##Nuclear Purr

##Video Link

Video [link](https://youtu.be/tYvtK8EYyqc)

##Build Link

Game:
[link](https://drive.google.com/file/d/1_cmB2xFf1nkEdmF3Bcck8AQuaKK897bA/view?usp=sharing)

##Game Description

Nuclear Purr invites you to embark on a thrilling post-apocalyptic adventure as a courageous cat cafe owner. Roam the radioactive wasteland, rescuing adorable cats and gathering essential resources to sustain your sanctuary. Battle scavengers and survivors, earning upgrades to fortify your defenses and protect your beloved feline companions. With each daring expedition, forge a haven amidst the chaos, proving that love and determination can thrive even in the bleakest of times. Can you navigate the dangers of the nuclear world, ensuring the safety of your cafe and its whiskered inhabitants? Unleash your inner hero in Nuclear Purr and experience a heartwarming tale of survival, bravery, and the enduring bond between humans and their purrfect companions.

##Technical Decisions

We chose to make our game a top-down 2D to cut down on the physics we'd need to implement in movement so we could focus on the combat, enemies, and other mechanics.

We chose to use Godot 4.1.2 because several of our group members have recent experience with it, and for those who haven't, it's a good learning experience. Additionally, Godot is a good choice for 2D games, which is another reason we chose it

Aseprite and Gimp were used to produce our art due to our artists' familiarity with it. We also used Itch to help supplement our art, primarily in regards to the tileset to lessen the workload of our artists. The artists also chose to work primarily in a 64 by 64 sprite configuration because of the amount of detail we could achieve with it. We were also confident in our ability to make high quality pixel art at such a resolution given the amount of time.

We have camera follow the player because it was easier to implement this with dynamic levels.

##Design Goals

Our biggest goal was to make a playable dungeon crawler game with implemented combat, upgrades, and at least two different enemies. In the current state of the game, we have finished a group of levels for the player to progress through, two enemies with different behaviors and attack methods, and two different ways for the player to attack. We have also implemented basic UI such as a pause menu, main menu, and death screen, and managed to finish a currency system to buy upgrades to player stats such as rate of fire, damage, speed, and health from the cafe shop. In short, we have finished the main game loop, accomplishing our goal of a playable dungeon crawler.

## Inspiration
- Binding of Isaac: top-down shooting mechanics, generative levels
- Soul Knight: Melee roguelike
- Dave the Diver: Idea of going in between two different game loops with minigames in the cat cafe and the main dungeon crawler

## What it does
- Allows the player to explore generated levels for variation in gameplay
- Lets the player upgrade and build their cat cafe and abilities to go further into the dungeon
- Allows for melee and ranged attacks with an auto attack on the back of the player for a little extra help from your furry friend.

## How we built it
- Built in Godot 4.1.2

## Challenges we ran into
- It was some people's first time using Godot so we ran into a learning curve which slowed development
- We ran into several problems with Github merging and reverting
- Some sprites refused to scale while we were rushing to finish, resulting in some weird enemy sizing
- The loss of our audio engineer the day before the jam started ended up in us not being able to get audio for this version of the game

## Accomplishments that we're proud of

- The functionality of all the damage interactions and collisions
- The assets and animations we were able to integrate
- How well we worked together, and how much we were able to get done
- The enemy functionality and their ability to detect, attack, and die to the player being transferable across the two different enemy types
- The dungeon manager singleton was super easy to set up but became integral to how the game functioned. Additionally, the load_level function was created and works universally when loading an active screen.

## What we learned

- Fundamentals of GDScript and Godot node, parent-based structure
- Animation and asset integration
- Collision and Movement
- Interactions
- UI elements

## What's next for Nuclear Purr

- Proper audio and music
- More levels
- Boss(es)
- Bringing different cats that do different things
- Cat Cafe Minigames and upgrades
- More Weapons, Gear, Resources
- NPCs and traveler merchants

## Epilogue

We could not have done any of this without the gracious support and presence of Ludwig the cat. He was a great addition to our team. He stayed near us the entire project and kept us going.