Plastic Protocol GDD

Scope

A third person shooter about an emergent AI toy comes to the world's rescue. Where scale, and improvised tools create emergent gameplay.
The initial level is one of awakening and you must stop the enemy toys before they can escape the toy shop. The player must find the fuses around the level to turn the power back on, thus gain access to the other floors.

Player Psychology

Aimed at the explorers and achievers

Design Pillars

1. Scale driven tactical combat
    - The shop contains cover, traversal objects and combat areas all defined by oversized objects.
    - Spatial awareness and vertical gameplay is the way rather than aim and shoot skills.
    - Encounters are about using the landscape as much as shooting the enemy
2. Improvised tools over conventional weapons
    - Not standard military equipment
    - Weapons are various types of nerf guns and batteries
    - Traps can be made from these items and objects like staplers
    - Encourages experimentation with loadouts

Core Gameplay Loop

1. The player enters a new zone.
2. Enemies begin moving towards the player.

3. The player uses weapons/traps to stop them.
4. The player traverses the environment to a new zone.
5. System pressure increases.
6. Loop repeats until the player reaches win state.

Player Mechanics

Set in third person, the character is a military style toy that will be able to:

- The character will have a jetpack ability to traverse the environment.
- Basic Shooting - Nerf Pistol, Nerf Rifle, battery grenades.
- Traps - improvised devices (eg post-it note pad, drawing pins - spike trap).
- The jump and crouch mechanics are bespoke to the character.
- The animations for movement, weapons, firing have all been created/adapted to be used with the character.
- Jetpack ability, with limited fuel. Regenerates after landing on the ground.
- The character will be able to aim their weapon in all directions to combat the enemy.

Enemy Design

Initially one enemy -

Action figures, similar to the player character, start either encased in their boxes or already broken out of them. They have had their safety features fried due to the pulse from the storm. They want to do nothing else, than to break free of the store and cause havoc to the rest of the area.

- Breaks out of their boxes when the player comes near.
- Simple behaviour tree, avoid obstacles or attack player.

- Advanced behaviour, swarm AI or second enemy.

Environmental design

Toy shop -

- Floor section for combat space.
- Elevated areas, shelving, for traversal sections.
- Interactions - push objects, switch interactions
- Weapon pickups - nerf type ammo boxes, packets of duracell batteries

Weapon Design

- Pistol - Nerf type pistol, does minor damage to the enemy. Can be used to affect the environment - firing nerf bullets.
- Assault rifle - Nerf type rifle, does extra damage to enemies. Several shots can destroy some environment type - firing Nerf bullets.
- Grenade - round watch type batteries , does area effect damage. Environments can be destroyed in a single attack.
- Grenade Launcher - fires the above grenades a larger distance.

Component Mechanics

EntityUpgradeComponent

- Contains a EntityStats PrimaryDataAsset, which contains multiplier values for stats such as damage, speed and health. Also contains values and functionality for XP and levelling up. When attaching to an object, you can access the EntityStats in the details pane, and edit the stats of the entity there. You can also get this through C++ and Blueprint, enabling getting and setting each value during runtime. When levelling up, an event OnLevelUp is called, allowing blueprint to react to the levelup.

HealthComponent.

- Manages the health of the object, using unreal engines OnTakeAnyDamage() event. This must be hit by a valid projectile. OnTakeDamage, OnHeal and OnDeath are events that can be recieved in blueprint.

Projectile base class.

- A C++ class to be inherited from that contains unreal engines projectile movement, and a damager that uses unreal engines ApplyDamage event. Properties such as the initial speed, max speed, bounciness and homing ability is edited in the projectile.

ProjectileSpawner

- Spawns a selected projectile that inherits from the ProjectileBase class. Properties such as the fire rate and enabling autofire. The projectile is spawned from blueprint using the SpawnProjectile function

UI

The UI is basic with HUD showing health, stamina, weapon type and ammo count per weapon.

- Weapon name will show on the HUD when selected by the player with the corresponding ammunition.

- Jetpack fuel will be a progress bar that shows the amount of fuel left in the tank.

- Health will also be a progress bar showing both the damage and updating when the health is restored.
- Aiming will show a closer, first person style view from that of the player and as the player is an AI toy, it will be a computer stylised viewport.

Audio

Toy like noises, squeaks, beeps, blips. Environmental track changes in different toy sections. The player weapons will be audio of various nerf guns firing.

Art


NERF
LOADOUT
SHADOWSPEED RECON


NERF
MODULUS
BARRELSTRIKE
CAUTION:


SHOOTING GAMES
BARREL ROCKET
AIR POWERED GUN


NERF
WARNING:
ATTENTION:
ADVERTENCIA:
ZOMBIE STRIKE
ZOMBOMBZ