

Austin Crook

19 April 2023

Thesis Director: Dr. Jeff Matocha

2nd Reader: Dr. Jeffery Sykes

3rd Reader: Ferris Williams

The Creation of *Blacklight*

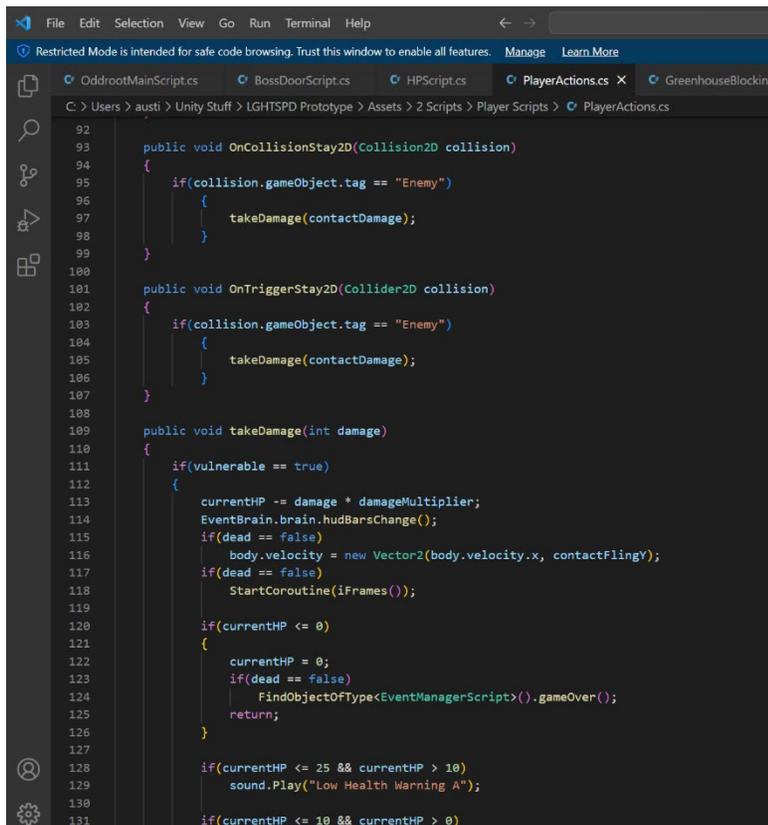
For my senior thesis, I made a video game.

First of all, why a video game? Aside from the fact that I've been playing video games for almost as long as I can remember, video games hold an attraction to me as the perfect fusion of nearly every creative discipline. Any game—even any one element of any game—requires artwork. That artwork has to be animated, that animation needs sound effects, and all of it should work together with a story that guides its design. The question of whether or not video games can be considered “art” has always been a little silly to me. They are not just art; they are *every* art. I've always enjoyed drawing, animating, writing, music, and designing board games, so why not do something that flexes all of those creative muscles?



I drew inspiration for *Blacklight* from a number of sources. The *Metroid* series and *Cave Story* were my main inspirations for the gameplay. Both are full of 2D exploration-

based platforming, shooting, and interesting and colorful enemies, and I wanted to recreate that in *Blacklight*. My inspiration didn't only come from other games; Terry Virts's *How to Astronaut*, a memoir of the author's time aboard the ISS, greatly influenced the setting of my game. The stark cold of space and metal-plated rooms of the ISS perfectly complemented the somewhat moody atmosphere of the game I wanted to create. Thus, the SISS—the Second International Space Station, the setting for *Blacklight*—was born. As for the story itself, it owes its conception to an incomplete graphic novel I drew when I was 15. While the final game is entirely different to that of the comic, both feature two protagonists, twins named Erik and Sofia, each wielding the power of the sun and the moon, respectively. Reimagining them in the context of the game was my way of honoring my junior high self's ambitions.



```

92
93     public void OnCollisionStay2D(Collision2D collision)
94     {
95         if(collision.gameObject.tag == "Enemy")
96         {
97             takeDamage(contactDamage);
98         }
99     }
100
101     public void OnTriggerStay2D(Collider2D collision)
102     {
103         if(collision.gameObject.tag == "Enemy")
104         {
105             takeDamage(contactDamage);
106         }
107     }
108
109     public void takeDamage(int damage)
110     {
111         if(vulnerable == true)
112         {
113             currentHP -= damage * damageMultiplier;
114             EventBrain.brain.hudBarsChange();
115             if(dead == false)
116                 body.velocity = new Vector2(body.velocity.x, contactFlingY);
117             if(dead == false)
118                 StartCoroutine(iframes());
119
120             if(currentHP <= 0)
121             {
122                 currentHP = 0;
123                 if(dead == false)
124                     FindObjectOfType<EventManagerScript>().gameOver();
125                 return;
126             }
127
128             if(currentHP <= 25 && currentHP > 10)
129                 sound.Play("Low Health Warning A");
130
131             if(currentHP <= 10 && currentHP > 0)

```

Coding

The first thing that became apparent when I started this project was that I needed to learn how to code. I had very little experience in programming, mostly limited to a few hours several years ago playing around with Scratch, a visual coding language designed for children. As such, I always quickly grew frustrated by the small scope of the toolset. Scratch was limited to

whatever premade blocks of logic it came packed with, which meant tight restrictions on gameplay.

For *Blacklight*, I decided to go big and use a much more robust system at the cost of a higher learning curve. I decided on Unity, a game creation engine that uses the C# coding language and is used by professionals the world over, even by big-name companies like Nintendo. Syntax, documentation, compiler errors—these were all things I had never encountered before. Still, I stuck at it, and a few hours of YouTube tutorials (and many, many missing semicolons) later, I had a somewhat functional skeleton of a platformer: a little man made of royalty-free art jumping around on a couple of obstacles. It wasn't much, but it was a start.

Every platformer is physics-based; running, jumping, gaining or losing momentum—all of this is controlled by mathematical calculations run dozens of times per second. Unity



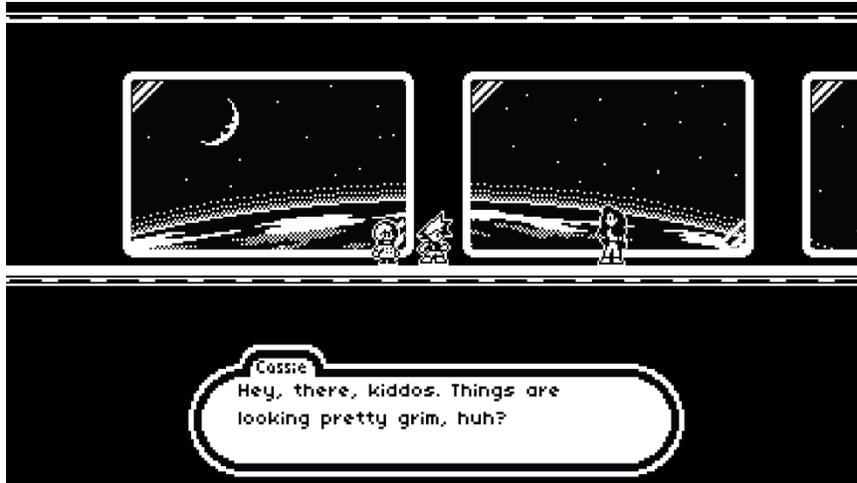
comes prepackaged with tools to streamline the physics coding for the average developer, making it much easier to assign mass, gravity scaling, and friction to in-game objects and have them behave realistically. Still, setting up the physics of the game presented a number of admittedly hilarious obstacles. For instance, one bug I had involved the player's speed. While holding the projectile-reflecting mirror, the player is slowed, and when the mirror button is released, the player returns to their original speed. Initially, a mistake in my code meant that if

the button was rapidly pressed over and over, the player could attain theoretically limitless speed and instantly blink off the screen. Needless to say, that had to be fixed.

I learned about scene loading, prefabs, instantiated objects, object pooling, workflow optimization, tileset creation, collision layers, and a whole host of other game creation principles. I had to learn how to organize files properly; even in small projects like this one, the number of files quickly grows to ridiculous proportions. Having an easily understandable and efficient method of categorizing dozens, then hundreds, of scripts, sprites, and sound files became a much more important task than I initially thought. The extra time spent organizing a hierarchy of folders was well worth it, however, as it saved potentially hours of time spent searching for the right files over the course of the three semesters I worked on the game.

Artwork

Art is a major part of any video game, not only to look pretty, but to effectively communicate the information about the game at a glance. For *Blacklight*, I settled on an art style that required minimal time and skill: a black and white “1-bit” art style. That’s not to say it was easy making art for the game—far from it. The 1-bit style comes with its own challenges, mainly clarity issues. In only two colors, with a very limited number of pixels, how does one create recognizable characters? How are characters and enemies distinguished from the background? It’s astonishingly easy to make everything an unidentifiable mess when only working with an art style like that, so extra care has to be taken to make it readable.



By the time I decided on the art style of *Blacklight*, I already had the rough designs of the two main characters in mind. To make them work in a handful of pixels, it came

down to picking key details to exaggerate for the sake of clarity. For Erik, the brother, I chose to greatly increase the size and spikiness of his hair. Sofia, the sister, was a little more difficult. Making her hair bigger didn't work for her. I didn't want to change it, so instead I focused on her clothing: I made her "sweater" (really just the top half of a torn spacesuit) bigger and bulkier until it covered most of her character sprite. It gave her a unique appearance to distinguish her from Erik. She doesn't stand out in a crowd as much as her brother, but I think that fits her personality better.

For the cast of alien enemies, the task was similar, but came with the added requirement of making it apparent at a glance what each enemy was going to do. So, the one that shoots at the player has a cannon for a mouth, the one that flies has wings, and the one that punches is almost 50% arms. The designs are simple—almost no-brainers—but the simplicity freed up time to make more complex and fluid animations, which is where their real character shines through.

Animation is a slow, time-consuming process that's easy to get wrong. I should know; I got it wrong many times myself. I estimate half of the animations that made it into the final game are second, third, or fourth tries at animating the same action. Some actions that looked fine in a vacuum wound up not working when put into context of the game; many times, the motion had

to be greatly exaggerated to account for the small size of each character relative to the size of the screen. As they say, animation is an iterative process; in my own words, for anything to be good, it must be bad at least once.

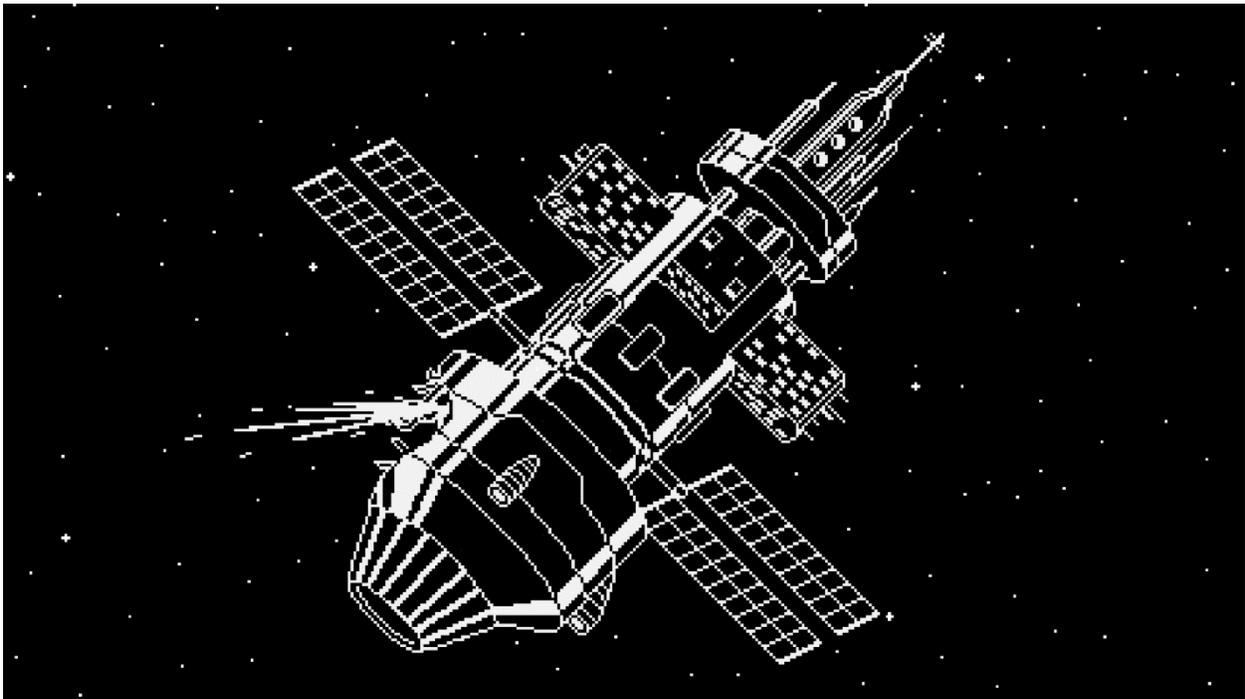
Overall, I estimate 50 to 60 percent of my time spent working on *Blacklight* was devoted to artwork and animation. While I don't necessarily regret that time, I imagine that if I were to make another game in the future, I would consider collaborating with a dedicated artist for efficiency's sake.

Game Over

There is an old adage in game design circles: fail faster. Generally, what this means is to ignore the fineries of the game, like nice artwork or music, and make a prototype as soon as possible so you can see what works and what doesn't. At the heart of it, though, is the idea that failure is inevitable. Instead of avoiding it, one should pursue it to get the mistakes out of the way. It's an oddly comforting sentiment. I tend towards perfectionism, so knowing beforehand that I was going to make mistakes—say, an infinite speed glitch or a hundred bad animations—was freeing. It helped me to not get discouraged when, for instance, a game object refuses to behave properly, only for me to find out it is my fault the whole time.

At the outset, I initially planned for *Blacklight* to have six large, interconnected areas, complete with four boss fights and a complete story with eight characters. I quickly realized I was overambitious. Most game development teams are just that: teams. Even the smallest independent studios usually have at least four members, and one man (technically two, as my good friend Graham Scarborough provided some of the music) could not possibly make a full-length game in less time than it takes some two-hundred-person teams. So, I had to cut down on

the scope of the project. The six main levels became two, the four bosses, one, the eight characters, four (counting the playable ones), and the grand, epic story was pared down to just a few cutscenes ending on a cliffhanger. This proved to be a much more manageable task and allowed for a higher degree of polish. I do intend on finishing the game and bringing it to the glory of its initial vision, but for now, *Blacklight* remains more of a proof of concept than a fully fleshed-out title. Still, I'm proud of what I've done, and I've learned more by jumping into things feet-first than I ever could have imagined.



Visit <https://royal-beetle.itch.io/blacklight-version-02> to download
and play *Blacklight* for free on any Windows PC.