Why I Create Games

~ 1 ~

Place one element down. Then another. The Inspector overflows with scripts, lighting, images, and sound — glued together by the engine. The tilemaps brim with squares upon squares of handcrafted objects. Some of the elements contain physics, some contain code, and some contain colliders. But, all is unmoving, until, with a "*click!*", I press play.

My creative process, stemming from an intersection of art, programming, music, and more, developed from some seemingly insignificant choices I've made throughout my life, creating a domino effect that built the passion I have today. And it all started with one tiny domino, all the way back in my timid years of first grade.

~ 2 ~

It all began with my first exposure to coding, when my parents signed me up for Khan Academy so I could grasp the concepts as a 6 year old. This online class taught me the basics of a JavaScript-like program that gave me the ability to create a scene using shapes and animation.

After a few weeks, I completed my first ever project, which made me incredibly proud. My task was to build a snowman using JavaScript. My snowman had a little orange carrot nose, a black top hat, and was surrounded by animated falling snow. I felt so accomplished that I made something on my own that I shared my creation with my parents. Even though my snowman was pretty rudimentary, my parents supported my creativity and encouraged me to keep going.

~ 3 ~

From 6th to 8th grades, I attended a Montessori school called Bowman, where I volunteered to take a basic computer science class. This was my first ever in-person coding class, and I was excited to have someone to learn from. My coding teacher noticed my enthusiasm for computer science *immediately*. Very early on, he tasked me with miniature solo projects, such as creating *Snake* from scratch using HTML and JavaScript — this is a game that requires the player to steer a snake toward food without running into itself, but the snake grows each time it encounters food. Combining something entertaining, like a game, with computer science introduced me to game development.

Near the end of 8th grade, I finished a game called *Moon Clicker,* my biggest project in middle school that showed my "creative" side. The game consisted of thousands of lines of code, the most out of any project I've created at the time. Like

other clicking games, the player's goal is to click an object, in this case a moon, to make money. With this money, the player can buy upgrades, letting them gain even more money faster. Initially, I only shared this game with two of my friends, and within weeks, the project link spread through the class. During lectures, I'd even sometimes see some of the students secretly playing my game on their computers, pecking away at their keyboard.

~ 4 ~

The transition from 8th grade at Bowman to 9th grade at MVHS was challenging, and my confidence evaporated once I left my middle school keyboard and stepped into the massive high school campus. This was my first experience in public school, and I was overwhelmed; there were way more people and more classes than I could count. I had a hard time choosing my classes that year, although one stood out that I knew I wanted to take.

That year, I made the vital decision of picking an Intro to Computer Science class as my elective, which quickly became my favorite class. Truthfully, with my background in coding, I was probably a bit too advanced for this class, but that gave me the opportunity to go above and beyond for each assignment. Plus, it gave me the foundation I needed for AP Computer Science, which I took as a sophomore. While AP Computer Science focused on the fundamentals of Java, it leaned heavily on projects. Such projects often focused on developing video games, including the process of maintaining a workflow to make art and code on time. I even won Best Gameplay in the class for one of these projects! As a person who likes both being creative and writing code, I realized that a game development major would be perfect for me: video game development requires an understanding of art, music, and programming, while still taking into account my value of enjoyment. When working on a game, if I got bored of coding, I could just switch to making art. If I got burned out on art, I could switch to music. Even if I was having a terrible day, there was always something fun to create.

~ 5 ~

Despite my academic success in computer science classes as a freshman and sophomore I began forming a solitary bubble around myself. But once Advanced Computer Science class rolled around my junior year, which featured a year long project, that bubble burst open.

I’ve never been a fan of working in groups. For every single project that I've done, I always did it solo because, well, I thought it was faster, you know? I was faced with the uncomfortable truth that from then on, more and more projects would require collaboration, something that I was not used to at all. I like solving my own problems! Multiple people programming on one project can cause developmental inconsistencies.

I prefer to be in control of all decisions that go into my code, and I don't want to defend my viewpoint from others. It's more efficient and painless.

The task was relatively simple; we needed to create a game that would reinforce the understanding of students, specifically younger kids in low-income families, about how different vehicles affect pollution. We weren't exactly teaching them per se, but we were instead trying to make what they learned in their lessons a little more enjoyable.

Unfortunately, my partner and I quickly realized we didn't know how to collaborate in the slightest. Besides merging conflicts, which is where one person's code can overwrite the other, we needed to learn how to plan properly. Perhaps in a more peaceful, solo environment, this project's developmental process would have been easier to manage, and I could simply just... do the project; as it stood that day, however, we were forced to organize and make a plan.

From this moment on, I realized the value of collaboration. While it may be a bit more confusing, it is *way* more efficient once you get your act together. Of course, it wasn't easy to get to that point — I never did any significant ounce of collaboration until then, so it took trial and error. Nevertheless, we finished the project quite quickly after that realization, getting an A on the assignment. To this day, I am still sending out updates to improve the game as more of a passion, so hopefully, at maybe one point or another, I could officially publish the game for anyone to play in a complete state.

Learning to merge my thoughts with someone taught me one of the main values of game development — it has a soul. Unlike regular software development, creating games is an art form, requiring the developer to value their creative input and imagination in order to create something personal and human.

Unfortunately, it's not all that bright, and I have had fears about what the future could hold for this passion I have fostered throughout my entire life. More and more often, companies are replacing their developers with AI. And honestly, it made me wonder sometimes — what's the point of my dream if there are no jobs left for me? What even is the point of culture, if it's not from humans? What even is the point of collaboration, if AI does everything for me? While AI doesn't feel the frustration of a merge conflict, it also can't grasp the emotional delight of finishing that project on time, or working with friends, or watching others play a game that you know you've brought your entire soul into. What even is the point of experience, of art, of building a community, if you can't even call it yours?

Even though I have my doubts, I have come to terms with the situation. Indeed, AI is changing the job landscape, but there will always be jobs for me. I don't have to

work at a big company that doesn't care about their employees as people. I could work at indie studios, or maybe even create my own, tailored to an audience who wants authentic, human work. The experiences I've learned, the community I've developed, the collaboration I've participated in, all carry connections that can't be recreated with a machine. Personally, after years of thought, I heavily doubt that game development will die, and I'll die on that burning hill alone if that's the only human thing left in this world.