Warren Tech Tycoon

- Project description:

  Warren Tech Tycoon was the first game jam project that I worked on. It was a 3D, single player game that saw you going around and building up different programs within Warren Tech to simulate the progression of the school over it's 50 years

- Team list - 10
  - Me
  - Audrey
  - Calloway
  - Dylan
  - Emerson
  - Eli
  - Keon
  - Cameron
  - Pink
  - Nash

- What worked well?

  We managed to get the game completed and more or less feature complete within the allotted game jam time. We also were able to build and submit by the deadline.

- What didn't work?

  Fundamental gameplay flaw (ends too quickly).
  Camera suffers from jerky movement when players reach the end of the screen.
  Tried to take advantage of classes and OOP principles without proper utilization of those practices.
  The post process filter that we used was pretty bad in my opinion.

- What did I learn?

  I think that this game jam was useful for having us learn the game development pipeline. It was new and a refreshing break from our main project. I learned about packaging our games and getting them playable. I learned about time utilization as I didn't make the best of my time during this jam.

Spider Boss Laser Game

Project Description:

This project was meant to be a 2D side scroller where the player is an alien creature fighting bosses in a boss-rush format. The player would be able to flip the arena around in order to avoid attack and traverse the levels.

- Team list - 7
    - Me
    - Marvin
    - Travis
    - Emerson
    - Maddie
    - Pink
    - Nash

- What worked well?

  The art produced by Emerson, Marvin, Maddie, Nash, and Pink was really nice. Travis did his best to get base functionality in and something there. I enjoyed the team dynamic on this project a lot.

- What didn't work?

  The game was to put it lightly a piece of s#!t. I completely dropped the ball on getting our AI to work, not even managing to get something together. We didn't even have one boss at the end. I ended up being sick for a few days in the second week, losing me a lot of time and leaving my team in a bad spot. I think all that I personally ended up contributing when everything was said and done was some crappy knockback code. Travis did a majority of the work. Our project also wasn't packaging correctly for reasons I have no idea. Overall, a pretty terrible experience that I blame myself for.

- What did learn?

  I think that I learned a lot about deciding a game's scope. Making a boss fight is a fairly complex matter and not something that anyone should underestimate. This is pretty much an entirely new enemy that will likely not be reused and is held to a higher standard than most. They also require complex AI. I did learn a solid amount about AI, though fundamentally I think I would need to revisit it if I ever want to implement it into a new game. Everything from storing data on a blackboard to using blueprint tasks that trigger and manipulate those values. I learned a fair bit about resilience and determination. While the end product was poor and we struggled with making our game, learning to deal with a product just not coming together is invaluable.

Velvet Worm Simulator

- Team list - 5
    - Me
    - Marvin
    - Eli
    - Calloway
    - Nash*

    *Nash was on a separate team but contributed assets.

Project Description:

This not-a-game was supposed to be a simulator based on being a velvet worm, a species of worm native to South America. It was a 3D third person simulator game where players are able to wander around in a forest and use the velvet worm's patented goo expulsion to trap prey (beetles) and can then eat them. Certain secrets were also hidden around the map for players to find.

- What went well?

    The teamwork experience on this project was awesome. We managed to get a fairly put together project to package and submit for the game jam. I'm pretty happy with the goo code, everything from how it sticks to the wall and adapts to the face of the object that it hits to how the gooify effect on the beetles looks (even though it's just a timeline, but it's better than just randomly appearing). We got animations to work which was pretty cool because I hadn't worked with animation blueprints before. I'm pretty happy with how the adaptable worm body ended up, even though it's not even close to perfect. I'm happy i got a simple AI to work for the beetles, I feel like that makes up a little bit for my failure on the last game jam.

- What went wrong?

    I feel that our game wasn't really fleshed out. Sure you can wander around and eat beetles and a counter goes up, but that's really it. Otherwise it's terribly boring. Certain things needed refinement, such as worm body movement and our goo shooting stuff. I wish we had grass and plants in the level, and a more cohesive art style as a whole. I don't feel like I used my time as well as I could, leading to an inferior product.

- What did I learn?

I learned that I like smaller teams. I learned who I liked to work with more. I learned how to use animation blueprints by making our worm move and AI behavior trees by making our beetles move around randomly. I learned about timelines and normal values to make things face the center of an object. I learned about timers more when making our beetle spawners. Hell, I think I even learned a little about materials.

Tiny Titans

Project description:

Tiny Titans was my main project that I worked on for the majority of the year. It was to be a 3D, top town, local multiplayer arena combat game meant for the table where players would assemble their own mechs and fight each other. Player's capabilities would be based on the mechs that they constructed. Players would have access to different abilities that they could use to spice up combat a little bit.

- Team list - 8
    - Me
    - Eli
    - Dylan
    - Calloway
    - Pink
    - Keon
    - Cameron
    - Nash

- What went well?

  Cameron and Nash did a pretty good job with what assets we ended up getting. Once we got it working, getting to put your own mech together was pretty cool. Predictive aiming ended up working pretty well. I think in the beginning there was a somewhat clear idea about the game we were making. We had a pretty decent multi touch system.

- What went wrong?

  I am not a good team leader. We as a collective did not have a single, cohesive idea of what the game was going to be. This resulted in a severe lack of productivity from pretty much everyone. We probably should have held more meetings. We severely lacked art assets to use in our game, leading to it looking terribly generic and boring. I'd argue that the code was a mess, to be fair to myself I was still learning how to code games, but it was a mess. We never established a solid gameplay loop. Overall the game lacked direction.

- What did I learn?

I learned that I am not good at leading teams. I learned that for a game to succeed it needs a strong direction from the beginning. I learned that I have severe focus issues when it comes to longer term projects. I learned that having constant team meetings and making sure everyone is on the same page or doing something that they are passionate about is vital. I learned how to collaborate more on a team. I learned that prototyping fast and failing quickly is better than dying over a prolonged period of time.

CloudTD

Project description:

This game was meant to be a 3D isometric tower defense game. In this game players would place towers in an effort to prevent the evil enemy shapes from reaching their base and destroying it.

- Team list
  - Me
  - Eli
  - Dylan
  - Calloway
  - Pink
  - Keon
  - Cameron
  - Nash

- What went well?
  We started this project with relatively little time left in the year, and I believe that compared to our previous project we got a good number of things done in a good amount of time. I think that the Niagara effects created by Nash are really cool looking. I think we did a good job getting a lot of the basic functionality in. I feel like there are certain ways that the code I wrote improved, primarily in readability. We held regular team meetings which added a semblance of organization and collaboration. I feel we improved in terms of productivity coming from our last project.

- What went poorly?
  I feel like throughout the lifespan of this particular project my motivation was severely lacking, and I feel like I brought down the rest of the team as a result. Call it fatigue from Tiny Titans or call it senioritis, I wasn't on my A-game. I do feel like the team lacked sufficient and clear leadership.

- What did I learn?

I had to learn to make a queue system for spawning our enemies and I think that was pretty cool to use event and function timers to kind of bounce off of each other. I feel like I learned to better communicate with my team and play my role as a programmer. I think I learned to let go a little more. On our previous project I feel like I was controlling over certain things such as the code. With this project I put more trust in my teammates to do a good job and I feel like that was for the better. I used more && and || statements more in my code to simplify some boolean expressions as well.