Arena.setPlayerTeam()

* Arena.Player.$ShapeBuilder.setTeam(“player”)
  + Player’s shape is now set to be on the player team

Arena.SpawnEnemy()

* Enemy onReady() -> Randomize()
  + Enemy.emit\_signal(“change\_team”,”enemy”)
    - Sends out a signal connected to enemy’s ShapeObject
      * Enemy.ShapeObject.on\_change\_team(team = “enemy”)
        + ShapeObject.GoodArea.add\_to\_group(“enemy”)

Enemy collision are now set to the “enemy” team

* + Enemy.RandomizeShape()
    - Enemy.ShapeObject.DeleteUnusedShapes()
      * Enemy decides what shape it wants to be
      * Enemy.RandomizeBlasterPlacement()
  + Enemy.RandomizeBlasterPlacement()
    - Enemy.SpawnBlaster() x random num of times
      * + Enemy has blasters placed on random sides (if difficulty is high enough. At lower levels, don’t give them blasters)
  + Enemy.chooseSpawnPoint()
    - Enemy’s spawn point is chosen randomly
  + Enemy.randomizeVelocity()
    - Two outcomes: static velocity or follower
      * If static: choose random direction and stay that way
      * Follower: follow the player’s coordinates

Difficulty ranks:

1. At first, enemies just move in random directions from off screen.
2. They start following you (and you have to shoot them to get to the next round)
3. Some enemies now have blasters.
4. Subsequent levels, give more enemies blasters, then give more blasters to singular enemies. (number of enemies++, then number of enemies with blasters++, max number of blasters to give to enemies++, loop)

Global vars: MaxNumOfEnemies MaxNumOfEnemiesWBlasters, MaxBlasterNum. Increase one of these each level.

Win conditions: make sure every enemy is off the screen.

ShapeObject.setTeam(team)

* GoodArea.add\_to\_group(team)

# Enemy

KinematicBody2D – EnemyBody