll document the account during the documentation that I write for the project so that the location of the source code is documented.

Can we have one of our employees work on the solution with you?

Unfortunately no. Training eats into necessary work time like you wouldn't believe. They will be able to read the documentation at the end and understand how the project works, and will be able to dig into the project and change it over time without issue IF they have already done their technology training. It's a very long train in this particular field, so most likely best for another computer professional to take over some time down the road rather than having someone that isn't trained on this technology try to work on it at some point.

That said, you can certainly have your employees take over the project at the end. I will provide complete documentation and will spend a day or two with them discussing the project. This is the best way for your devs to learn the technology anyways if they aren't yet familiar with it. They'll learn it 10x as fast at the end of the project when they can see the complete solution.

I know this guy in computers. He's looking for work. Can he work on it with you?

Again, unfortunately the answer is no. Not for this price. Put me on a project with 6 other developers that has an actual real world budget along with it - that's great. Small business solutions are different. You have to be in the software development field to know where this answer comes from. 95+ % of people will hear this answer is that he doesn't like being part of a team. But that's just not the truth. You have to have worked on a software development team to understand. There are literally at least 10 reasons why I wouldn't do this, and it's not because I don't like working with others. I'll just state a few without going further to give you an idea: 1) I don't really know them, and others don't know how they'll work with me - regardless how someone else feels that they would, 2) we all know different technologies, and we barely understand each other's technologies, so going down that road means someone is learning new technology during a project, which never works, 3) so many decisions and everyone has their own

ideas, which yields a lot of combat and you don't end up with a unified solution - no single person's view, but rather a smorgasbord of ideas, 4) most partners will do about 25% of the project (not very well), and expect 50% of the pay, no matter how much their friends talk them up… This is a short list of 4 reasons. I could go on, but in short, it's the fastest way to ensure the project will be toast - put 2 devs on it.

Even just having a person who interns is a net negative, by far. All that time you're spending on them is time that you are not developing - precious time. If you have 2 devs - get them going on 2 separate projects. That's the way they prefer it anyways. They want to be able to use the skills they know. Each knows how to do it in their own ways. Then, later in the project, if and when you see a project working out, then you can build out a team and take it to the next level. At that point, there won't be much high-level decision making needed in the project. The project has already taken a direction, and has proved successful. Developer frustrations will be minimal at that phase in the project. Most successful software projects have started off with a single individual creating the vast majority of the inner workings of the project, and that's because they enjoy the challenge and were able to **get away** where they could think clearly. The corporate environment of bullying you from the top down doesn't ever allow anyone to get into that headspace where they can actually think and work.

We use "xyz business" for all our computer work. Can we let them manage the project? I already have a working relationship with them.

Definitely not - for most of the reasons mentioned above. They would get in the way and likely wouldn't help in any way. And they would want to micro-manage the project which is precisely what I'm not allowing for this price. Lastly, they are unlikely to understand the technology at all.

Tell us more about your background.

I'll send out a resume on request, but the overall gist is that I worked for many years as both a desktop application developer and web application developer with a focus on applications that had lots of interactions with databases. Some of the types of things I did were creating database schemas, creating database stored procedures, writing import/export and transformation scripts (typically sql), creating user interfaces for applications (desktop and web), creating back-end processing for applications. These are just some of the types of things. My last job was actually nearly 6 years ago now (not quite yet). This was working on a

client project for Ciber/HTE (HTE bought out Ciber). The client project was ThyssenKrupp Aerospace. I had actually worked on that team during a prior contract, but decided to go back and do training. Unfortunately, the project at TK Aerospace was dwindling down because the client was converting to an SAP system (and the consulting company I worked for (Ciber/HTE) wasn't part of the SAP development). So our team was getting cut little-by-little, month-by-month. After I got let go from the project, I was still part of Ciber/HTE, but also saw all the technology running faster than I was able to keep up with and still work. To give you an idea of the technology field, I spend over a year training in that gap period to come back and work on a new project for much less time than I spent training… You can go net negative in this field really easily - they don't get it… I brought in all my new skills but didn't get paid for going out and training.

Today, I'm still training. Almost unimaginable, but that's the way when you choose a field like 3D, and then you add in all this VR/AR/XR and all the AI/NN on top of that. Plus all the programming languages changed and there's a whole set of new databases that are quite different from the old ones. **That's how crazy it is right now**. Hug your computer friend because they are getting slammed with new technology, and because they're part of that field that's bringing forth this next generation of craziness - assuming it goes well.

Do you really expect us to believe you have your hands in all these fields?

I'm not claiming to be an expert in ANY of them. Just that I know what I need to do for most projects and when there's a project that has an aspect that I need to touch up on or learn something new, it might be a day or two of learning, but that's 10x easier than finding someone when needed to do a day or two worth of work.

Jeff, why don't you let us include this audio engineer we know in the project?

I'm sorry to say, but that won't work for me, regardless of whether or not you were planning on paying them outside of our contract. The game / experience would no longer be a creation. It's just another project completed. I wouldn't be happy with it, and not because the audio engineer wasn't good. They might be an expert at their craft, but when you have multiple people contributing to artwork, there is no unified sense to it. No vision. Nothing to grab you. When you let a single person just roam with their ideas and vision and bring it to fruition, it can be a much better experience even if there is less overall depth/quality in the

audio produced. It's still going to have the vibe, and the overall essence that the artist had in mind. It's going to sound the way the artist had in mind. In a follow up project, you could always have an audio engineer make improvements to the sound. I'm sure they can, but I need it to have the feel I'm after, so this main phase of the development is just going to be myself on all parts regardless if I have weaknesses. And in doing the audio myself, I can record it from the synth into FL Studio, cut what I need to, and try it out in the game - all within a few minutes - I don't have to schedule anyone, or get them to understand my vision. I also don't have to step on someone's feet when I think it needs to be redone, which is why changes that should be made often are not made on team projects.

Money is actually the secondary concern here. Being able to deliver on my vision will be the primary concern, and I can do that on my own, albeit with an expectation of intermediate-level quality. That said, on the money front, doing these small business projects wouldn't make it worth it for me once I start splitting the profit. The typical developer (any type) will want a whole lot of compensation for doing a tiny bit of the project. Well, there goes my profit??? That scenario is just not happening for me.

Well what are your weak areas?

Right now, math and sound/audio are my biggest weak areas for game development. That's not saying that I don't have good math skills. I've definitely refreshed quite a bit while learning 3D. Everything in the 3D field involves math. That said, I'm not currently a math expert, and I also don't need to be in order to create 3D games and experiences. I always did well in math in school - it's just the type of thing that doesn't hang around, so you need to refresh in the areas you plan on using for a project, if and when needed. You can't keep your head around all of math, and I certainly do not plan on it.... You just need to know what you need to look up. You need to be at that point first. If you aren't sure what to research, then you're halted. The initial training in 3D solidifies some core concepts that allow you to always be able to envision the solution - even if you need to go figure out the exact math needed during the project.

Regarding music, I've learned the application named "FLStudio" which is a digital audio workstation (DAW) used for creating songs, background music, and sound effects. I also bought and learned to use a Moog Grandmother synthesizer for sound effects. I can also do sound effects in FL Studio via various synthesizer plugins, however, the natural sound of the analog sounds of the physical synthesizer recorded into the DAW can be more pleasing. I've also learned the

basics of music theory and understand simple concepts like what a chord is, what a scale and key are, the basics of harmonious sounds and understanding things like harmonics, rhythm, what tempo to choose for what style of song. Just the basics - I'm not trying to be a musician, nor do I need to be for making these games/experiences. I know how to create chord progressions and melodies in the piano roll and drum loops in the channel rack FL studio. I know how to arrange songs on the main playlist/arrangement screen from all parts and pieces. I'm no music wizard and not trying to sell myself as one, but I will certainly be able to come up with great background music and sound effects that fit the game - and will be tailor made for it actually. I don't play any physical musical instruments at this point. I won't deny that actual instruments can have benefits, but that's okay - you likely won't notice that much in the games. There's lots of sound effects coming over the background music, and thus doesn't need to necessarily be "pleasing guitar music" like you'd want in a song. It's often more a repetitive drum loop or some spooky piano music. You don't need harmonious guitar music for games. The DAW is mostly all that's needed, with analog synth being useful as well, and then actual instruments being a bonus at times.

What are you doing right now - just hanging out?

That's what everyone likes to think isn't it. I'm back learning an aspect of Blender called "Geometry Nodes", which is a procedural workflow in blender. It's just another way of doing things that is much more flexible. It's not needed at all to do my job though. Then I plan on some additional math training - taking it further. Then I'll likely be back studying more audio/music production for a little while. It's like this - being an indie dev and furthering your skills. You have to jump around and keep pushing each piece forward, but you never have to be an expert at all of them. I never plan on attempting to do so. Every project needs different skills, so you just work within the domain of the things you currently know. I feel 100% capable at this point on almost any project with my current skills, but it never hurts to learn more and other ways of doing things faster, for instance. Geometry Nodes is something I already know, but not like the pros. So I'm currently working on breaking out of beginner-level Geometry Nodes into intermediate territory, which can take time, but the important point is that it isn't something that I need to know AT ALL in order to be successful on 3D projects, so please try not to read this section as "he doesn't know how to model 3d objects yet". Geometry Nodes is just a new thing. It's actually still being developed in a lot of ways, but already has a lot of functionality. It's just helpful

and expands possibilities, and adds flexibility into modeling via tweaking the model via parameters. This course I'm working through will be the first of my next journey into furthering my Blender skills with Geometry Nodes. It's a big train because procedural model generation can be quite involved, depending on what you are creating. The great thing though is that there are many "generators" that come about from procedural workflows that you can just buy and use. But if you know the skills of how everything works under the hood, then you can change a Geometry Nodes based generator to fit your needs, or create a new one if you need to (often times you wouldn't need to create new Geometry Nodes setups in a project because it's used more for creating object generators, and for looping simulations). It can improve the overall speed of development, but that's for someone that needs to make 3D objects and scenes all day long. For an indie project, modeling is just another part of the project, so efficiency of modeling isn't necessarily any major concern, especially when so many 3D objects can be purchased or generated, which is likely to be done for many objects, and is typically great for background objects. For foreground objects, you typically want to model these yourself.

Where would you work?

I'd work out of my apartment 95+% of the time. The only time I would come out is to test the game out there on site if you are choosing to do a mixed reality game. Otherwise I would work 100% from my apartment and just meet up with you for a few meetings to test it out with you in person even though it's not outdoors.

Hey Jeff, we don't really know you… How's that background looking?

Well, I can really only speak to a criminal background check, and not necessarily to what others are going to say (you can't control that). The criminal background check should go quite well with my 30-year clean record. I only had 2 murders before that!, and one very small cannabis arrest in college - not that it should have ever been illegal… Can you handle it? Yes, the murders were a joke (to me anyways, I wasn't convicted!).

Jeff, we don't like any of your ideas so far…

These ideas might sound exceptionally stupid to you, so you tell me what you think would be entertaining in some way. That's what I really want. You can come up with a concept that is unique, then let me run with it. It really does take time to

come up with fun ideas, and I'm certainly willing to help out to brainstorm. As someone who sees the technology, I can probably help guide that effort as I know what can be done.

Jeff, don't you see bigger things happening out there?

Yes, of course, I see all the craziness with the robots/automation and job loss. We could hit socialism tomorrow, and this effort is likely pointless, but as of right now, nobody has said anything to me, so I'm continuing forth…

Let's hear a conclusion - wrap it up, WRAP IT UP!

It's great the AI machines will build it all for us soon.. Some day, not too far out, building a game or entertainment solution will just be a matter of walking around with the VR headset on and saying, put this here, put this over there, when the ball hits this thing, perform this action, etc. You'll just build the games by directing it. It's probably not too far away with the pace things are going. But, as of yet, that's not the case, and there's no telling exactly how far out that day is. Let's build something fun together and increase your customer base. Lastly, I don't plan on doing follow-ups. I'm giving everyone their space to think whether or not an entertainment solution is right for their business. I know already that most will not like some aspects of my process, and they indeed won't get the skills that I just spent nearly 6 years training on. I'll only let those who know how to behave as human beings, and how to respect that amount of training that I just did get my skills. Sadly, the typical process is someone grabs some technical person and starts throwing a bunch of technical questions at you in an interview. If you haven't looked at a certain technology in a few years, you won't answer interview questions well (at all) - you have to get your head back around it. They'll get the typical developer. They don't know what it takes to do it all.

Jeff Gillin
Independent XR Developer
407-451-4682
jeff.gillin@gmail.com