MicroCosm™
Minimal Object Set

the basic set of objects from which the MicroCosm fantasy world is built

Lucasfilm Ltd. Games Division
January 15, 1986

Introduction

This document describes the basic set of objects from which the **MicroCosm** fantasy world is constructed. In addition to functional objects which the players manipulate, this set also includes scenic objects and objects which are solely internal to the simulation. It is the constitution of this object set which defines the fantasy.

Standard Actions

In the object descriptions which follow, the following phrases are used to describe standard actions for object behaviors. These standard actions are provided because the objects share a common underlying world model and there is much redundancy in the user interface. Standard actions provide consistency.

Broadcast — typed text message is to be broadcast to everyone in the region.

Cease — cease any ongoing activities if possible.

Depends — the action to be taken depends on the contents of the avatar's hand(s). In other words, the action becomes a Reversed Doon the object in the a vatar's hand(s) instead of a Do on the object pointed at.

Go if necessary — if not already at the object's location, go there.

Go and drop — go if necessary. Deposit any item(s) in the avatar's hand(s) at that the object's location.

Go and get — go if necessary. Pick the object up and put it in the avatar's hand(s), if that is possible.

Go and fill — go if necessary. If carrying a potential water container, fill the container from the object.

Goto — walk to the object's location if it is possible to do so.

No effect — nothing happens as a result of doing this.

Throw — throw the object at the indicated spot.

Wear — if wearing, no effect. Otherwise, go if necessary, put on.

Wear or unpocket — if wearing, remove item(s) from pockets. Otherwise, go if necessary, put on.

Remove — if wearing, take off. Otherwise, go and drop.

Miscellaneous Comments

Day and Night

In the world we have a concept of day and night. Time passes. The time of day is determined by the host, and probably follows a fast clock (e.g., six hour days instead of 24). When it is night, it is dark throughout the world (though if the world is cylindrical we might want to implement actual time zones with moving dark and light areas). Darkness is indicated by setting the color map so all the colors are dark and contrast is poor. Various means of artificial illumination can light up a region as if it were daytime. Dark and light is determined on a whole-region basis.

Containers

Many objects are *containers*. That is, they can carry or hold some number of other objects. Each container has a list of its *contents* as one of its properties. The most important characteristic of a container, from a functional standpoint, is that when you move a container from one location to another you also automatically move all of its contents.

All containers have a notion of whether they are *open* or *closed*. Some containers can be opened or closed on command, while others are just permanently open. The distinction is that a closed container is a black box: all that you see is the container itself and you have no information about its contents. The contents of an open container, on the other hand, are visible to any observer.

Some objects are obviously containers, such as boxes or backpacks. However, there are other objects which are containers of a more subtle type. For example, avatars are containers of the permanently open variety. Anything that an avatar is wearing or carrying is said to be contained by the avatar object. This means that whenever the avatar walks around or teleports to another region, his clothing and possessions go with him.

Performing a Get action on some kinds of containers while you are holding them activates a menu display mechanism that lets you select from and examine the contents. This typically happens with containers whose screen images are too small to allow any plausible display of the contents without ''zooming in'' for a close-up look. The screen is replaced with a stylized view of the container's contents: a grid of icons each representing one item. These icons are simply the canonical images that would be used to display the objects were they not inside the container. You can then further select items or return to the main display using a Put action.

Transportation

The **MicroCosm** transportation system is what allows an avatar to get from place to place in the wide, wide world. There are basically four ways of getting around: walking, driving a vehicle, using the teleport booth network, or taking the bus. Except for walking, these all hinge on objects of one sort or another to work.

Teleport booths, from the player's point of view, are actually the simplest way to get around. You just drop a coin in the slot, dial a number and **poof**, there you are. It's just like making a phone call (though more expensive). Use of the teleport system does presume that you know the teleport number of your destination. We will need some sort of mechanism to distribute this information (a guidebook, perhaps).

Vehicles are a special class of container objects. When an avatar is seated in the driver's seat of a vehicle, the motion control commands that would ordinarily direct walking now direct the movement of the vehicle (and all its contents).

Busses are implemented as a kind of region whose connectivity with the rest of the world varies with time in an orderly way. Thus, you enter the bus region from a bus stop region, spend some time aboard the bus, then get off someplace else. To use the bus you have to pay bus fare, so the bus interior region contains a fare box object. You have to drop money in the fare box or you will be left behind when the bus leaves.

Communications

There are several ways of communicating with other players and with the system operators. These include broadcast and point-to-point talking, the mail system, and the telephone system. There is an analogy to the transportation systems in that all but the most elementary communications system (talking) require one or more objects to make them work. Several different kinds of objects are used to implement the communications facilities. A variety of other kinds of objects add features and capabilities to the basic systems for the convenience of the player.

The mail system uses the paper, pencil and mailbox objects. Messages are written on pieces of paper using a pencil (and need not be mailed, actually, but can be left around as notes or stored away for future reference, just like real paper). Each player has a mail address. You address a message to a player and then

drop it in a mailbox — either a public mail drop box or your personal household mailbox. Mail arrives for you in your personal household mailbox, from which you extract pieces of paper whose contents you can read.

The telephone system is much more sophisticated. The basic object is the telephone. There are two types of phones: home phones and pay phones. Each phone has a phone number, which you use to connect to it from another phone. Phones in **MicroCosm** work pretty much like they do in the real world. You pick up a phone, dial somebody's phone number, the phone rings at the other end and if they are home they answer and then you talk to each other. Everyone has a home phone on their turf which they can use free of charge. Pay phones are scattered throughout the world and may be used by anyone with a token. Embellishments include portable phones, which work like home phones except that you can carry them around with you; answering machines, which answer the phone when you are not home, deliver a message from you and collect a response; and beepers, which let you know when somebody is trying to call you.

Addressing

From the above discussion you should realize that, as in the real world, there are a number of parallel, partially redundant addressing schemes for referring to people and places in the world:

- Physical coordinates (region number + `(x,y)` within region)
- Residence address (region of your turf)
- Mail address
- Phone number
- Teleport number
- Name

Oddly, the only one of these whose function is unclear is the player's name.

The Objects

1. Pseudo-Objects

These are present because the player interface requires them but they are not really objects.

Object:
ground
Description:
The basic background below the horizon.
Function:
Can be walked on. Can be pointed at, returning a location.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Absolute blank, single color background from bottom of graphics window up to the horizon line.

Drawn by background rendering initialization routines.
Properties:
`color`, `horizon`
Notes:

Object:
sky
Description:
The basic background above the horizon.
Function:
Can be pointed at, returning a location.
Command Behavior:
Do:
Depends.
Go:
No effect.
Stop:
Cease.
Get:
No effect.
Put:
No effect.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Blue from top of graphics window to the scenic horizon. Scenic background color from scenic horizon down to the true horizon. Scenic background horizon line is a jaggy line procedurally determined from the terrain type. Drawn by the background rendering initialization routines.
Properties:
`color`
Notes:

2. Scenic Objects

These are always part of the background.

2.1. Inert Scenic Objects

These just sit there, though they may modify the effect or behavior of other objects.

Object:
bridge
Description:
Your basic small foot or highway bridge.
Function:
Provides pathway across water.
Command Behavior:
Do:
Depends.
Go:
Go to part of bridge indicated.
Stop:
Cease.

Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Two static images: side view and end view.
Properties:
`orientation`
Notes:

Object:
bush
Description:
Your basic bushy plant.
Function:
Scenic element. Obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image. Variations possible.
Properties:
`bush-type`
Notes:

Object:
fence
Description:
A section of an impassable, man-made barrier.
Function:
Linear obstruction.
Do:
Depends.
Go:
Goto.
Get:
No effect.

Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Two static images, side view and end view, of fence section. Variations possible (picket fence, wooden fence, chain-link fence, barbed wire fence, brick wall, stone wall, etc.).
Properties:
`path`, `fence-type`
Notes:

Object:
plant
Description:
Your basic generic plant.
Function:
Scenic element. Obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Uproot plant and get it into two hands, if possible.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image. Variations possible.
Properties:
`size`, `plant-type`
Notes:

Object:
pond
Description:
A small (within the region) body of water.
Function:
Water source. Water obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.

Get:
Go and fill.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Pond image on ground, rendered during background processing.
Properties:
`coverage`
Notes:

Object:
river
Description:
Body of water flowing through a region.
Function:
Water source. Linear water obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and fill.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
River image on ground, rendered during background processing.
Properties:
`path`
Notes:

Object:
sidewalk
Description:
Your basic suburban sidewalk.
Function:
Scenic element. Can be walked on (keeps feet off the lawn!).
Command Behavior:
Do:
Depends.
Go:
Go to nearest part of sidewalk if not already on it. Once on, follow along it to point designated.

Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Sidewalk image on ground, rendered during background processing.
Properties:
`path`
Notes:

Object:
street
Description:
Your basic roadway.
Function:
Can be walked on. Carries ground vehicles.
Command Behavior:
Do:
Depends
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Road image on ground, rendered during background processing.
Properties:
`path`
Notes:

Object:
tree
Description:
Your basic tree.
Function:
Scenic element. Obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.

Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image. Variations possible.
Properties:
`tree-type`
Notes:

2.2. Functional Scenic Objects

These can actually do something.

Object:
atm
Description:
Automatic **T**oken **M**achine.
Function:
Dispenses tokens.
Command Behavior:
Do:
Causes machine to tell you how many tokens are in your bank account.
Go:
Goto.
Stop:
Cease.
Get:
Takes 50 tokens or your entire balance, whichever is less, from your account and puts them in your avatar's hand.
Put:
Go if necessary. If item in hand is tokens, these are deposited in your account and disappear from hand, otherwise item is dropped in front of machine.
Talk:
Broadcast
Other Behavior:
None.
Graphics:
Single static image of machine. Sound effect of money being withdrawn. Sound effect of money being deposited.
Properties:
Notes:
We may wish to add some sort of dialogue that lets you select the amount you wish to withdraw.

Object:
fare box
Description:
Mass transit fare collection box.

Function:
Controls access to the bus system.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go if necessary. If item in hand is a token, drop in fare box, otherwise drop at foot of box.
Talk:
Broadcast.
Other Behavior:
The fare box is found in regions which represent the interior of busses. If you drop a token in the fare box you will be carried with the bus when it moves, otherwise you will be left behind.
Graphics:
Single static image of fare box. Sound of coin dropping in box.
Properties:
Notes:

Object:
fountain
Description:
Generic looking tacky town square fountain.
Function:
Scenic element. Water source. Oracle.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and fill.
Put:
Go and drop.
Talk:
Typed text message is sent to host as a question for the oracle. Also broadcast.
Other Behavior:
The oracle answers questions, makes predictions, and generally does all of the things that oracles are supposed to do. However, it is, by tradition, subtle and not completely reliable. Sometimes it answers your questions immediately. Sometimes it even carries on a conversation with you. Most of the time though, it takes quite a while to get an answer: you have to come back several days later. Occasionally it says things spontaneously. Such things are usually important. Anyone in the region with the oracle can hear what it says.
Graphics:
Single static image of fountain. Possible animation of water spurting up.
Properties:
Notes:
Of course, the minds behind the oracle are our own. Somebody has to respond, but the nature

of the oracle business is such that a response need not be timely, nor need all questions be answered.

Object:
gate
Description:
A gate in a fence or wall.
Function:
Provides a way to put a passageway through a fence.
Command Behavior:
Do:
If adjacent, open if closed (only openable if gate is unlocked or if avatar has the key), close if open. Otherwise, depends.
Go:
If adjacent, go through (opening and closing as needed), otherwise goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Two state image: open gate and closed gate. Variations possible due to fence/wall type.
Properties:
`open-or-closed`, `gate-type`, `locked-or-unlocked`, `key`
Notes:

Object:
mailbox
Description:
A conventional household roadside mailbox.
Function:
Interface to mail system.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. If there is mail for you, it is taken out of the mailbox and placed in your avatar's hand. Note that only the mailbox's owner can take mail out of it.
Put:
Go if necessary. If your avatar is carrying a letter (i.e., a piece of paper with and address and a message on it) it is taken from your avatar's hand and mailed away. If you are carrying anything else, it is dropped in front of the mailbox.
Talk:
Broadcast.

Other Behavior:
Mailbox has a little red flag on it that is up if mail is waiting and down otherwise.
Graphics:
Two state image of mailbox (flag up and flag down).
Properties:
`mail-arrived`
Notes:
A variant object would be a standard post-office style letter drop-box (like the one outside Z Building along Bellam), which would, of course, be a send-only device.

Object:
phone booth
Description:
The traditional walk-in phone booth with real-live working pay phone (which are getting rarer and rarer these days).
Function:
Interface to the telephone system for avatars away from home.
Command Behavior:
Do:
If in booth, hang up phone if not already hung up. Deactivates phone. Answer if ringing. Otherwise, depends.
Go:
If not adjacent to the phone booth, go to it. If adjacent, enter it. If inside, exit it.
Stop:
Cease.
Get:
Go if necessary. Answer the phone if it is ringing, otherwise do nothing.
Put:
Go if necessary. If avatar has a token in hand, activate the phone, otherwise drop whatever is in hand next to the phone booth.
Talk:
If the phone is activated (by dropping a token in it), interpret text as a phone call: first text message is phone number, further messages are conversation with the person at the other end. When you enter the number, the phone dials. If there is an answer, you can talk. If not, you get your token back (into avatar's hand) and the phone is deactivated. If the phone is not activated, broadcast.
Other Behavior:
The pay phone in the booth may be called from other phones. When this happens it rings. If you answer it you are connected to the other caller.
Graphics:
Single static image of phone booth. Sound of phone in booth ringing. Sound of phone ringing at other end of phone line. Busy signal sound. Dial tone. Dialing sounds (touch tones). Click when somebody answers.
Properties:
`phone-number`
Notes:

Object:
rock
Description:
Your basic rock.
Function:
Scenic element. If small can be picked up, thrown.

Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and pick up. (If not too heavy).
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Static image of rock. Possible variations depending on the size of the rock. Sound of rock hitting something.
Properties:
`weight`
Notes:

Object:
streetlamp
Description:
Conventional streetlamp.
Function:
Scenic element. Provides light at night.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
If it is night time, the region in which the streetlamp is found will be illuminated as if it were daytime.
Graphics:
Single static image of streetlamp. Possible stylistic variations (modern, old-fashioned, etc.).
Properties:
Notes:

3. Structural Objects

These are used to create buildings and other structures.

3.1. Component Structural Objects

These are the building blocks.

Object:
door
Description:
The conventional door.
Function:
Graphic element in buildings. Passageway through wall.
Command Behavior:
Do:
If adjacent, open if closed (only openable if door is unlocked or if avatar has key), close if open. Otherwise depends.
Go:
If adjacent, go through (opening and closing as needed), otherwise goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Two state image: open door and closed door. Variations possible due to style.
Properties:
`open-or-closed`, `door-type`, `locked-or-unlocked`, `key`
Notes:

Object:
roof
Description:
The roof of a building.
Function:
Graphic element in buildings.
Command Behavior:
Do:
Depends.
Go:
No effect.
Stop:
Cease.
Get:
No effect.
Put:
No effect.
Talk:
Broadcast.

Other Behavior:
None.
Graphics:
Roof image, rendered by background processor.
Properties:
`dimensions`, `roof-type`, `color`
Notes:

Object:
wall
Description:
An interior or exterior wall section.
Function:
Graphic element in buildings. Obstruction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Wall image, rendered by background processor.
Properties:
`color`, `dimensions`
Notes:

Object:
window
Description:
A conventional house window.
Function:
Graphic element in buildings.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.

Other Behavior:
None.
Graphics:
Single static image. Variations possible due to window style.
Properties:
`window-type`
Notes:

3.2. Composite Structural Objects

These are what hold the components together.

Object:
building
Description:
Any sort of building.
Function:
A ‘‘glue’’ object to organize building parts into a single entity.
Command Behavior:
Not a selectable object on the screen.
Other Behavior:
None.
Graphics:
The sum of its parts.
Properties:
`roof`, `walls`, `doors`, `windows`
Notes:
This is a higher-level data structure that organizes all of the building components into a single unit for reference purposes. It provides a base location for all the pieces and an organizing framework.

4. Furniture Objects

These are things found usually, but not always, *inside* buildings.

Object:
bed
Description:
An ordinary bed.
Function:
Can be laid upon (for lower energy expenditure?)
Command Behavior:
Do:
If adjacent, lay down. If laying down, get up. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.

Other Behavior:
None.
Graphics:
Three static images: side view, foot view and head view. Stylistic variations possible.
Properties:
`orientation`, `bed-type`
Notes:

Object:
chair
Description:
An ordinary chair.
Function:
Can be sat in (for lower energy expenditure?)
Command Behavior:
Do:
If adjacent, sit (if not occupied already). If sitting, stand. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Three static images: side view, front view, back view. Stylistic variations possible.
Properties:
`orientation`, `chair-type`
Notes:

Object:
chest
Description:
A chest of drawers.
Function:
Can hold stuff.
Command Behavior:
Do:
If adjacent and open, close. If adjacent and closed, open. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. If open, pick up contents (menu). If closed, no effect.
Put:
Go if necessary. If open, put contents of hands in drawers (if fits). If closed, drop.
Talk:
Broadcast.

Other Behavior:
None.
Graphics:
Two state image: open and closed chest.
Properties:
`open-or-closed`, `contents`
Notes:

Object:
couch
Description:
Your basic living room couch.
Function:
Like chair, but can be sat in by multiple avatars.
Command Behavior:
Do:
If adjacent and standing sit (if not full). If sitting, stand. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Three static images: side view, front view, back view. Possible stylistic variations.
Properties:
`orientation`, `couch-type`, `inhabitants`
Notes:

Object:
countertop
Description:
Store counter.
Function:
Can support things. Mediates transactions.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.

Other Behavior:
Objects placed on the counter behave differently. If you place an object on the counter, you can pick it up again. However, if somebody else places an object on the counter, you can only pick it up if you already have an item of your own on the counter. After you have picked up the other person's item, the ownership of the two items reverses.
Graphics:
Single static image of counter.
Properties:
`contents`
Notes:

Object:
display case
Description:
Store display case.
Function:
Can hold things visibly but safely, even if unattended.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
If you put an item in (or on) a display case, only you can pick it up again. Nobody else can.
Graphics:
Single static image of display case.
Properties:
`contents`
Notes:

Object:
floor lamp
Description:
A household floor lamp.
Function:
Provides light at night.
Command Behavior:
Do:
If adjacent and lamp is off, turn it on. If adjacent and lamp is on, turn it off. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.

Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
If it is night time and the lamp is turned on, the region in which the lamp is located will be illuminated as if it were daytime.
Graphics:
Single static image of floor lamp.
Properties:
`on-or-off`
Notes:

Object:
garbage can
Description:
Conventional garbage can or wastebasket.
Function:
Makes things disappear.
Command Behavior:
Do:
If adjacent, make contents of garbage can disappear (permanently). Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. Pick up contents of garbage can if there are any (menu).
Put:
Go if necessary. Put contents of hands into garbage can (if will fit).
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image of garbage can. Possible stylistic variations (garbage can, wastebasket).
Properties:
`contents`, `garbage-can-type`
Notes:

Object:
hot tub
Description:
A touch of Marin.
Function:
A place to hang out.
Command Behavior:
Do:
If adjacent and not in tub, climb in (if it's not full). If in, climb out. Otherwise, depends.
Go:
Goto.
Stop:
Cease.

Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image of hot tub.
Properties:
`inhabitants`
Notes:

Object:
sign/billboard/painting
Description:
A picture to display.
Function:
Displays artwork or text on wall or in space.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. Grab the item, if it is small enough.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
A single static image of a rectangular frame. Size is parameterized. Filled in with contents which is an image that is part of the object state.
Properties:
`size`, `picture`
Notes:
Due to the large overhead associated with such objects, we will guarantee that no picture will ever change. Rather, if a change is necessary a new picture will be substituted. This allows us to download the image once and save it, and then use it over time without worrying if it has changed when we were not looking.

Object:
table
Description:
A common table.
Function:
Can support things (hold them off the floor).
Command Behavior:

Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
A single static image of a table. Possible stylistic variations.
Properties:
`contents`, `table-type`
Notes:

5. Implement Objects

These are small, often handheld, things that can be manipulated to do something.

Object:
answering machine
Description:
A telephone answering machine.
Function:
Handle your phone calls when you're not at home.
Command Behavior:
Do:
If adjacent, plays back the next un-played-back message. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Typed text message becomes new answer message on machine.
Other Behavior:
The answering machine functions when it is placed at the same location as a private phone (i.e., not a pay phone). Whenever someone calls that phone, if it is not answered after the fourth ring the answering machine goes to work. The caller is sent the message that you have provided on the machine, and then has the opportunity to enter a response. The answering machine has a light on it that goes on whenever there are messages waiting.
Graphics:
Two state image: machine with light off and machine with light on.
Properties:
`waiting-messages`, `answer-message`
Notes:

Object:
beeper
Description:
One of those annoying little devices that disrupts movie theaters.
Function:
Alerts you that messages are awaiting.
Command Behavior:
Do:
If you are holding it and it is beeping, it shuts up; if it is not beeping, no effect. Otherwise depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
No effect.
Other Behavior:
If a phone call arrives at your home phone when you are not there, and you are carrying the beeper, it beeps until you either shut it up, go home and answer the phone, or the caller hangs up.
Graphics:
Single static image of beeper. Sound of beeper beeping.
Properties:
`beeping`
Notes:

Object:
book/newspaper/map
Description:
A readable document.
Function:
Displays text or artwork on paper.
Command Behavior:
Do:
If holding document, read it: fill graphics window with contents of document. If already reading it, stop. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
If holding, start reading. Otherwise, go and get.
Put:
If reading, stop reading. Otherwise, go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Single static image of document. Stylistic variations due to document type (book, magazine,

newspaper, flyer, etc.). Text or picture display of contents of document.

Properties:

`reading`, `text`, `picture`

Notes:

Due to the large overhead associated with such objects, we will guarantee that no document will ever change. Rather, if a change is necessary a new document will be substituted. This allows us to download the text or image once and save it, and then use it over time without worrying if it has changed when we were not looking.

Object:

bottle

Description:

A glass bottle.

Function:

Holds water, other liquids.

Command Behavior:

Do:

Depends.

Go:

Goto.

Stop:

Cease.

Get:

Go and get.

Put:

Go and drop.

Talk:

Broadcast.

Reversed Do:

Go if necessary. Pour the contents of the bottle on the indicated spot.

Other Behavior:

None.

Graphics:

Two state image: bottle empty and bottle full.

Properties:

`contents`, `capacity`

Notes:

Object:

credit card

Description:

Your basic plastic money.

Function:

Can be used at participating businesses to pay for purchases. Don’t leave home without it.

Command Behavior:

Do:

Depends.

Go:

Goto.

Stop:

Cease.

Get:

Go and get.

Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Purchase the indicated object with your card. If the indicated object is for sale and you have enought money in your bank account, then the indicated object becomes yours and the asking price for the object is deducted from your balance.
Other Behavior:
None.
Graphics:
Single static image of card.
Properties:
Notes:

Object:
compass
Description:
Your basic pointer to the West Pole.
Function:
Tells absolute direction.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
The image chosen to display the compass in any given region reflects the orientation of that region's viewpoint with respect to true west. The compass appears as an arrow pointing towards whichever edge of the region is west.
Graphics:
Four state image of compass with arrow: pointing left, pointing right, pointing towards viewpoint, pointing away from viewpoint.
Properties:
`orientation`
Notes:
This may not be needed.

Object:
die
Description:
A six-sided die.
Function:
For gambling, games.

Command Behavior:

Do:

If holding the die, roll it. It is thrown to the ground and displayed in a new state. Otherwise, depends.

Go:

Goto.

Stop:

Cease.

Get:

Go and get.

Put:

Go and drop.

Talk:

Broadcast.

Reversed Do:

Throw.

Other Behavior:

None.

Graphics:

6-state image showing the various possible die rolls.

Properties:

`state`

Notes:

This probably won't work.

Object:

escape device

Description:

Your basic panic button.

Function:

Gets you out of a jam, fast.

Command Behavior:

Do:

If holding the device, teleport your and your possessions back to your home turf. Otherwise, depends.

Go:

Goto.

Stop:

Cease.

Get:

Go and get.

Put:

Go and drop.

Talk:

Broadcast.

Reversed Do:

Throw.

Other Behavior:

None.

Graphics:

Single static image of device. Sound effect when device is activated.

Properties:

Notes:

Very rare and expensive.

Object:
flashlight
Description:
Your basic hand torch.
Function:
Portable light source at night.
Command Behavior:
Do:
If holding, turn on if off, off if on. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
If it is night time and the flashlight is turned on, the region will be illuminated as if it were daytime.
Graphics:
Single static image of flashlight.
Properties:
`on-or-off`
Notes:
We may wish to consider requiring batteries.

Object:
gemstone
Description:
Like a rock, only worth more.
Function:
Valuable. May contain magic.
Command Behavior:
Do:
If magic, activate magical function. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.

Graphics:
Single static image of gemstone. Stylistic variations possible due to stone type.
Properties:
`magic`, `gemstone-type`
Notes:
Magical function may vary. Set of magical functions needs thought.

Object:
handcuffs
Description:
Police-style wrist-or-ankle-cuffs.
Function:
Restrain another avatar's hands or feet.
Command Behavior:
Do:
If adjacent, unlock cuffs (if you have the key). Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
If restrained by them, no effect. Otherwise, go and get.
Put:
If restrained by them, no effect. Otherwise, go and drop.
Talk:
Broadcast.
Reversed Do:
If restrained by them, no effect. If adjacent to object and object is an avatar, bind the specified site (wrists or angles) with the cuffs. Otherwise, throw.
Other Behavior:
When wrists are restrained by cuffs, certain **do** actions are not possible. When ankles are restrained, walking is not possible.
Graphics:
Single static image of cuffs. Sound of cuffs clicking shut, clanking open.
Properties:
`binding`, `key`
Notes:
The effect of wrist restraint needs further thought.

Object:
key
Description:
An unpickable lock key.
Function:
Opens locked doors, handcuffs.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.

Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
If object is lock for which this key fits, unlock or lock the lock. If object is lock for which key does not fit, no effect. Otherwise, throw.
Other Behavior:
None.
Graphics:
Single static image of key. Sound of key turning in lock.
Properties:
`lock`
Notes:

Object:
knick knack
Description:
Your basic gewgaw.
Function:
Scenic element. Possibly magical.
Command Behavior:
Do:
If magic, activate magical function. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of knick-knack. Stylistic variations possible.
Properties:
`magic`, `knick-knack-type`
Notes:
Set of gewgaws needs thought.

Object:
paper
Description:
A piece of paper.
Function:
Can be written upon and then retrieved.
Command Behavior:
Do:
If holding, read. If reading, hold. Otherwise, depends.

Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
If holding paper and also holding pencil, text message typed is written onto paper for future reference. Otherwise, no effect.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of piece of paper. Large format display of text on page.
Properties:
`text`
Notes:
Any piece of paper whose text begins ‘‘to *name*’’ can be sent as a mail message.

Object:
pencil
Description:
Your basic pencil.
Function:
Writes on paper.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
If holding both pencil and piece of paper, text message entered is written onto the paper.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of pencil.
Properties:
Notes:
It’s a magic pencil: it never needs sharpening.

Object:
property changer

Description:
Miscellaneous gadget.
Function:
Changes some property of another object or avatar.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Activate property changing function against target object.
Other Behavior:
None.
Graphics:
Single static image of gadget. Stylistic variations possible. Sound of gadget working.
Properties:
`effect`, `property-changer-type`
Notes:
The set of possible property changes needs lots of thought.

Object:
security device
Description:
Another miscellaneous gadget.
Function:
Provides a way to make a region safe from intrusion.
Command Behavior:
Do:
If device is in hand, turn it off if it's on, or on if it's off. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
When device is on, an indicator light is lit. When the device is activated no one else can enter the region it is in.
Graphics:
Two-state image of device: on and off. Also, sound when it is turned on or off.

Properties:
`on-or-off`
Notes:
Only works in selected regions (like hotel rooms).

Object:
sensor
Description:
Another miscellaneous gadget.
Function:
Tells some property of a region, object or avatar.
Command Behavior:
Do:
If sensor is in hand, turn it off if it's on, or on if it's off. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Direct attention of sensor towards target object.
Other Behavior:
While sensor is on, it periodically changes its sound or appearance according to what it is sensing.
Graphics:
Multi-state image of sensor: sensor off, sensor on but not sensing anything, sensor sensing something, etc. Also, sound of detector in action.
Properties:
`sensor-type`, `detects`
Notes:
The set of things that sensors sense needs further thought.

Object:
telephone
Description:
The household telephone.
Function:
Remote communications with other players on-line.
Command Behavior:
Do:
If adjacent, hang up if not already hung up, answer if ringing. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. Answer the phone if it is ringing, otherwise get.
Put:
Hang up if holding and not hung up. Otherwise, go and drop.

Talk:
If adjacent, interpret text as a phone call: first text message is phone number, further messages are conversation with the person at the other end. When you enter the number, the phone dials. If there is an answer, you can talk. If not adjacent, broadcast.
Reversed Do:
Throw.
Other Behavior:
The phone may be called from other phones. When this happens it rings. If you answer it you are connected to the other caller.
Graphics:
Single static image of phone. Sound of phone ringing. Sound of phone ringing at other end of phone line. Busy signal sound.
Properties:
`phone-number`
Notes:

Object:
towel
Description:
Your basic towel.
Function:
Necessary for every traveler.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of towel.
Properties:
Notes:
Everyone knows that this is the single most useful object you can have.

Object:
walkie-talkie
Description:
A radio telephone.
Function:
Like a telephone, but portable.
Command Behavior:
Do:
If adjacent, hang up if not already hung up, answer if ringing. Otherwise, depends.

Go:
Goto.
Stop:
Cease.
Get:
Go if necessary. Answer if it is ringing, otherwise get.
Put:
Hang up if holding and not hung up. Otherwise, go and drop.
Talk:
If holding, interpret text as a phone call: first text message is phone number, further messages are conversation with the person at the other end. When you enter the number, it is dialed. If there is an answer, you can talk. If not holding, broadcast.
Reversed Do:
Throw.

Other Behavior:
The walkie-talkie may be called from other phones. When this happens it rings. If you answer it you are connected to the other caller.

Graphics:
Single static image of walkie-talkie. Sound of phone ringing. Sound of phone ringing at other end of phone line. Busy signal sound.

Properties:
`phone-number`

Notes:

6. Substance Objects

These are things which are reckoned in terms of their quantity.

Object:
drugs

Description:
Little pills.

Function:
Temporarily changes an avatar's properties.

Command Behavior:

Do:
If holding, take one of the pills. Drug immediately effects relevant property. Otherwise depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.

Other Behavior:
When quantity reaches zero, it's all gone and disappears.

Graphics:
Single static image of pill.

Properties:
`count`, `effect`

Notes:
The set of possible effects needs thought.

Object:
tokens
Description:
Filthy lucre.
Function:
Money in the MicroCosm.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw. Possible menu interaction to determine quantity given.
Other Behavior:
None.
Graphics:
Single static image of a coin. On one side it says *Good For One Fare*. On the other side it says*F iat Lucre*. Sound of coins jingling.
Properties:
`quantity`
Notes:
Handling of varying of quantities of loose change needs further thought.

Object:
water
Description:
You know what water is.
Function:
A useful fluid.
Command Behavior:
Must be in a container. Manipulated indirectly.
Other Behavior:
None.
Graphics:
None.
Properties:
`quantity`
Notes:
What good is it?

7. Clothing Objects

Clothing makes the avatar, as they say.

Object:
diving suit(?)
Description:
Wetsuit and scuba gear.
Function:
Lets an avatar travel through water.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
If an avatar is wearing this, he can travel as if walking through ponds and rivers.
Graphics:
Single static image of suit in a heap. Additional avatar animation frames showing avatar in suit.
Properties:
Notes:
Is this really a good idea?

Object:
hat
Description:
Your basic hat.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.

Graphics:
Single static image of hat. Possible stylistic variations. Additional avatar animation frames of hat on head.
Properties:
`hat-type`
Notes:

Object:
jacket
Description:
Your basic jacket.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear or unpocket.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of jacket in heap. Additional avatar animation frames of torso wearing jacket.
Properties:
`contents`, `capacity`
Notes:

Object:
pants
Description:
A pair of pants.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear or unpocket.
Put:
Remove.

Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of pants in heap. Additional avatar animation frames of legs wearing pants. Possible stylistic variations.
Properties:
`pants-type`, `contents`, `capacity`
Notes:

Object:
shirt
Description:
Your basic shirt.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of shirt in heap. Additional avatar animation frames of torso wearing shirt. Possible stylistic variations.
Properties:
`shirt-type`
Notes:

Object:
shoes
Description:
A pair of shoes.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.

Stop:
Cease.
Get:
Wear.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of shoes. Additional avatar animation frames of feet wearing shoes. Possible stylistic variations.
Properties:
`shoe-type`
Notes:

Object:
skirt
Description:
Your basic skirt.
Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear or unpocket.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of skirt in heap. Additional avatar animation frames of legs wearing skirt. Possible stylistic variations.
Properties:
`skirt-type`, `contents`, `capacity`
Notes:

Object:
swim suit
Description:
Swim trunks.

Function:
Decorative. Helps distinguish one avatar from another.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of swim suit in heap. Additional avatar animation frames of legs wearing swim suit.
Properties:
Notes:

8. Container Objects

These are objects which can hold other objects.

Object:
backpack
Description:
Your basic backpack.
Function:
Can carry multiple items without use of hands to hold them all.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Wear or unpocket.
Put:
Remove.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static frame of lone backpack. Possible additional animation frames of pack on torso of avatar.

Properties:
`contents`, `capacity`
Notes:

Object:
bag
Description:
The sack.
Function:
Can carry multiple items using only one hand.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
If holding, get stuff out of it. Otherwise, go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Two state image of bag: full and empty.
Properties:
`contents`, `capacity`
Notes:

Object:
box
Description:
Your basic box.
Function:
Can carry more than two items using only two hands.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
If holding, get stuff out of it. Otherwise, go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.

Other Behavior:
None.
Graphics:
Two state image of box: full and empty. Possible stylistic variations (cardboard box, wooden crate, treasure chest).
Properties:
`contents`, `capacity`
Notes:
We may want to consider making boxes open/close -able.

Object:
cage
Description:
A cage with bars and everything.
Function:
Can contain an avatar (and hold him against his will).
Command Behavior:
Do:
If adjacent or inside, unlock cage (if you have the key). Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
None.
Graphics:
Two state image of cage: open (unlocked) and closed (locked).
Properties:
`locked`, `contents`
Notes:

9. Weapon Objects

These are things which can adversely effect other avatars and objects.

Object:
club
Description:
The most basic weapon.
Function:
Pain and injury at close range.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.

Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
If the target is adjacent, hit it with the club. Otherwise, no effect.
Other Behavior:
None.
Graphics:
Single static image of club.
Properties:
Notes:
We need to consider what the damage effects will be.

Object:
grenade
Description:
A little bomb that goes boom.
Function:
Death and mayhem tossed into another region.
Command Behavior:
Do:
If holding, pulls pin on grenade. Several seconds later it will explode. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw. Grenade can be tossed into an adjoining region.
Other Behavior:
If the pin is pulled, the timer is set. The timer counts down each second. When it reaches zero the grenade explodes, doing damage to avatars and objects in the region it exploded in.
Graphics:
Two state image of grenade: static state and pin pulled. Animation of explosion. Sound of pin being pulled. Sound of explosion.
Properties:
`timer`, `pin-pulled`
Notes:
We need to consider what the damage effects will be. We need to think about how you indicate that you wish to toss it into another region.

Object:
gun
Description:
A pistol.

Function:
Death and destruction from a distance.
Command Behavior:
Do:
If holding, toggles safety switch. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Shoot at designated object. In order to shoot, the safety must be off.
Other Behavior:
None.
Graphics:
Two state image: static gun and gun shooting. Sound of shot.
Properties:
`safety-setting`
Notes:
We need to consider what the damage effects will be. We should decide if requiring ammunition would be a good idea.

Object:
knife
Description:
Sharp pointy thing.
Function:
Death and injury at close range.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
If the target is adjacent, stab it with the knife. Otherwise, no effect.
Other Behavior:
None.
Graphics:
Single static image of knife.
Properties:

Notes:
We need to consider what the damage effects will be.

10. Vehicle Objects

These take things from place to place.

Object:
boat
Description:
A nice little runabout.
Function:
Carries avatars and their possessions across water.
Command Behavior:
Do:
Depends.
Go:
If adjacent, get in. Otherwise, goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
The boat can travel across water. An avatar operates it by sitting at the controls and then executing movement commands as if he were walking.
Graphics:
Three static images of boat: side view, front view, back view. Sound of boat driving.
Properties:
`contents`, `capacity`, `speed`
Notes:

Object:
bus
Description:
Mass transit.
Function:
Carries many avatars and their possessions along roads.
Command Behavior:
A bus is just a special kind of region. It is not a selectable object on the screen.
Other Behavior:
A bus is a special region whose connection to other regions changes with time. Busses follow predetermined paths. You board a bus by entering its region when it is at an adjacent bus stop. You get off by leaving its region when it is stopped. It costs money to ride the bus. Bus fare is collected by a fare box object — each bus region contains one.
Graphics:
Background image of bus from the outside. Background image of the bus from the inside. These are rendered by the background processor.
Properties:
`route`
Notes:
A mechanism for collecting the fare is still needed.

Object:
car
Description:
The great American automobile.
Function:
Carries avatars and their possessions along roads.
Command Behavior:
Do:
If adjacent to the car or inside it, opens the car door if it is closed or closes it if it is open. Otherwise, depends.
Go:
If adjacent and door is open, get in. If inside and door is open, get out. Otherwise, goto.
Stop:
Cease.
Get:
No effect.
Put:
Go and drop.
Talk:
Broadcast.
Other Behavior:
The car can travel along roads (only). An avatar operates it by sitting in the driver's seat and then executing movement commands as if he were walking.
Graphics:
Three static images of car: side view, front view, back view. Sound of car driving.
Properties:
`contents`, `capacity`, `speed`
Notes:

Object:
teleport booth
Description:
Like a phone booth, but it carries all of you instead of just your voice.
Function:
Zaps avatars and their possessions elsewhere instantaneously.
Command Behavior:
Do:
Depends.
Go:
If not adjacent to the booth, go to it. If adjacent, enter it. If inside, exit it.
Stop:
Cease.
Get:
Go and get.
Put:
Go if necessary. If avatar has a token in hand, activate the teleporter, otherwise drop whatever is in hand next to the phone booth.
Talk:
If the teleporter is activated (by dropping a token in it), interpret text as a teleport booth number, and teleport the avatar and everything he is carrying to the teleport booth dialed. If the other teleport booth already has an avatar in it, you get a busy signal and then you get your token back (into avatar's hand) and the booth is deactivated. If the booth is not activated, broadcast.
Other Behavior:
The booth may be called from other booths. When this happens an avatar materializes from

someplace else, unless the booth is occupied already.

Graphics:

Single static image of teleport booth. Sound of busy signal. Sound of yourself zapping from one place to another. Sound of somebody else zapping in from elsewhere.

Properties:

`booth-number`

Notes:

Object:

truck

Description:

A big truck.

Function:

Carries lots of objects along roads.

Command Behavior:

Do:

Depends.

Go:

If adjacent, get in. If inside, get out. Otherwise, goto.

Stop:

Cease.

Get:

No effect.

Put:

Go into truck and drop.

Talk:

Broadcast.

Other Behavior:

The truck can travel along roads (only). An avatar operates it by sitting in the driver's seat and then executing movement commands as if he were walking.

Graphics:

Three static images of truck: side view, front view, back view. Sound of truck driving.

Properties:

`contents`, `capacity`, `speed`

Notes:

11. Magical Objects

These can do any number of things...

Object:
amulet
Description:
A hang-around-the neck type amulet.
Function:
Generic magic talisman.
Command Behavior:
Do:
If holding or wearing, activate magical function, if there is one. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of amulet. Stylistic variations possible.
Properties:
`effect`, `amulet-type`
Notes:
Magical functions need to be more carefully considered.

Object:
instant object pill
Description:
A little pill, until you use it.
Function:
Add water, it turns into some object.
Command Behavior:
Do:
Depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.

Other Behavior:
If you pour water on it, it turns into something else.
Graphics:
Single static image of pill. Sound of metamorphosis.
Properties:
`instant-what`
Notes:

Object:
magic lamp
Description:
Just like Aladdin had...
Function:
Rub it and a genie appears to grant a wish.
Command Behavior:
Do:
A genie appears to grant you a wish.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of lamp. Sound of genie appearing.
Properties:
Notes:
Need to define a genie object! Genie himself works like the oracle.

Object:
magic staff
Description:
A stick about as tall as an avatar.
Function:
Generic magic talisman.
Command Behavior:
Do:
If holding, activate magical function, if there is one. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.

Talk:
Broadcast.
Reversed Do:
If target is adjacent, hit with staff, unless magical function is projectile in nature, in which case activate the magical function.
Other Behavior:
None.
Graphics:
Single static image of staff. Sound of staff operating.
Properties:
`effect`, `staff-type`
Notes:
Magical functions need to be figured out.

Object:
magic wand
Description:
Just like your fairy godmother has.
Function:
Generic magic talisman.
Command Behavior:
Do:
If holding, activate magical function, if there is one. Otherwise, depends.
Go:
Goto.
Stop:
Cease.
Get:
Go and get.
Put:
Go and drop.
Talk:
Broadcast.
Reversed Do:
If target is adjacent, hit with wand, unless magical function is projectile in nature, in which case activate the magical function.
Other Behavior:
None.
Graphics:
Single static image of wand. Sound of wand operating.
Properties:
`effect`, `wand-type`
Notes:
Need to think out magical functions.

Object:
ring
Description:
Your basic magic ring.
Function:
Generic magic talisman.
Command Behavior:
Do:
If wearing, activate magical function, if there is one. Otherwise, depends.

Go:
Goto.
Stop:
Cease.
Get:
If not wearing, put it on. Otherwise, go and get.
Put:
If wearing, take it off. Otherwise, go and drop.
Talk:
Broadcast.
Reversed Do:
Throw.
Other Behavior:
None.
Graphics:
Single static image of ring. Sound of ring operating (maybe).
Properties:
`effect`, `ring-type`
Notes:
Magical functions need more definition.

12. Avatar Objects

These are the people themselves.

Object:
avatar
Description:
You or somebody else.
Function:
The animated figure.
Command Behavior:
Do:
If adjacent, strike the avatar with your fists. If self, ??. Otherwise, depends.
Go:
If self, change posture: standing, sitting, or lying. Otherwise, goto.
Stop:
Cease.
Get:
If not self, go if necessary and receive an object from the other avatar. If self, ??.
Put:
If not self, go if necessary and give whatever is in your avatar's hands to the other avatar. If self ??.
Talk:
If not self, text message is sent to other avatar as a person-to-person message. If self ??.
Other Behavior:
None.
Graphics:
Complex set of animation images expressing a wide range of motions and actions.
Properties:
`position`, `orientation`, `weight`, `health`
Notes:
There are undoubtedly lots of properties not touched on here. What to various actions taken with respect to one's self mean? We need to define the animation very rigorously.