# Class definitions

# GameObjects

# InventoryObject

## Name

### Definition:

String

Required. No default

The text used to match with user input.

Very basic name for the object, what you might use to talk about that kind of object in general

“basic shoes”

### Rules for use:

Do not begin with a/the

End name with general noun of object:  
 i.e. “small rock”, “basic synth shirt”

Does not need to be unique

Begin with lower case letter

## BasicDesc

### Definition:

String

Required. No default

Used when ‘searching’ the object

A more in-depth definitional name of the object. What you might use to describe the thing to someone buying one for you, differential from other things of the similar type.

“A pair of basic synthetic fabric and rubber shoes”

### Rules for use:

Should be different and more in depth than the name but not include any individuality to the item – that is for inspectdesc

Begin uppercase

## InspectDesc

### Definition:

String

Required. No default

Used when ‘searching’ the object

Some feature to the item that is perhaps unique to that instance of it, is interesting, a reaction to your inspection or a clue to its use within the story of the game.

In common practice the InspectDesc should provide additional information to that specific object, something you might say to differentiate that object from another instance of the same object.

### Rules for use:

Should not include information on the mechanical utility of the object – that is handled elsewhere

Should be a complete sentence with capitalised first letter and full stop. May reference the player: “You notice it is slightly worn at the edges”

## Unlocks

### Definition:

Array of string, variable length

Not required. Default empty array.

Used when attempting to use that object to unlock or unblock a door or furnishing

List may be any length long, include specific key/door IDs as well as more general opening techniques such as ‘prying’ or ‘smashing’ that could be attributes of a crowbar for instance

Will match up with the ‘locked[1]’ attribute of doors and some furnishings – if there is an intersection of the lists then the object will unlock

Form “K000” for ID, user will never see ID however they will know the general use ‘unlocking’ methods available on an object

### Rues for use:

If opens specific object that ID should be in position 0 of the array

General opening methods must be an exact match with an element of these opening methods:

[“YET”, “TO”,” MAKE”,” THE”,” LIST”]

Specific opening ID should be in the form “K000” with literal uppercase “K” and triple digit number

## Expendable

### Definition:

Two element array, [Boolean, integer]

Not required, defaults [False, None]

First element (bool) defines if item is consumable or not

Second element (int) is only used if item is consumable; if first element is True.

Second element defines conceptually how many of the object there are.

Second element decreases as item is used until the item is finally removed from inventory when this number reaches 0

Should not be forgotten that this number is representative of the number of said item purely conceptually, for all purposes this is and should still be treated as a single object that has a finite number of uses.

### Rules for use:

If first element should only be True when this attribute is used, else it should be omitted and left to default

If attribute is used, first element will always be True or attribute should be entirely omitted and left to default

Second element should be integer greater than 0

## Droppable

### Definition

Droppable is a Boolean variable. It defines if an item is determined ‘essential’ and cannot be dropped.

### Rules for use:

There are no specific usage rules for droppable.

## Equippable

### Definition:

Two element array, [Boolean,[Array of string]]

Not required, defaults to [False,[]]

First element (bool) defines if object can be equipped

Second element (array) is only used if item is equippable; first element is true

Second element (array) lists the position(s) that the item is equippable in

Second element can be of any length

### Rules for use:

If first element should only be True when this attribute is used, else it should be omitted and left to default

If attribute is used, first element will always be True or attribute should be entirely omitted and left to default

Second element should only contain exact elements of the possible locations to equip to:

["clothesTorso", "clothesLegs", "armwear", "headgear", "footwear", "armourTorso”, "armourLegs", "accessories", "weaponLeft", "weaponRight", "weaponBoth"]

## Taken

### Definition:

Two element Array [String, String]

First element is who the item belongs to – the player or an NPC.

The second element is the text that will be displayed to the screen when the player picks up the item, for example:

“You pick up the shock baton. Its surprising heft is strangely comforting.” – A brief description that is less in-depth that the inspect description.

### Rules for use:

‘Taken’ defaults to [“No-one”, “You pick up the “] (or something similar), so if the item is generic and isn’t the property of an NPC, this applies.

!! Include your canBlock and doing stuff, ya numpty!

# Furnishing

## Name

### Definition:

String

Required. No default

The text used to match with user input.

Very basic name for the object, what you might use to talk about that kind of object in general

“basic shoes”

### Rules for use:

Do not begin with a/the

End name with general noun of object:  
 i.e. “small rock”, “basic synth shirt”

Does not need to be unique

Begin with lower case letter

## BasicDesc

### Definition:

String

Required. No default

Used when ‘searching’ the object

A more in-depth definitional name of the object. What you might use to describe the thing to someone buying one for you, differential from other things of the similar type.

“A pair of basic synthetic fabric and rubber shoes”

### Rules for use:

Should be different and more in depth than the name but not include any individuality to the item – that is for inspectdesc

Begin uppercase

## InspectDesc

### Definition:

String

Required. No default

Used when ‘searching’ the object

Some feature to the item that is perhaps unique to that instance of it, is interesting, a reaction to your inspection or a clue to its use within the story of the game.

In common practice the InspectDesc should provide additional information to that specific object, something you might say to differentiate that object from another instance of the same object.

### Rules for use:

Should not include information on the mechanical utility of the object – that is handled elsewhere

Should be a complete sentence with capitalised first letter and full stop. May reference the player: “You notice it is slightly worn at the edges”

## Interactive

### Definition

A two-element array arranged in the following structure:

[Boolean, [Array of objects]]

The Boolean defines the object’s ability to be interacted with.

The array defines what objects can be used for this interaction.

### Rules of Use

This isn’t used for anything yet, so there’s no rules for its use.

## containedObjects

### Definition

An n-element array in the following structure:

[object0, object1, …, objectn]

Where object is an inventoryObject

### Rules of Use

Furnishings should be declared after all inventoryObjects are initialised so as to not disrupt the setting of containedObjects.

## opened

### Definition

A two-element array in the following structure:

[Boolean, Boolean]

Where the first item is if the item *can* be opened, and the second item is if it *is* already open.

### Rules of Use

The default argument for opened is [False, None]. You might wonder if that ‘None’ is meant to be there. The answer is that it doesn’t matter.

## locked

### Definition

A two-element array in the following structure:

[Boolean, Array of Objects]

Where the first item is if the furnishing is locked.

Where the second item is an array of the items usable to unlock it.

### Rules of Use

As with ‘opened’, the second item of the default argument will never be used and is hence ‘None’.

Array order currently doesn’t matter.

## Blocked

## Blocking

I didn’t write these. Not my problem.