Candy-Themed Offensive and Defensive Items

This document contains 30 candy-themed offensive and defensive items for your game 'Candy Rush Grand Prix.' Each item comes with a step-by-step guide on how to create its blueprint in Unreal Engine 5.

# Lollipop Whirlwind

Category: Offensive

Description: A spinning lollipop that creates a vortex pulling in nearby racers.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Lollipop Whirlwind, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Jellybean Mines

Category: Offensive

Description: Exploding jellybeans that act as mines on the track.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Jellybean Mines, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Candy Cane Snare

Category: Offensive

Description: A ranged attack that pulls other racers closer.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Candy Cane Snare, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Licorice Rope

Category: Defensive

Description: Blocks one attack and slows down the attacker.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Licorice Rope, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Peppermint Speed Burst

Category: Defensive

Description: A sudden burst of speed to escape or gain an advantage.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Peppermint Speed Burst, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Gummy Bear Barricade

Category: Defensive

Description: A shield that can be dropped to form an obstacle.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Gummy Bear Barricade, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Cotton Candy Cloud

Category: Offensive

Description: Reduces visibility of racers behind the player.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Cotton Candy Cloud, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Chocolate Syrup Spill

Category: Offensive

Description: A slippery trail left on the track that makes racers lose control.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Chocolate Syrup Spill, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Taffy Slingshot

Category: Offensive

Description: A sticky projectile that slows down the targeted racer.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Taffy Slingshot, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Marshmallow Cushion

Category: Defensive

Description: Absorbs damage and bounces off walls without slowing down.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Marshmallow Cushion, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Candy Corn Cannon

Category: Offensive

Description: Shoots rapid-fire candy corn at opponents.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Candy Corn Cannon, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Sour Sugar Rush

Category: Defensive

Description: Gives the player a speed boost with a sour kick.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Sour Sugar Rush, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Gumdrop Shield

Category: Defensive

Description: Surrounds the kart with a protective gumdrop barrier.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Gumdrop Shield, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Bubblegum Bomb

Category: Offensive

Description: A bomb that sticks to other racers and detonates.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Bubblegum Bomb, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Pop Rock Grenade

Category: Offensive

Description: Explodes on impact and stuns other racers.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Pop Rock Grenade, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Licorice Whip

Category: Offensive

Description: Whips racers in close proximity, slowing them down.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Licorice Whip, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Fudge Avalanche

Category: Offensive

Description: Drops large fudge boulders behind the player.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Fudge Avalanche, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Cotton Candy Net

Category: Defensive

Description: Catches incoming projectiles, stopping them in their tracks.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Cotton Candy Net, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Chocolate Shield

Category: Defensive

Description: Blocks all attacks for a short duration.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Chocolate Shield, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Candy Cane Spike Trap

Category: Offensive

Description: Deploys spikes that pop other racers' tires.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Candy Cane Spike Trap, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Peppermint Portal

Category: Defensive

Description: Teleports the player ahead on the track.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Peppermint Portal, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Sour Patch Blockade

Category: Offensive

Description: Creates a wall of sour patches that other racers must navigate around.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Sour Patch Blockade, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Molasses Trap

Category: Offensive

Description: Slows down racers who drive over it.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Molasses Trap, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Caramel Rope

Category: Offensive

Description: Binds a racer in sticky caramel for a short period.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Caramel Rope, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Jawbreaker Shield

Category: Defensive

Description: A tough shield that absorbs multiple hits.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Jawbreaker Shield, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Chocolate Volcano

Category: Offensive

Description: Erupts chocolate lava onto the track, creating obstacles.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Chocolate Volcano, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Cotton Candy Ramp

Category: Defensive

Description: Creates a ramp to jump over hazards.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Cotton Candy Ramp, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Fruit Gusher Explosion

Category: Offensive

Description: Explodes on impact, covering racers in sticky fruit goo.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Fruit Gusher Explosion, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Candy Confetti Blaster

Category: Offensive

Description: Fires a blast of candy confetti, momentarily blinding other racers.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Candy Confetti Blaster, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Peppermint Shuriken

Category: Offensive

Description: Throws peppermint shuriken that slice through the air.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Peppermint Shuriken, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.

# Jelly Trap

Category: Offensive

Description: A jelly puddle that makes other racers slip and lose control.

## Step-by-Step Instructions:

1. In Unreal Engine 5.3, open the Blueprint Editor and create a new blueprint class.  
2. Use the Actor class as a base.  
3. Add a Static Mesh component for the visual representation of the item.  
4. In the Event Graph, set up the functionality:  
 a. For Jelly Trap, use a Projectile or Trigger Volume component (depending on the item).  
 b. For offensive items, use a Projectile Movement Component and add logic to detect collisions.  
 c. For defensive items, create shield or buff systems.  
5. Set up the item’s behavior when activated (e.g., damage other players, block attacks, etc.).  
6. Customize the movement, lifespan, and any effects (e.g., explosions, speed boosts, or slowing effects).  
7. Once done, compile the blueprint and test in-game.