# Arena

KinematicBody2D player

* ShapeBuilder

Camera

AnimationPlayer

Timer EnemySpawnTimer – how long until another enemy should be spawned? Default: 1 second

Timer roundTimer – how long until we should stop spawning enemies?

# ShapeBuilder

Shapeobject Player

Blaster – The component you click and place on the shapeObject

RichTextObject instructions - has some default text “pick a side!...to add a component to.”

Node2D InputManager – manages the keys pressed. If any of the fire keys are pressed, it releases the signal \_side\_fire\_key\_pressed (key) to fire all blasters matching that key

UIBox UIBlasterBox – gray ui box that goes behind the blaster. When clicked, adds a blaster the player can add to their shape (if possible)

# UIBox

Node2D Box – contains the sprite and nine patch. Primarily used to modulate both of its children at the same time

NinpatchRect – the colored rectangle behind the sprite

RichTextLabel – the label of the box

Area2D – the clickable area of the box

CollisionShape2D – use for the Area 2D to define clickable area

9Patch hoverBox – transparent box that appears when player mouses over the button

# ShapeObject

Mostly handles the movement of the player, but also spawns in its own collision based on chosen shape.

String shape – what shape is this object using? Default: “Square”

Bool isPlayer – is this the player character? Used for movement

KinematicBody2D Player – Handles the movement of the player. Spawns in a Playercollision scene based on what shape the player has selected.

Square/Triangle/CircleObject Playercollision – Custom object that changes based on what current shape the player has selected. Have all three at once on spawn and then just delete the collision shapes that are NOT being used by the player.

Blaster – the blasters the player placed on the shape

Shield – the shield the player places on the shape

# Square/Triangle/Circle Object (all their own objects)

This handles the collision if the player chooses a specific shape

Sprite square– the square the player sees

CollisionShape2D 2x – defines the outside edges where the blasters can be places

Area2D goodArea – used for blaster placement. If the blaster is only touching this area, it can be placed.

Area2D BadArea – used for blaster placement. If the blaster touches this area, it is too far in the shape and cannot be placed.

Area2D \_\_Edge – used to determine what angle the blaster should be when placed. Make sure its group name is <degree in 3 digits>DegreeEdge, so the blaster auto turn based on its group name

# Blaster

Vars: assignedKey

animatedSprite blaster – the sprite. Has three animations: normal, green (placement is acceptable), red (placement is unacceptable)

Area2D boundry – handles collision for placement that is acceptable

Projectile – the projectile that will be created and shot out when player presses the assignedKey

# Projectile

bool isPlayerProjectile: was this shot by the player? Or by an enemy? Default: off

vector2 velocity: in what direction is this projectile going?

kinematicbody2D – allows the projectile to move

sprite projectile: the shape of the projectile

collisionbody2D – handles collision

# Config

String key

Export string action – is this fire\_<number>?

RichTextLabel Instruction: “Press me to shoot this side!”

RichTextLabel ChangeInstruction “Click me to change key” or “Press the key to be assigned”

9Patch Box

9Patch hoverBox – the semi-transparent box that appears when player hovers over it

RichTextLabel keybutton – displays the key pressed to fire the side

Area2D clickArea – area to click if the player wants to change the key

Collision2D

Custom classes

# Edge

Used for Area2Ds that contain the degrees that the blaster should turn when they touch. Attach this script to the object, and make sure the object is in group “Edge”!

Var angle – what angle should the blaster turn?

String assignedKey – which key is this connected to? Ex. The W key will make all blasters on this side fire