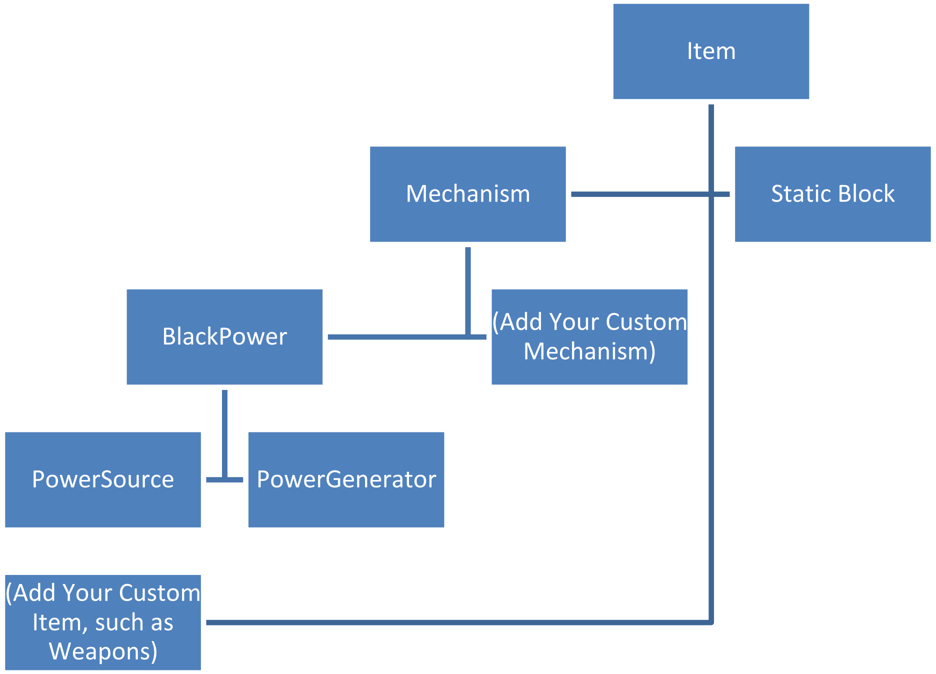
Mechanism Blocks

Documentation

Inherited Scripts


Item
Mechanism
Static Block
BlackPower
(Add Your Custom Mechanism)
PowerSource
PowerGenerator
(Add Your Custom Item, such as Weapons)

- Item is the necessary parent of all items in the inventory. All items inherit this class. You can add any item types as you would like as long as they inherit Item, such as Weapons, Armor, Crafting, etc.
- Static Block is only inherited once. The name says it all.
- Mechanisms are usually powered blocks or unique blocks that have certain mechanisms with them, such as the premade mechanism BlackPower. You can add any mechanisms as you would like as long as they inherit Mechanism.
- BlackPower is a premade Mechanism to show an example of how to make a mechanism block. Feel free to use it in your own projects.
- PowerSource and PowerGenerator are inherited by BlackPower.

Closer Look: Item

Item is the base component to every item in the Inventory.

What it inherits:

None

What it contains:

Variables:

public ItemTypes type	Defines the type of Item it is, feel free to add your custom types to the enum.
public string itemName	Defines the unique name of the item. Also, item icons have to be the same name as the itemName.
public int howMany	Defines how many items are currently in the stack out of the stackSize.
public int stackSize	Defines the number total stack size the item can contain.
public bool inInventory	If the item is currently in the Inventory.
public bool stackable	If the item can stack at all. If this is set to true, it's recommended to set the howMany and stackSize variables to 1.

Closer Look: Item (Continued)

Methods:

All methods are non-unique (get; set;) methods except for:

public void setHowMany	Input variables: HowMany type, int value Return variables: void (null)	Based on the HowMany enum, the howMany variable is either set to the value or increased by the value.

Closer Look: Static Block

This block has no special mechanisms, primarily used for building.

What it inherits:

Item

What it contains:

Variables:

None.

Methods:

None.

Closer Look: Mechanism

Mechanism is the component that makes certain blocks have special mechanisms. The UI will highlight items red if they inherit Mechanism.

What it inherits:

Item

What it contains:

Variables:

public string itemDescription	Describes the description of the mechanism in the MechanismUI.
public List<Mechanism> surroundingMechanisms	This List will return mechanisms in all directions in one radius. It’s recommended to see how BlackPower utilizes this List.

Methods:

public List<Mechanism> SurroundingMechanisms	Input variables: None. Return variables: List<Mechanism>	Use the variable above instead of this method.

Closer Look: BlackPower

A premade mechanism used as an example of how to create mechanisms. This mechanism features power as BlackPower (BP).

What it inherits:

Item ⇒ Mechanism

What it contains:

Variables:

public BlackPowerType maxPower	Sets the BlackPowerType for the BlackPower mechanism.
public int currentPower	The current power of the BlackPower mechanism.

Methods:

All methods are non-unique (get; set;) methods except for:

public void setCurrentPower	Input variables: HowMany type, int value Return variables: void (null)	Based on the HowMany enum, the currentPower variable is either set to the value or increased by the value.

Closer Look: PowerSource

The PowerSource holds BP inside this mechanism per block.

What it inherits:

Item ⇒ Mechanism ⇒ BlackPower

What it contains:

Variables:

public List<PowerGenerator> generators	List of all generators attached to this mechanism using surroundingMechanisms.
public bool generatorAttached	If there is currently at least one generator attached to the power source.
private bool isTiming	If it is currently timing. Generators must be attached for this to ever be true.
static float startTimer	Starting/Reset time.
float timer	Current Time.

Closer Look: PowerSource (Continued)

Methods:

void Timer	Input variables: null Return variables: null	Used for the timer.
void AddPower	Input variables: null Return variables: null	Adds power to the currentPower based on how many generators are attached and what level the generators are.

Closer Look: PowerGenerator

The PowerGenerator generates BP to be connected to a PowerSource based on its level.

What it inherits:

Item ⇒ Mechanism ⇒ BlackPower

What it contains:

Variables:

None.

Methods:

None.

Enums:

HowMany	totalValue	Used to set a howMany variable.
	increment	Used to increment a howMany variable.

ItemTypes	Item	Set the item type. Add your custom itemTypes here.
	StaticBlock	
	Mechanism	
	PowerSource	
	PowerGenerator	

BlackPowerType	lowPower	Has a int power associated to the type, from low to high. Used in BlackPower for power levels.
	medPower	
	highPower	

Additional Help:

I posted a few videos on how to set up MechanismBlocks, even from scratch here:

https://youtu.be/NeeaHLtamls

https://youtu.be/VTIyrWVwdp8

How to add a custom UI:

https://youtu.be/sPJPi5if6Uo

If you have any other questions/concerns/comments, please write me an email at dan.devilbissjr@gmail.com. I will respond ASAP and will not take more than 1 business day.