## Interaction Control (Done)

|  |  |
| --- | --- |
| Class | InteractionControlClass |
| Inheritance | Parent Class |
| Description | Classes for objects that can interact. All of these deal with the position, animation and material changes for interactions, NOT the interactions themselves. |
| Fields | Animator anim  IndicatorClass indicator |
| Constructors |  |
| Functions | setPosition(transform)  setAnimation(string, bool)  setAnimationTrigger(string)  setIndicator(bool)  updateThisInteraction()  setAngle(vector3)  unsetAngle() |
| DONE | D |

## Interaction Classes

|  |  |  |  |
| --- | --- | --- | --- |
| Class | InteractionClass | EnergyInteractionClass | InputInteractionClass |
| Inheritance | Parent | InteractionClass | InteractionClass |
| Description | This will keep interaction control items. This can be input or energy interactions, but this will mostly take inputs and set objects. | This is for interactions that will affect turning off and on an energy object in a grid.  Note: This takes another energy screen object as an option that would provide the ability to lock the set object based on the input on the screen. | This is for interactions that would take a screen energy object and change the input in that system. This will be only a screen obj.  This will set that screen object with the new input. |
| Fields | interactType[] permittedInteractions  InteractionControlClass controller | gridManager powerBox  energyObj obj  ScreenClass lockObj  Boolean isOn  Boolean isPowered | String input  ScreenClass obj |
| Constructors |  | setPowerManager(Energy)  getIsOn() |  |
| Functions | Interact(transform, transform)  Interact(Vector3, Quaterion)  Interact(Gameobject)  isInteractionType()  setController() | setObject()  turnoffObject() | setInput(input) |
|  | D | D | D |

|  |  |  |  |
| --- | --- | --- | --- |
| Class | GridInteractionClass | GeneratorInteractionClass | HoldItemInteractionClass |
| Inheritance | InteractionClass | InteractionClass | InteractionClass |
| Description | This interaction works with turning the grid on and off. | This interaction works with turning off and on the generator | This interaction positions an item to the held position for the player.  This uses the position based Interaction. |
| Fields | SystemManager powerBox  Boolean isOn | SystemManager powerBox  Boolean isOn  Float maxPower | Boolean isHeld  Gameobject connectedObj  holdType type  Gameobject currentHolder  Transform anchorPos  Float anchorThreshold  Float stretchThreshold |
| Constructors |  |  |  |
| Functions | setObject(bool)  setToOff() | setObject(bool)  setManager(SystemManager) | setObject(Vector3, Quaterion)  setSystem(gameobject) |
|  | D | D | D |

|  |  |  |  |
| --- | --- | --- | --- |
| Class | ItemPositionInteraction Class | PlayerControlInteraction Class | CombinationLock Class |
| Inheritance | InteractionClass | InteractionClass | Interaction Class |
| Description | This interaction would get a current held item and input it onto the current target position. | This can be activated through an interaction. This will complete certain control aspects of the player.  For example, change camera, lock movement, lock rotation, activate keyboard mode, ect.  Default is lock movement and change camera position. | This is the combination interaction to rotate the combination. When it is set, it will set a connected energy or lock object. |
| Fields | Transform targetPos  gameObject currentHeldItem  holdType[] permittedTypes  string permissionOverride  gameObject connectedObj | FPSController player\_  Bool isOn | float rotationPositiveOffset;  float rotationNegativeOffset;  bool isRotating;  Quaternion newRotation;  float timer = 0; |
| Constructors |  |  |  |
| Functions | setObject(Vector3, Quaterion)  canHoldItem(obj)  setCurrentHoldItem()  hasPermission( interactionType)  OnTriggerStay() |  |  |
|  | D | D |  |

## Energy Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Class | EnergyObject | ScreenClass | Light Class |
| Inheritance | Parent Class | EnergyObject | Energy Object |
| Description | Classes for objects that use the energy system. This can be turned on and off, and completes a certain action when activated.  This is for all objects which require energy to function. This includes electric doors, screens, lights, etc. | This class shows an output and takes input from input interaction classes. If on, shows input, if off, shows error. Can also be used in energyInteraction classes as a locking mechanism. | This is specific to light objects to turn on and off lights. |
| Fields | Int energyUsage  Bool isOn  Bool isPowered  GridManager powerBox | Text text\_  Int inputSize  String commandCode  String currentCode | Material[] switchMats  GameObject light  MeshRenderer rend\_ |
| Constructors | setEnergyManager  (GridManager)  getEnergyAmount()  bool isObjectOn() | setEnergyManager  (GridManager) |  |
| Functions | powerObject(bool)  useObject() | addString(string)  clearString()  bool isCurrentCode()  displayText(string)  powerObject(bool) |  |
|  | D | D | D |

|  |  |  |
| --- | --- | --- |
| Class | Door Class | Lock Class |
| Inheritance | EnergyObject | Energy Object |
| Description | This class opens and closes a door animation based on whether it is turned on or off. | When activated, this object would change the permissions of a specific interaction to stop the player from using it until it is unlocked again. |
| Fields | Animator anim |  |
| Constructors |  |  |
| Functions |  |  |

## Manager Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Grid Manager | System Manager | CameraManager |
| Inheritance |  |  |  |
| Description | This is the manager of the grid. It keeps all the energy objects and manages interactions to turn them on and off.  This also turns off all switches if too much power is routed through it. WARNING: This will look for all energyInteraction classes currently under it’s system. No others. Need system for generator as well. | This will manage several grids, turning them on or off. Each system will function on it’s own. A system manager can also be linked to a generator position. It doesn’t know of this connection, but plugging in, turning on and off a generator will turn the systems on and off without tripping the switches.  This is usually figure out if generatorPower starts at 0 or not. Any internally powered system manager cannot be changes as it has no connected position. | This manager switches virtual cameras by turning them on and off.  This manager can be found on the **player**, and can be used by any system that requires a camera. |
| Fields | EnergyObjects[] objs  Bool[] objectOn  Bool gridPowered  Bool systemPowered  SystemManager manager\_ | GridManager[] objs  Bool generatorPowered  Float generatorPower  Float currentPower  Generator gen | CinemachineVirtualCamera[] cams\_;  int curCam; |
| Constructors | setManager(SystemManager)  getPowerUsed() | setGeneratorPower(float)  setGenerator(bool)  getSystems() |  |
| Functions | turnOnObject(EnergyObject)  setGrid(bool)  updatePower()  getIsOn()  updateGrids()  updateTheGrid();  setSystem(bool) | depowerGrid()  updateThePower() | updateCamera(int)  findCamera(VirtualCam)  setCamera(int) |
|  | D | D | D |

|  |  |
| --- | --- |
| Class | Inventory Manager |
| Inheritance |  |
| Description |  |
| Fields | GameObject[] theItems  InventItem[] inventoryItems |
| Constructors | setManager(SystemManager)  getPowerUsed() |
| Functions | turnOnObject(EnergyObject)  setGrid(bool)  updatePower()  getIsOn()  updateGrids()  updateTheGrid();  setSystem(bool) |
|  |  |

## Inventory Classes

|  |  |
| --- | --- |
| Class | InventItem |
| Inheritance | Parent Class |
| Description | A class that would keep all the different types of inventory and the common fields for them.  Unsure what items would need there own child class, but will find out. This basic will be set up for items that can be used based on their own id, like keys. |
| Fields | String nameID |
| Constructors |  |
| Functions | addToInventory()  removeFromInventory() |
|  |  |