Peridot

Peridot is a cutting-edge AR mobile pet-sim game. The game is played with camera passthrough AR and uses AI to build a 3d mesh of the world and identify the objects around the player. This lets the player’s pet, referred to as a “Peridot”, navigate around obstacles, swim through water, and stop to sniff the flowers.

I serve primarily as a Unity engineer in our team, but I also do some server, tooling, and tech art work. The nature of a smaller team means I’ve touched most major features in the game, but some specific systems I’ve worked on are friending, gifting, questing, notifications, streaks, news, and telemetry. I work with internal and external partners to integrate services and build native features where engine capabilities did not meet user needs.

Pengu Peril

Global warming strikes again! Pengu peril is a Ludum Dare 50 entry about a penguin on a melting iceberg! Your mission: keep them dry for as long as possible.

I lead Unity development on Pengu Peril and had a lot of fun learning tile map wizardry. Pengu Peril finished top 10% overall.

Lunacia

Lunacia is an atmospheric puzzle / platformer about finding yourself through dreams. I co-lead this project with Yoon Lee. I was involved in all aspects of the game, but I focused on technical leadership, programming, and visual effects.

One interesting system I worked on with this project was a publisher / subscriber system. Early on it became apparent our technical team would be both large and inexperienced. Additionally, our plans for obstacles, respawning, differing visibility states, and a dynamic soundtrack called for a fair amount of interaction between different systems. I decided the best approach would be to implement an asynchronous messaging queue that would allow developers to communicate with minimal knowledge of each other’s systems. The system proved very effective in reducing integration issues between different developers’ work.

Melody Star

Melody Star! is a rhythm game played with a real piano. Instead of pressing buttons on a controller, players use an electronic instrument to play the actual notes of a song! The game follows a cast of lovable characters as they strive to become the best musicians in the land! I lead this project and focused on the tricky technical challenges of creating a system that adapted songs into beatmaps for input from any midi controller or keyboard.

Mi So Madness

Mi So Madness is a multiplayer infinite runner that uses the Steam Online Subsystem for its networking functionality. I contributed c+ and blueprint networking code for some of the internal systems.

Path to Harmony

Path to Harmony is one of the games I'm most proud of. I led this project as part of VGDev, Georgia Tech's game creation club, and was involved in every aspect of its creation. The game is a tactics game which follows the adventures of Blair, an idealistic strategist in the Xingatan army, as they find victory in battle and discover uncomfortable truth’s about their country. Over six months my team made a fully featured game with a compelling narrative, soundtrack, dozens of levels and units, and deep game play. Heading the project let me develop leadership and project management skills, in addition to game development skills. PtH was my first time architecting a project of its scale, and I learned an enormous amount from my successful decisions as well as the ones that caused growing pains.

RVVR

RV VR is a game about a family road trip through the cosmos, taco bell, and aliens. It was created for GGJ with several of my good friends. I served primarily as a gameplay programmer. Among other things, I created a system that applied different audio and visual effects based how the player collided with an obstacle.

Won honorable mention at GGJ GSU 2019

System.Exit

System.Exit was my first game with VGDev, Georgia Tech’s game development club. I had the fortune to work under a patient lead who helped me learn skills that are necessary to work on larger projects, such as developing with an established code base. My work on System.Exit focused on development; I designed and implemented mechanics, items, enemies, and battles.

System.Exit also taught me the value of play testing. Our intuition based balancing efforts produced clunky power curves, and the enjoyability of the game improved markedly as we iterated.

Trouble on Beryl Isle

Trouble on Beryl Isle is a sunny platformer I worked on through VGDev. I coded AI for an enemy in addition to creating levels and environments.

Clean my Desk

Clean my Desk, another VGDev game, is about procrastination, interior decoration, and art(?). I primarily contributed models and textures, as well as some code. This project was a great experience learning how to model and texture a large number of assets efficiently. I worked with Blender and Substance Painter.

Dreamworld

Dreamworld is a walking simulator about alien worlds I made with a few friends in high school. I drew environment pixel art.

Hengliding

Hengliding, a Tamagotchi meets Mario Kart game about raising chickens. I co-lead this game with my dear friend Sarah Tsai and contributed architecture and featuring code, as well as models, animation, and more!