Arena.triggerUpgrade()

* Arena.pauseGameplay()
  + ShapeBuilder.PausePlayerMovement()
  + Arena.PauseEnemySpawn()
    - Player cannot move and no enemies spawn while player selects their upgrade
* Arena.zoomInOnPlayer()
  + Player.fadeInUI()
    - Instructions start to fade in so player knows what to do
  + Camera.ZoomInOnObject(delta, player)
    - Player fills up the screen so they can build their next shape

Arena.\_on\_Camera\_finish\_zoom()

* ShapeBuilder.triggerUpgrade()
  + ShapeBuilder.emit\_signal(“player\_started\_building”)
    - Blaster.disable()
* BlasterBox.setText(“Blaster\n x1”)
* Player.disableMovement()
  + Player cannot move while picking an upgrade
* GlobalFuncs.AddAnim(instructions, fadein, 1 second)
* GlobalFuncs.AddAnim(BlasterUIBox, fadein, 1 second)
* GlobalFuncs.AddAnim(instructions, fadein, 1 second)
* GlobalFuncs.AddAnim(UIBlaster, fadein, 1 second)
* GlobalFuncs.AddAnim(BlasterCount, fadein, 1 second)
  + Player now sees the instructions and can pick an upgrade

GAME HALTS UNTIL PLAYER CLICKS ON UIBOX

* BlasterBox.mouse\_entered()
  + BlasterBox.toggleShade()
    - The button is reactive when player mouses over it, so they know it is interactive
* UIBox.Area2D.input\_event() - UIBox emits signal for shapeBuidler to spawn itme
  + ShapeBuilder.\_on\_UIBox\_clicked(item: Node2D)
    - UIBox.decreaseItemCount()
      * BlasterBox.setText(“Blaster\n x<amount>)
      * If no more items in UIBox: UIBox.disable()
      * A new blaster is created and given to the player. UIBox decreases its count of items, and if there are no more items it disables itself.

* Blaster.toggle\_Pickup()
  + Blaster.changeAnim(“BadSpot”)
    - * Blaster turns red so player knows they can’t place it
  + Blaster.follow\_mouse()
    - Blaster now follows the mouse cursor

GAME HALTS UNTIL BLASTER IS IN VALID SPOT

* Blaster.boundry.area\_entered()
  + Blaster.changeAnim(“GoodSpot”)
    - Blaster turns green so player knows it is a valid spot
  + Blaster.boundry.getAreaAngle(area) – get rotation of area that was entered
    - Blaster.boundry.rotateBlaster(rotation of area)
      * Boundry emits a signal to the Blaster parent so the thing will rotate correctly
  + Blaster.rotateSignalReceived(angle)
    - * Blaster rotates to fit the angle of the shape side the player wants to place the blaster on
* Blaster.boundry.area\_exited()
  + Blaster.changeAnim(“BadSpot”)
    - Blaster turns red so player knows spot is invalid. Might act funky if it exits one area yet enters another?

GAME HALTS UNTIL PLAYER CLICKS AND BLASTER IS PLACED

Blaster.boundry.input\_event()

* If bad:
  + Playsound(error)
* If good:
  + Blaster.togglePickUp()
    - Blaster places itself and releases item\_placed
  + ShapeBuilder.on\_item\_Placed()
    - For each UIBox in scene:
      * UIBox.getItemCount()
      * If every itemCount = 0
        + ShapeBuilder.AnimationPlayer.play(“FadeOutUI”)
        + ShapeBuilder.emit\_signal(all\_items\_placed)
  + Arena.on\_all\_items\_placed()
    - Arena.AnimationPlayer.play(“ZoomOut”)

ShapeBuilder. \_on\_AnimationPlayer\_animation\_finished (“FadeOutUI”)

ShapeBuilder.emit\_signal(“Player\_finished\_building”)

* GlobalFuncs.AddAnim(instructions, fadeout, 1 second)
* GlobalFuncs.AddAnim(BlasterUIBox, fadeout, 1 second)
* GlobalFuncs.AddAnim(instructions, fadeout, 1 second)
* GlobalFuncs.AddAnim(UIBlaster, fadeout, 1 second)
* GlobalFuncs.AddAnim(BlasterCount, fadeout, 1 second)
* Player.enableMovement()
  + Player can move again
* Shapebuilder is done, then emits a signal to the arena scene so it can zoom out the camera back to normal gameplay

Some sort of zoom out wi

MOVING THE PLAYER

Player.physics\_process

* Player moves