

# Dreamstudio (Import) FAQ

by MGreen

Updated to v0.4 on Jan 9, 2002

This is the Dream Stud!o Giga-FAQ (well, eventually) v0.4

By Mark Green

Email address for submissions/comments:

mark [at] antelope [dot] demon [dot] co [dot] uk

The latest version of this FAQ will always be available from

<http://www.gamefaqs.com>.

-----  
wwwwwwwwwwwwww If the block of characters on the left has a straight  
mmmmmmmmmmmmmm right-hand edge, you are using a monospaced font.  
iiiiiiiiiiiiiii This FAQ looks far better viewed in a monospaced font.  
OoOoOoOoOoOo Use EDIT, DOS TYPE, the Netscape text viewer, or MORE  
12345678901 to view this file.  
-----

-----  
\*\*\*\*\* INDEX \*\*\*\*\*  
-----

## INTRODUCTION

### QUESTIONS THAT ACTUALLY HAVE BEEN FREQUENTLY ASKED

- Q1 Is Dream Studio coming out in America/Europe?
- Q2 Where can I get Dream Studio from?
- Q3 Can you use Dream Studio if you don't know Japanese?
- Q4 Can you use Dream Studio to make a game in English?
- Q5 Does Dream Studio work with the Dreamcast CDX?
- Q6 Does Dream Studio work with American/European DC Keyboards?
- Q7 What Battle system/Magic system does Dream Studio use?
- Q8 Can you transfer the games you make to the PC?
- Q9 Can somebody who doesn't have Dream Studio play the games I make?
- Q10 Can I stop other people loading my games into the editor?
- Q11 Can I make a Zelda or Mario game with Dream Studio?
- Q12 Have any games actually been made with Dream Studio?
- Q13 Is it true that there are bugs in Dream Studio?
- Q14 How can I upload/download games from the Internet?

## TUTORIAL SECTION

- T1 Setting up and getting a game started
- T2 Beginning Editing
- T3 Editing Different Maps
- T4 Moving between maps - your first Event

## REFERENCE SECTION

- R1 General Notes
  - R1.1 Editor modes
  - R1.2 Dialog structure
  - R1.3 Text entry
  - R1.4 Safety lines
  - R1.5 Start Menu
  - R1.6 Game properties
  - R1.7 HSV Colour
  - R1.8 File types
- R2 Opening/Setup
  - R2.1 Create A Game

- R2.2 Play A Game
- R2.3 Options
- R2.4 Internet
- R3 World Editor
  - R3.1 Basic operation
  - R3.2 Objects
    - R3.2.1 Creating Objects
    - R3.2.2 Editing Objects
    - R3.2.3 Object Properties
    - R3.2.4 Action Properties
    - R3.2.5 Ambient behaviour properties
  - R3.3 Maps
- R4 Scenario Editor
  - R4.1 Book
  - R4.2 Sheet
    - R4.2.1 Condition Editor
    - R4.2.2 Spark control
  - R4.3 Script
    - R4.3.1 Script editor basics
    - R4.3.2 Event list
      - R4.3.2.1 Messages track events
        - R4.3.2.1.1 Message
        - R4.3.2.1.2 Menu
        - R4.3.2.1.3 String Entry
        - R4.3.2.1.4 Button
        - R4.3.2.1.5 Timed Button
        - R4.3.2.1.6 Probability
      - R4.3.2.2 Hero track events
        - R4.3.2.2.1 Serif
        - R4.3.2.2.2 Action
        - R4.3.2.2.3 Emotion
        - R4.3.2.2.4 Move
        - R4.3.2.2.5 Fly
        - R4.3.2.2.6 Turn
        - R4.3.2.2.7 Colour
        - R4.3.2.2.8 Visibility
        - R4.3.2.2.9 Model
        - R4.3.2.2.10 Carrying
        - R4.3.2.2.11 Switches
        - R4.3.2.2.12 Map move
      - R4.3.2.3 Object track events
      - R4.3.2.4 Camera track event
      - R4.3.2.5 BGM track events
        - R4.3.2.5.1 Start new
        - R4.3.2.5.2 Stop all
        - R4.3.2.5.3 Adjust
      - R4.3.2.6 Sound FX track events
        - R4.3.2.6.1 Effect
        - R4.3.2.6.2 Stop all
      - R4.3.2.7 Letterbox track events
        - R4.3.2.7.1 Blackout
        - R4.3.2.7.2 Letterbox
      - R4.3.2.8 Map track events
        - R4.3.2.8.1 Sky
        - R4.3.2.8.2 Lighting
        - R4.3.2.8.3 Spotlights
        - R4.3.2.8.4 Fog
        - R4.3.2.8.5 Colour
        - R4.3.2.8.6 Switches
      - R4.3.2.9 Special text track events

- R4.3.2.9.1 Special text
- R4.3.2.9.2 Narrative
- R4.3.2.10 Vibration track event

TECHNIQUES SECTION

- X1 Multiple sheets firing in one event
- X2 Hit Points (or counters in general)

LIBRARY INDEXES

- L1 Model Library
- L2 SFX Library
- L3 Sample game library
- L4 Map Library
- L5 BGM library
- L6 Human action library

CREDITS

-----  
\*\*\*\*\* LEGAL BITS \*\*\*\*\*  
-----

Neither the author of this FAQ, nor any person who distributes it in any way, shall be responsible or liable for anything that results from using this FAQ for any purpose, including but not limited to damage to your Dreamcast, controllers, hands, eyes, or sanity (which may be oddly likely with this title).

This FAQ may be freely distributed provided that it is kept unmodified and in its entirety. This FAQ may not be sold, or included as part of a publication that is sold, without the author's express permission.

Dream Studio is (c) and (tm) Sega Enterprises, 2000. Zelda and Mario are trademarks of Nintendo. Final Fantasy is a trademark of Square/Sony. ShenMue and very possibly QTE are trademarks of Sega Enterprises. The use of any trademarks within this FAQ is not intended to represent a challenge to their validity.

-----  
\*\*\*\*\* INTRODUCTION \*\*\*\*\*  
-----

Dream Studio is an 3D adventure creation tool for the Dreamcast. It allows the creation of fully 3D interactive adventure-style games. Although it does not have hard-wired systems for magic or combat, these can be added using the condition and scripting language included within it.

Dream Studio is an intensely complex piece of software, and this FAQ attempts to unravel it. This first version of the FAQ is extremely incomplete, but I have posted it so people can get an initial flavor of what's being found out. It will be updated as I find out more.

The FAQ is divided into sections:

TUTORIAL SECTION - A walkthrough-style introduction to Dream Studio.

REFERENCE SECTION - A reference to each part of the tool.

LIBRARY INDICES - List of all items available in the libraries on the GDROM.

-----  
\*\*\*\*\* QUESTIONS THAT ACTUALLY HAVE BEEN FREQUENTLY ASKED \*\*\*\*\*  
-----

\$ Q1 IS DREAM STUDIO COMING OUT IN AMERICA/EUROPE?

Sega has made no statement on this as yet.

\$ Q2 WHERE CAN I GET DREAM STUDIO FROM?

Relatively few importers have carried Dream Studio because of its vertical market interest and heavy Japanese content. Two importers I know have actively listed it: In the USA, NCSX (<http://www.ncsx.com/>) and in the UK, Another World (<http://www.anotherworld.co.uk/>).

\$ Q3 CAN YOU USE DREAM STUDIO IF YOU DON'T KNOW JAPANESE?

When this FAQ is finished, I hope so. :)

\$ Q4 CAN YOU USE DREAM STUDIO TO MAKE A GAME IN ENGLISH?

Yes. The "Romaji" alphabet is the Japanese name for the English alphabet, so you can use this for all your dialogs and messages. The only pieces of text you cannot change are the options on the pause menu, and the error that appears if a string entry prompt is used and the user types a value that isn't recognised. You may find some dialog boxes are a little small for English messages, though, because Japanese is more compact than English in general (because of Kanji).

\$ Q5 DOES DREAM STUDIO WORK WITH THE DREAMCAST CDX?

Yes.

\$ Q6 DOES DREAM STUDIO WORK WITH AMERICAN/EUROPEAN DC KEYBOARDS?

Yes, although the shift and caps lock keys may produce Japanese characters as well as English capitals.

\$ Q7 WHAT BATTLE SYSTEM/MAGIC SYSTEM DOES DREAM STUDIO USE?

It doesn't have one wired in. You can make one yourself by using the Sheet system although it can be quite a lot of work. Many Japanese authors are writing battles using the QTE system from ShenMue; it is also possible to do FF7 menu-driven battles although it will take a lot of memory.

\$ Q8 CAN YOU TRANSFER THE GAMES YOU MAKE TO THE PC?

If you have an HKEMS memory card you can transfer files to the PC and upload and download them, but you can't play Dream Studio games on the PC.

\$ Q9 CAN SOMEBODY WHO DOESN'T HAVE DREAM STUDIO PLAY THE GAMES I MAKE?

No. There is, AFAIK, no runtime version of Dream Studio. All your players must have full copies of Dream Studio. Although it's possible that Sega could stick a runtime version on a magazine coverdisc...

\$ Q10 CAN I STOP OTHER PEOPLE LOADING MY GAMES INTO THE EDITOR?

No. There's no facility for doing this.

\$ Q11 CAN I MAKE A ZELDA OR MARIO GAME WITH DREAM STUDIO?

Yes and no. The problem is assigning actions to buttons. You can do this by using Events, but the response will probably be too slow

for attacks or jumps in one of these types of game. If you're prepared to put up with it, by all means go ahead.

\$ Q12 HAVE ANY GAMES ACTUALLY BEEN MADE WITH DREAM STUDIO?

Yes! Go to <http://www.kuusou.co.jp/dstudio/dendou.htm> for a list of games. However, all the games there are in Japanese.

\$ Q13 IS IT TRUE THAT THERE ARE BUGS IN DREAM STUDIO?

Unfortunately, Kuusou have reported that there are bugs in Dream Studio. Fortunately, they can be easily avoided once you know what they are. They are as follows:

- Dream Studio may freeze if you use Play From Cursor in an long event that includes Letterbox or Blackout commands.
- A "ghost" hero object may appear on some maps. This hero object can be recognised because the object manipulation menu for it will \*SCROLL\*, which no other object manipulation menu will. If one of these does appear, do NOT scroll the menu and use the other options, as they will confuse the engine terribly.

\$ Q14 HOW DO I UPLOAD/DOWNLOAD GAMES FROM THE INTERNET?

You will need one of these things: an HKEMS (Nexus/Interact) memory card and a PC, \*OR\* a Dreamcast that can connect to the net in your region.

If you have the latter, you can just head off to the game site and download the file directory. The drawback is you have to worry about memory card slots. Slots? Yes. VMSs use an internal file structure and Dream Studio files are named D\_STUDIO.G01, D\_STUDIO.G02, etc. The problem is, if you have a game stored as D\_STUDIO.G01 and you try and load a game that somebody else had stored as D\_STUDIO.G01, one of them will get overwritten. So be careful when downloading files, and save your own games in higher-numbered slots.

If you have the former, things are a bit trickier. HKEMS memory cards take data in .DCI format, which isn't provided on most DreamStudio sites. So here's what to do. Find the game you want. Click on it to download it. You should get a .VMI file. Now, go to the link you downloaded the game from, press the right mouse button, and use Copy Link Location (or Explorer equivalent) to get the path of the VMI file on the server. Paste this into the location bar, and change the file extension to .VMS, then hit return to download the .VMS file (which is the actual game).

Now, you'll need a copy of a utility called DCI\_VMI, which can be gotten from any good VMS programming page. Run it on the .VMS file and answer the questions as follows:

Data or game file?	DATA
Filename for .DCI?	Whatever you like
DCI file copyable?	YES
Filename for VMS directory?	D_STUDIO.G01
VMI Program Description?	Whatever you like
VMI Copyright Description?	Whatever you like

The advantage of this method is: you can change D\_STUDIO.G01 to CHOOSE which memory card slot the game goes in. Make sure you only change the number after the G, though. Pick a filename not used by any of your games at the moment.

Finally, use WCPY1 (which should have come with the card; if it didn't there should be a link to it in the documentation that did come with it) to download the .DCI file to your VMU, and you're set.

\$ T1 Setting up and getting a game started

Open the GDRAM case. (Open the left-hand half. For some reason, Sega decided to put the disc in the side where it's not supported in the case when the case is the right way up..) Hold the disc between thumb and forefinger. Press the open button on your.. Ok, you know all this stuff, right? Ok, just meet me at the DS Main Menu. :)

Right. Hit the start button, then choose the first option (Create A Game), then the second option (From the Beginning). Yep, we're going to dive right in with this tutorial - most of the explanatory stuff included on the disc is in Japanese and it'd take longer for me to explain it in a seperate text file than it will for us to just wade in. :)

A big screen of Japanese text pops up. Press A to get rid of it, then A again to dismiss the next box (which is asking you to enter your name), and then.. well, enter your name. Pull the left trigger twice to get the Romaji (English) alphabet and use the joystick and A button to enter your name. Go and have a quick peek at section R1.3 for all the controls on the text entry box - really do, because we're going to be using it a lot.

Once you've entered your name, hit START then A to quickly end the text entry (you COULD have moved the cursor manually onto the bottom of the menu and hit A, but this way's much quicker) and then A again to dismiss the box (which is asking you to enter the title of your game), then back to the text editor to enter the title. Call it whatever you like at the moment.

You're now at another screen full of Japanese with a menu selection at the bottom. Choose the right hand option. The system is asking you to choose Beginner or Expert mode - and you're choosing Expert mode. Why? Because all Beginner mode does is to remove options from the menus that might be confusing. However, unless you're fluent in Japanese, taking a few options away WON'T make the menus much less confusing! ;) Also, if you're in Expert mode from the start, you won't have to relearn all the menu locations.

And finally, you'll come to the model menu, to choose the model for the hero of the game. Pick the first model offered, the standard hero. Or, just to be PC, you can press Right on the digital joystick to get the standard heroine. Or you can go wild and scroll up and down the menus until you find a model you like - but it's important that you make it human. (You can actually select a non-human object as the hero, but it'll cause some things later on to not do what I'm expecting.) Once you've found a model, select it with the A button.

And finally! After that long, long trip, you're finally in the World Editor, looking at your hero standing in a lovely green glade.

\$ T2 Beginning Editing

Right. Now we're in the Editor, let's get down and make some stuff.

Before we do, though, let's just take a quick look over the basic items you'll be editing and how they're structured.

Dream Studio is divided into two halves: the World Editor and the Scenario Editor. The World editor lets you edit the environment in which your game will be taking place. The Scenario editor lets you specify the events that actually take place in the game.

To break things down a bit further, here's what you can edit with the World Editor: Maps, which are physical locations that make up the game, and Objects, which are things that will be encountered in those locations. So, a Map might be the Town of Arelin, the Evil Wizard's Tower, or the Dungeon of Destruction. An Object might be the town's armor shop, the evil wizard's treasure chest, or the hero him/herself.

Now the Scenario Editor edits three things: the Book, which describes the order in which things in the game take place; Sheets, which describe when things happen, and what must take place in order for story events to happen; and Scripts, which define what happens on screen when an event in the plot occurs.

So, suppose you've got to go into the Wizard's Tower to retrieve his treasure chest and give the treasure to the king to claim the Royal Wand of Power. The Wizard's Tower is a map, and so is the king's castle. The treasure chest is an object - and so are the Royal Wand of Power, the king, the hero, and the wizard. The Book says that once you've gotten the treasure chest, the king will give you the Wand when you meet him. One Sheet says that if you go to a certain place in the wizard's tower, you get the treasure chest, and another says that if you go to a certain place in the palace, you meet the king. And one Script shows the hero triumphantly raising the chest above his head; another shows the King handing over the Wand, and maybe there are others too that show the king dismissing the hero for arriving without the treasure, or the battle between the hero and the wizard's pet dragon.

Ok, digression over. Let's do stuff. Use the digital stick to pan the viewpoint of the initial map around. Let's make this map a town from which the hero can start. We have the land, we have the hero, we just need some buildings. Use the analogue stick to move the cursor away from the hero and scenery, then hit A. You should go straight to the Model List for creating a new object. (If you get a submenu, choose the bottom - blue - option, which is Create Object. You were near an object or objects at the time you pressed A, and the submenu was listing those objects in case you wanted to edit one. You don't, you want to create one, so use the bottom option.)

Scroll three categories to the right, and pick the first building. The Object Properties sheet will pop up. Hit A on the very first item - this changes the name of the object - and change the name to "HOUSE" or similar. Then, START-A to get out of the text editor, and START-A again to get off Object Properties. You're now back in the World Editor with the house glued to your cursor. Move around with the analogue and stamp down two houses with the A button. Use the X key and the triggers to rotate the houses in place.

Once you're done, hit B to get the house off the cursor. You should now be on the object list submenu again, so choose the bottom

option to Create Object (if you \*aren't\* on the submenu, hit A to get the object submenu (if you're near any objects) or the Model List (if you're not)). This time, choose the fourth building model down. Rename it "SHOP" on Object Properties, and then put one down near the houses.

Ok, time to see that this looks like. Hit START and choose the top option (Playtest), then the top option again from the submenu (From This Map). This will take you into the game. Run about a bit and look at the houses. Note you can't get into any of the houses at the moment. Press START and choose the second option to return to the editor.

Notice that even though you have returned to the editor, the hero is still in the place you left him while you were playing. The editor shows the \*current\* state of the world at all times. To reset the game, press START again, and choose the third option (Game Reset), then the bottom option (All).. and you're reset.

### \$ T3 Editing different maps

-----

We'd probably like the hero to be able to go into some of these houses. We can't just move the hero inside the house object; every area the hero explores has to be represented as a separate map, so we need a map for every house we want the Hero to be able to enter.

Press the Y key to pop up the Map Menu and choose the bottom (blue) option to create a new Map. (The Map Menu lists all the maps you've defined and the one other option was the name of the map we were editing just now.) The Map library will pop up. Press right to select interior maps and choose the top map option - this is a nice standard house map. The map properties page will appear - dismiss it immediately with START-A.

Now you're in the editor for the house. It's quite a nice house - it's come prefurnished from the Map Library with a mat, a bunk bed, and a desk. Note also that the hero's standing in the middle of it. How'd he get there? Well, in the editor, there will always be an image of the hero on the map you're editing - this enables you to playtest the map. In the actual GAME, though, how the hero gets between maps can be a bit more involved. We'll be looking at that in a moment.

Let's put something in the house. Something adventury. How about a sword? You know the object creation drill by now: hit A, choose the bottom option on the submenu if you get one, and pick from the library. There's a nice sword about five categories left (assuming the last thing you placed was a house). Pick "sword 1". Change the name in Object Properties to "SWORD" and START-A out. Now carefully place the sword on the wall. Make sure to use the digital to ensure you view all sides of the sword - you don't want it floating in front of the wall! If you need to move the sword up and down, hold X and use the analogue stick.

So, now there's a sword there. But we still can't pick it up. Don't worry.. first, let's deal with getting our hero into the house!



## \$ T4 Moving between maps

-----

Hit the Y button, select the first map we edited (the top option) and then select the top option again (Go To Map). It's time for us to link these maps together so that the hero can enter a house by moving up to one. And for this.. we need to use the scenario editor. Yes, anything that moves the hero between maps is considered to be an Event, and hence we need to add one to the scenario.

The Scenario Editor is more abstract than the World Editor and can be a lot more intimidating first time you see it. Don't worry too much, it's not as complex as it looks.

Hit START, and choose the second option (Switch Editor). You'll enter the Scenario Editor. You'll start up in the Book mode. Remember the Digression in T2? The Book defines the order and flow of events that happen in the game. At the moment, there are only two Sheets shown in the Book - the S sheet (the start of the game) and the E sheet (the end of the game). They're also not connected in any way. In other words, we know the game starts, and we know it can end, but we don't know how it could get between them.

You can use either joystick you like for this bit. Move to the E box, hold A, and drag it down a bit - we're going to create a new Sheet, so need some more space. Now point to any empty area and hit A. A new sheet will spring into existence - with a central icon marked with an A. A sheet represents an event in the game. In this case, we want it to represent the hero going into the house. At the moment, the sheet has default settings, which are no use at all (because they don't do anything) - we need to edit it. Press A on the sheet, and then choose the second option down (Edit Sheet).

Ok, we're in the Sheet Editor. First and foremost, move up to the very top, and change the sheet name to something like "ENTERHOUSE" (you only get 10 characters). Now let's take a look at what makes up the sheet. There's a condition, a script, inputs, and outputs. The inputs and outputs are the groups of four circles at the top and bottom respectively of the big box on the screen (those are the circles you also saw in the Book). The condition is the left-hand box that's INSIDE the big box (near the A-in-a-circle icon), and the script is the right-hand one (with the bar graph like display inside). The big box on the right is for your comment and has no effect on the sheet.

Move the cursor until it's on the text area next to the A-in-a-circle icon and hit A (Edit Condition). The icon used for a sheet tells you what Condition it's using, and the A-in-a-circle icon means it isn't using one at all, so it triggers as soon as an appropriate point in the game plot is reached (we'll see how that's determined in a moment), but we don't want that; the player has to actually do something - try and get into the house - to trigger this event. When you pressed A, the list of potential triggers appeared. Choose the second one - touch map area. (There is one for "touch object", but if we used that and said the object was the house, the hero could crash into the back wall of the house and wind up going in - the door isn't a separate object!)

You're now sent back to the World with a red, circular area highlighted. That area is the area the hero must enter for the event to take place. Find the house you want to use as the hero's house and move the area so that the centre of it is inside the door and a semicircle sticks out

at the front - close enough that if the player moved into it they'd definitely want to go in the door. You can change the size of the circle by holding X and pushing left and right, but you shouldn't need to - the default circle is OK for size. Once you have it placed, press A.

We now have a sheet that fires when the player approaches the door. But at the moment it won't work. Why? Because as well as the player doing something, the sheet also needs to be ready - it needs to be an appropriate point in the plot for the event to happen.

Press the B button on the Sheet Editor to go back to the Book Editor. It's on the Book Editor that you specify how the plot flows. Now, when can you get into this house? Well, we want the player to always be able to get into it. In other words, it's appropriate as soon as the game starts. The "S" sheet represents the game starting. Put the pointer on the circular "handle" at the bottom of the S sheet, hold down A, and DRAG to the left-hand "handle" at the top of the sheet we just created. There'll be a weird gunshot sound, and the two will link together. What does this mean? It means that, once the game's started, the player can enter the house. If we linked another sheet to the bottom of the "ENTERHOUSE" sheet, it would mean that something else could happen once the player had entered the house. If we linked the E sheet to the bottom of the "ENTERHOUSE" sheet, it would end the game when the player entered the house (don't actually do this).

Ok, so we now have a thing called "ENTERHOUSE" which is available from the start and which happens when the player walks up to the door. But what does it DO? Point at the ENTERHOUSE sheet again, press A, and this time choose the TOP option (Edit Script). You're now in the script editor, where you specify what's happening in an Event. We'll be looking at this in more detail later. For now, move the cursor to the big button on the left, press A and choose the second option (Hero Track). A track representing the actions that the Hero in the event will appear. Point to the very beginning of the track (just beyond the first red guideline), press A, and choose the bottom option (move between maps).

When the requester appears, hit A to set the destination. The Map List will appear again; move down to the house map, select it and press A. You'll now be placing the hero as if you were placing an object (you are setting the point in the new map where the hero will arrive). Ignore the copy of the hero already in the room (that's the playtesting one); place the new hero by the door, facing so it looks like he's just come through it, and press A. When the requester reappears, press START, A again. Notice the big white block that's now appeared on the hero track.

Press start and choose the second menu option (Switch Editor) to return to the World Editor. Now reset the game as described in T2 and playtest again. (You need to reset the game, because you added an event that was available when the game was started. Problem is, because you'd playtested before, the event didn't actually EXIST when the game was started and so isn't properly marked available - resetting fixes this out.) Run over to the house you put the event on. As you enter the region you placed around the door, your hero will fade out (an effect of the "move between maps" event) and appear in the house!

Ok, we're in the house. Now we want to be able to get back out of it

again. We do that in the same way - via another sheet that fires when touching the door's area. This time, however, the area has to be part of the map we just made of the house. Quit playtesting and return to the scenario editor, Book Mode. In the Scenario editor, you can only change the map while in Book Mode. Press Y, and choose your house map. Now, the easiest way to deal with this is going to be to use a copy of our original sheet. Point to your ENTERHOUSE sheet, press A, and choose the fourth option down (copy to another map), then move to the side and stamp down the new copy. Notice that the linkage of the old sheet is copied as well.

Open the sheet editor on the new copy (A button, second option). Change the name of the new sheet to "EXIT HOUSE" (or similar), and go to the Condition editor (text box on the left of the icon). When the menu of Conditions appears, just press A to keep the same condition. You will now see a hitzone floating in midair. What's happened? Well, because you copied the sheet from the original outside map, it's kept the same absolute location for the hit zone. However, since it's no longer on the outside map, but the house map, that location no longer makes any sense. Move the hit zone onto the door out of the house in the same way as you originally placed it on the one inwards. To get quickly to the Script Editor from the Sheet Editor, you can point to the RIGHT-hand box, above the mock-up script editor. Point to the white block (the Move Between Maps event), press A, and choose the top option (edit properties). Press A on the default choice (destination), pick the outside map, and then place the hero outside the house. Leave the Sheet Editor and return to Book Mode.

Now, switch back to the outside map, reset the game, and start playtesting again. Walk up to the door, and you'll go into the house. Walk around, then go to the door again, and you'll leave. Now, you might find at this point that when you arrive in the house, you can't move away from the door without leaving the house. That's because the area that your "ENTERHOUSE" sheet places the hero in, is inside the hit zone for the "EXIT HOUSE" sheet! Enter the script editor for whichever sheet is causing the problem, and move the hero slightly away from the door.

Go back to playtesting (if you left). Go into the house, then out, then try going back into the house. And.. gasp.. nothing happens. Why?

Go back to the Scenario Editor and look at the Book again. Your Book shows that the hero can enter the house after the game has started. But a connection in the Book only lets an event happen ONCE. So once the hero has entered the house, the event has happened, and it can't happen again until the game is started again. So, how do we make the hero able to leave and re-enter the house at will? Quite easy. Point to the leftmost of the four output circles on the bottom of the "ENTERHOUSE" sheet, press and hold A, and drag the line up to the SECOND leftmost of the four circles on the TOP of the SAME "ENTERHOUSE" sheet - you're making a loop. Connections to different inputs are separate; as soon as any of the four connections is satisfied, the event becomes ready to go. So what you've said is that the player can enter the house after the game is started, OR the player can enter the house AFTER THEY HAVE ENTERED THE HOUSE (!) Since one of these is always true, the player will always be able to enter the house.

Now, you might be at this point thinking this sounds a bit daft - surely it would be logical to say the hero can't enter the house until after he has \*LEFT\* the house, or that he can't leave until he's entered. Well,

you CAN link the sheets together in that way if you want - but there is no need to, since it is impossible simply due to the maps for the hero to enter the house twice in a row without leaving it. We don't need the Book to enforce this restriction, since it's already enforced. Also, in playtesting we \*might\* need to enter the house twice without leaving it (we can do this, because we can jump between maps however we like), so we allow the possibility, although a non-editing player will never be able to do it.

Do the same thing with the "EXITHOUSE" sheet - link its leftmost output to its leftmost input. Reset the game, and playtest again. You should now be able to move freely in and out of the house.

Just for a finishing touch, open the Script Editor for "ENTERHOUSE" again. Hit the large button again, and this time choose the sixth option down - sound effects. Point to the start of this track and press A to create an event. Choose the first of the two event types, and then when offered the menu, choose the default option. Press down 45 times and you'll have an effect of a door opening. Press A to select it, then START-A to get off the properties menu for the sound effect. Now you should find the door sounds like a door when entering the house. You can do the same thing for "EXITHOUSE" too if you like. (Of course in normal circumstances we'd have done this before we copied the sheet, to avoid doing it twice, but I didn't want to digress in the middle of a lesson. :) )

#### \$ T5 Picking up objects

-----

Ok, we have that lovely tempting sword hanging in the house, but no way to pick it up - yet. So, let's deal with that. First of all, you need to know that there's no inventory in Dream Studio. To pick up the sword, all we need to do is to make an Event that shows the hero picking up the sword. If we need to make other things that depend on possession of the sword, we have to use the linkage system in the Book to make them dependant.

First, let's deal with making the event. Go to Book Mode in the Scenario Editor, change the Map to the house map, and create a new sheet. Link it to the start sheet, and then open its sheet editor. Point to the condition box. This time, the sword on the wall is an entirely seperate object - so we'll use the Touch Object condition type, which is the fourth one down. Choose that, and then move the selector box over the sword and press A. There will probably actually be several items inside the selector box (since the house is so small), so a list of objects will appear - pick the sword (if you didn't rename it, the name is a single kanji followed by the number 1). Ok, now we have to deal with the Script, so head to the Script editor.

The Script here is going to be a little more complex. All we NEED to do to show the hero picking up the sword is to use an event to put it into his hand, but we might like to make a little more happen, so let's.

The first track we'll need is the Camera track. Press the Big Button and then move down to the Camera option (the fourth one down) and hit A. We need this because, since we used a Touch Object trigger, the hero might be touching the sword from any old angle, and the camera might not be ideally placed to show

him picking it up. Point to the start of the Camera track, and press A to create an event. There's only one potential event on the Camera track, so pick that one, and when the properties menu appears, press A on the default selection. Choose the middle option (View Absolute) and then the top option (place by View), then use the joysticks to move until you have a good view of the sword on the wall that isn't too zoomed in. Tilt it back and to the side a little as well, though - remember we don't actually want to see the sword, we want to see the hero when he's in front of it. We'll just line the camera up on the sword because we know that's where the hero will be.

Once the camera's placed, push A to select the location, and then START-A to exit the camera configuration. The camera movement now appears as a purple bar on the camera track.

Next thing we need is a Messages track, so that the hero can say how delighted he is to be picking up his sword. Usual drill: hit the Big Button, choose Messages (the top option), then create a message event by pointing and clicking on the Messages track directly AFTER the Camera movement block. Choose the first type of event (a regular message). When the properties list appears, press A on the only option, and type in a suitably cheesy message for the hero to say. START-A the text entry, then START-A the event properties.

Now, last but not least, actually picking up the sword. For this, we need two tracks: an Object track for the sword, and a Hero track. Add them both with the Big Button: Hero's the second option down, and Object's the third. When you choose Object, you'll have to select the sword in the same way that you did when you chose it for the condition.

Point to the Hero track at a point AFTER the message event on the message track, press A, and choose the tenth option down (Carrying). Move the cursor left and right to select the contents of the hero's hands - put the sword in his right hand. Note that this is purely a visual effect - you aren't obliged to do this.

START-A to complete that event, then enter an event on the Sword's Object track at the same time. This time you want the eighth option down - visibility. We have to turn the sword's visibility off when the hero picks it up, or it would continue to be on the wall after it had been picked up. (We can't just delete it, because you can't add or delete options from edit events, so we'll make it invisible as a make-do.) Set the visibility setting to "OFF", then start-A out. Note that the event is quite long - this gives the sword a slow fade out of visibility, which is NOT what we want. Grab the handle on the right hand end of the event, and drag it to the left until the event is as small as it can be (same size as the Carrying event).

Now, reset and playtest and the hero should be able to pick the sword up off the wall. (You might have to adjust the position of the sword if it's hard for the hero to make enough contact with it for him to be considered touching it.)

-----  
\*\*\*\*\* REFERENCE SECTION \*\*\*\*\*  
-----

-----  
\$ R1 General Notes  
-----

\$ R1.1 Editor modes  
-----

The Editor can operate in one of two modes: beginner or expert mode. You are asked to choose which you want when you create a game and you can change your mind while editing by going to the Game Properties screen and changing the setting there.

Beginner mode and Expert mode are more or less the same except that in Beginner mode certain options will be disabled or ghosted out. Most of the menu translations listed here apply to Expert mode.

\$ R1.2 Dialog structure  
-----

In diagrams of Dialog boxes and menus in this FAQ, the following notation has been used:

_____	Text entry area; A button opens text entry (R1.3)
_____ v	Menu selection shown as a string; A button opens menu
!__Rx.x__!	Dialog choice: A button opens dialog referred to by indicated section of FAQ
< _____ >	Inline selection: use A to change to move
<ON> <OFF>	Inline toggle; A toggles, value shown is default
.....	Safety line (R1.4)
-[]-----	Slider; left and right sets value
[ ]	Command button; A button activates

On all dialogues, the B button will cancel your changes, and the START button will move the cursor immediately to the OK button.

Most Dialogues in Dream Studio are divided into two parts: a Standard section and an Advanced section. You can fold down the Advanced section of any dialog by hitting the button in the lower right-hand corner. ALL dialogues will then display their Advanced section until you press the lower right-hand button on any dialog again.

The lower-right hand button will be ghosted and give an error sound if you select it while the editor is in Beginner Mode.

In Dialog translations shown here, the Advanced part of the dialog will be shown separated from the basic dialog by a line looking like this:

}===== {

\$ R1.3 Text entry  
-----

Many parts of Dream Studio require you to enter text. The text entry system is as follows.

Use the digital joypad to move around and select letters from the pad, and choose menu items from the menu on the right. The A button selects letters and options. Letters are added at the point of the cursor; the epsilon sign indicates the end of the string and cannot be moved. The other buttons perform the following functions:

L Switch displayed character set (Hiragana/Katakana/Romaji/Dingbats)

X End of word (see below)  
B Delete last character  
R Insert carriage return (if appropriate)  
Y Hold Y and move digital joystick left and right to move cursor

The menu is as follows:

Switch character set (same as L trigger)  
End of word (see below - same as X)  
Insert carriage return (if appropriate) (same as R)  
Delete last character (same as B)  
Delete current character  
Cancel  
Ok

Pressing START will immediately jump the cursor to the OK button on the menu.

Note that when you're editing a Script, the Text Entry box will often be preceded by a list of all strings already in the game (comments, object names, other messages, etc). You can select one of these to reuse it. This will help make your game smaller, since the game doesn't have to store the same string twice.

If you are typing in Romaji (English), X inserts a space. If you are typing in Japanese, then the last word you have entered will be highlighted in yellow, and hitting the X button will cause the program to replace the highlighted yellow kana with appropriate kanji or symbols. You can select the kanji or symbol by hitting the B button repeatedly. Enter another character to lock the selected kanji into place. If you don't want to enter another character, lock the kanji into place by pressing START.

The character sets are as follows:

Hiragana

a i u e o ha hi fu he ho ga gi gu ge go  
ka ki ku ke ko ma mi mu me mo za ji zu ze zo  
sa shi su se so ya yu yo da ji zu de do  
ta chi tsu te to ra ri ru re ro ba bi bu be bo  
na ni nu ne no wa xx xx wo n pa pi pe pe po  
\*a \*i \*u \*e \*o \*tsu \*ya \*yu \*yo \*wa ? ! - , .

Katakana (more or less the same)

Romaji

A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z  
a b c d e f g h i j k l m  
n o p q r s t u v w x y z  
0 1 2 3 4 5 6 7 8 9 @ ; :  
+/- + - \* / = /= < > <= >= , .

#### \$ R1.4 Safety lines

-----  
The term "safety line" will be used several times in this FAQ. A safety line is a line of hyphens appearing on a menu as an option. It will do nothing when selected other than produce an error beep. The only

reason for it is to separate dangerous menu options (like Delete) from the rest of the menu.

#### \$ R1.5 Start Menu

-----  
Hitting the Start Button anywhere in the editor will give you this menu:

-----  
>Playtest game >  
>Switch editor >(World Editor to Scenario Editor or back)  
>Reset game >  
>Game Properties>  
>Exit to title >  
>File menu >  
-----

PLAYTEST GAME: Takes you into the playtesting mode to try out your game. If you are in Expert Mode, you will get a submenu:

-----  
>From this map >  
>From this sheet>  
>From the cursor>  
-----

"From this map" starts the game from the current state of the World Editor, with no events in progress.

"From this sheet" can only be used if you pressed start from a Sheet in the Scenario Editor, and causes the sheet to trigger immediately.

"From the cursor" can only be used if you pressed start from a Script in the Scenario Editor, and causes the script to run from the place where the pointer was. In Beginner mode, "From here" is selected automatically.

During playtesting, pressing Start gives:

-----  
>Play On >  
>Return to Editor>  
-----

You cannot save or load games while playtesting. If you "Return to Editor" while in the middle of an event, a warning is shown that this may leave the game in a weird state if you return from the editor again. Choose the left option to edit anyway, right option to return to pause menu.

Note that changes made to the map in the course of play WILL be reflected in the editor. For example, if you start the game and move the hero a bit to the left, he will remain there when the editor is restarted.

GAME RESET: Restores the game back to its initial state, undoing any changes made by playtesting (or by the "Alter Current Position" option in Expert Mode). In Expert mode, you get a submenu:

-----  
>Reset everything except the map>  
>Reset everything >  
-----

"Reset everything except the map" will reset all your sheets and stati but not the map. "Reset everything" will reset the map objects as well. "Reset everything" is the default if you are in Beginner



mode.

GAME PROPERTIES: Game Properties will take you to the Game Properties screen (R1.6).

EXIT TO TITLE: Returns you to the main menu of Dream Studio, asking if you're sure first.

FILE: Gives a submenu:

```
----  
>Save<  
>Load<  
----
```

Which is self explanatory.

\$ R1.6 Game properties

-----

```
-----  
>Game Title          _____>  
>Author's name      _____>  
>Hero                ! _____R3.2.3_____!>  
>Hero's initial location {map name} >  
>                    x: ***  y: *** >  
>Editor mode        < _____>>  
>[OK]                >  
-----
```

GAME TITLE: The name of your game.

AUTHOR'S NAME: Your name.

HERO: Takes you to the Object Properties sheet for the hero. The text shown is the Hero's name.

HERO'S INITIAL PLACEMENT: Where the hero starts the game. The World Editor will be started to allow you to specify the location.

EDITOR MODE: Beginner or expert mode; see R1.1.

\$ R1.7 HSV Colour

-----

Dream Studio uses the HSV system for determining colours. Under this system, a colour is made by mixing a pure light colour with a base grey, rather than mixing together red, green and blue. Each colour is defined by the following properties:

HUE: The actual base colour involved in the mix. Varies based on the position of the colour in the EM spectrum: reds are low, blues are high.

SATURATION: How much of the base colour is mixed in with the base gray. 100% is pure base colour. 0% is pure grey.

VALUE: How bright the base grey is.

\$ R1.8 File types

-----  
Dream Studio manipulates two types of files: actual games that you have written, and files that represent a point of progress within a game. The latter type are called "saved games" by most games, but using this terminology in Dream Studio is not appropriate, since it could be confused with a game that you have written and saved.

Hence, Dream Studio and this FAQ refer to a file containing an actual game definition as a "saved game", and a file containing a point of progress within a game as a "saved PLAY".

-----  
\$ R2 Opening/Setup  
-----

The Opening Menu is as follows:

-----  
>Create A Game<  
>Play A Game >  
>Options >  
>Internet >  
-----

\$ R2.1 Create A Game  
-----

After selecting the Create A Game option, a menu will appear:

-----  
>Edit an existing game<  
>Make a new game >  
-----

If you choose "Edit existing game", you will go to the file management screen to load the game you want to edit.

If you choose "Make a new game", a series of prompts will appear:

- First, the game asks you to confirm that you really want to make a new game from scratch. Press A to confirm or any other button to back out.
- You're then asked for your name.
- You're then asked for the title of your game.
- You're then asked to select the initial mode of the editor: the left option is beginner mode, the right option is expert mode.
- Finally, you're asked to choose the initial model for the hero object from the model library. Once you've done so, you'll pop straight into the World Editor. (Yes, you can choose \*any\* object from the library, so if you want to write "The adventures of a Dungeon T-Piece", go ahead.)

\$ R2.2 Play A Game  
-----

After selecting to Play A Game, a menu will appear:

-----  
>Load a saved play<

>Play a new game >

-----

If you choose "Load a saved play", you will go to the file management screen to load your play. Note that the GD-ROM disc will not be available as saved plays can't go there.

If you choose "Play a new game", you will go to the file management screen anyway, to load the game file that you want to play. The GD-ROM will be available.

Once you have chosen, you will begin to play the game as normal. The analogue joystick controls your character and the digital one moves the camera. The effect of other buttons is dependant upon the game design. Pressing START will lead you to a menu:

-----

>Play on >

>Save play>

>Load play>

>Main menu>

-----

### \$ R2.3 Options

-----

There are only two options:

-----

>Sound <\_\_\_\_\_>>

>Vibration <\_\_\_\_\_>>

-----

Sound is Mono if the second character is a single line, Stereo otherwise. Vibration is On if the string is 4 characters long, Off if it is 5.

### \$ R2.4 Internet

-----

I don't know the details of this bit.

### \$ R3 World Editor

-----

#### \$ R3.1 Basic operation

-----

The World Editor is used to place objects within your game world for the player to interact with.

The controls for the World Editor are:

Digital Stick	Move camera
Analogue Stick	Move object cursor
L	Zoom in
R	Zoom out
A	Open object selection menu (see below)
B	Cancel (does nothing by default)
X	Movement axis shift (allows the analogue stick to move the object cursor in the Y axis (up and down) rather than in the X and Z axes (along the ground))

The top of the screen will show "energy bars" indicating the amount of memory you have used overall and how near you are to exceeding the object limit for the map.

Note: When a map is first loaded, the Dreamcast loads the objects one at a time. The map will slowly fill itself in as the objects are loaded, and the Map limit bar will slowly rise. If you start Playtesting while the Map is still loading, the "Now Loading" box will appear until it's caught up.

Most of the controls are self explanatory. The object selection menu shows all objects in range of the cursor. Use the joystick to select an object to edit - the cursor will jump onto the selected object as you do so. The bottom option (in a different colour) will create a new object. If there are no objects in range of the cursor when you press A, you will automatically create a new object.

## \$ R3.2 Objects

### \$ R3.2.1 Creating Objects

To create an object, hit the A button when there is no object near the cursor, or hit the A button and choose the bottom option from the menu of objects.

When you create an object, you will first be asked to choose the model for the object (L1). You will then go to the Object Properties page (R3.2.3). When you finish there, you will return to the menu with the object highlighted in the cursor. You can then place your object on the map by moving the cursor and hitting A to actually put the object in place. If you want to place more than one, continue moving the cursor and hitting A to create copies. When you're finished placing, press B. While placing an object, you can rotate the object by holding down X and pressing the triggers.

### \$ R3.2.2 Editing Objects

To edit an object, hit the A button when near that object and then select that object's name from the list of objects that appears.

The edit menu will then appear:

```

-----
>Object Properties  >
>Object Position   >
>Current Position  > (Expert mode only)
>Copy              > (See below)
>Copy to another map> (See below)
>.....>
>Delete           > (See below)
-----

```

Note that the Copy and Delete options do NOT appear if the object you selected is the hero.

OBJECT PROPERTIES will open the Object Properties sheet (R3.2.3).

OBJECT POSITION will allow you to place the object again as you did when creating it (R3.2.1).

CURRENT POSITION will allow you to move the object, except that the position isn't retained after a reset or saved in the game; you will move the object just as if it had been moved by the player or by a script in

the game.

COPY lets you place a copy of the object.

COPY TO ANOTHER MAP is self explanatory.

DELETE will get rid of the object.

### \$ R3.2.3 Object Properties

The object properties sheet is as follows:

```
-----  
∕Object name      _____ |          Picture of          ∕  
∕Object model    ! ___ L1 ___ ! |          Object          ∕  
∕Mood            _____ v |      Use L and R to      ∕  
∕Scheme          _____ v |      rotate            ∕  
∕Behaviour       _____ v -----∕  
∕Left Arm        _____ v      Right Arm _____ v∕  
∕[OK]                                     [ADVANCED]∕  
}===== {  
∕Visibility      <ON>      Animation  <ON>      Pose Lock <OFF>∕  
∕Colour          _____ v ∕  
-----
```

OBJECT NAME: This is the name the object will appear with in menus and the editors. By default this is the same as the name of the model, but you'll probably want to change it to something in English.

OBJECT MODEL: The model used for the object.

MOOD: May be ghosted if inapplicable for the model. The mood of the character represented by the object. Choose from normal, angry, happy, etc. They are clearly pictured on the screen so there is no real need for translation.

SCHEME: May be ghosted if inapplicable for the model. Pick one of up to four colour schemes for the object.

BEHAVIOUR: Gives you a submenu:

```
-----  
∕Action          ∕(takes you to Action Properties R3.2.4)  
∕Stand still∕(all three other options take you to Ambient  
∕Roam           ∕ Behaviour Properties R3.2.5)  
∕Follow         ∕  
-----
```

Ghosted if the object is the hero; you CAN make the hero do fixed behaviours, but only in scripted events.

LEFT ARM, RIGHT ARM: May be ghosted if inapplicable for the model. Allows you to choose what item the character is holding in its left or right hand. Choose from either "items" section of the Model Library.

VISIBILITY: Controls if the object will be visible or not. If you set this OFF, the object won't be visible until a script makes it become so. This is necessary to create effects, since scripts cannot add objects to the scene; you must place the objects in advance and then the script can make them visible. This option will be ghosted if the object is the hero.

ANIMATION: Controls if the object's ambient animation (based on its Action) will run or not.

POSE LOCK: Controls if the object's turning and other behaviour will be automated. If OFF, the object will turn to face the player when an event triggers, turn the way it's moving, etc. If ON, it will remain constantly facing the same way and in the position you specify unless it's

altered by a script.

COLOUR: The base colour of the object.

#### \$ R3.2.4 Action Properties

-----  
This window is used to control the action taken by an object. It is as follows:

```
-----  
>Action select _____ v >  
># repeats      -[]----- >(max left is infinite)  
>Reverse ani    <OFF>      >  
>[OK]           [ADVANCED]>  
)===== {  
>Speed          -[]----- >  
>Return <ON> Equipment<ON>>  
>Face <ON> Sound <ON>>  
>Freeze pose    <OFF>      >  
>Frame select   -[]----- >  
-----
```

ACTION SELECT: Choose an Action from the Action Library (L5). If the model isn't human the set of actions might be different.

# REPEATS: How many times the action should be repeated. The maximum left value repeats infinitely.

REVERSE ANI: If ON, the animation is played backwards.

SPEED: Sets the speed of the animation.

RETURN: My approximate translation is "Completion Return", but I haven't tested what this does yet.

EQUIPMENT: My approximate translation is "Equipment Display", but I haven't tested what this does yet.

FACE: My approximate translation is "Facial Expression", but I haven't tested what this does yet.

SOUND: My approximate translation is "Sound Effects", but I haven't tested what this does yet.

FREEZE POSE: If selected, the animation will be frozen.

FRAME SELECT: Select which frame from the animation is frozen. Use these controls to create original static poses by pinching them from action animations.

#### \$ R3.2.5 Ambient behaviour properties

```
-----  
>Behaviour      _____ v >  
>[OK]           [ADVANCED]>  
)===== {  
>Action         !_R3.2.4_v!>  
>Speed          -[]----- >  
>[Area]         Details of >  
>               area      >  
>Move type      _____ v>  
>Leader         _____ v>  
-----
```

BEHAVIOUR: Choose one of these three:

```
-----  
>Stand still >  
>Roam >
```

>Follow >

-----

(Note: I'm not sure what the difference is between having "Stand Still" selected here (and hence in R3.2.3) with an Action selected, and having "Action" selected in R3.2.3).

Stand Still keeps the character still. Roam causes the object to wander around randomly inside the assigned area. Follow causes the object to follow another whenever that other object is within the assigned area.

ACTION: Takes you to Action Properties R3.2.4 to set the action the object performs during the behaviour.

SPEED: How fast the behaviour is performed.

AREA: Takes you to the World Editor to specify a circular area in which the behaviour is appropriate. Use the X button and the triggers to change the diameter of the area. If you leave the diameter at 0, as it is by default, the behaviour will apply to the entire map.

MOVE TYPE: Two options:

-----

>On the ground >

>Through the air>

-----

Ghosted if Behaviour is Stand Still.

LEADER: Choose the object that will be followed. Ghosted unless Behaviour is Follow.

### \$ R3.3 Maps

-----

Hitting Y in the World editor brings up the list of maps. The current map is shown in Red; the lowest option, marked with squares, creates a new map.

If you select the current map, the following menu appears:

-----

>Map properties >

>Copy map >

>.....>

>Delete all objects>

-----

If you select a map that is not current, you get the following menu:

-----

>Make map current >

>Make map current + edit properties>

>Copy map >

>.....>

>Delete map >

-----

Most of these are self explanatory. The map properties sheet is as follows:

```

-----
>Name          _____>
>BGM          !__L5__!>
>Sky          !__L6__!>
>Lighting     !R3.3.1!>
>[OK]         [ADVANCED]>
}=====
>Animations    <ON> >
>Sounds        <ON> >
>Colour        ____v >
>Fogging      !R3.3.2! >
>Spotlight 1 !R3.3.3! >
>Spotlight 2 !R3.3.3! >
>Spotlight 3 !R3.3.3! >
-----

```

Name: Name the map. That's all. By default it's named after the Model map it was created from, but you probably want to rename it in English.

BGM: Backing music of the map. Choose from a BGM Properties window.

Sky: Choose one of several sky designs for the map, from the Sky Library.

Lighting: Choose one of several lighting conditions (which are shown as you select them, so aren't translated directly), or choose the top option to make your own. This leads to a window:

```

-----
>AMBIENT LIGHT >
>Hue           -[]----->
>Saturation    -[]----->
>Value         -[]----->
> >
>HERO'S LIGHT >
>Hue           -[]----->
>Saturation    -[]----->
>Value         -[]----->
>X ofs         -[]----->
>Y ofs         -[]----->
>Z ofs         -[]----->
>[OK]          >
-----

```

Animations: Turn on or off the landscape's ambient animation (water flowing, etc).

Sounds: Turn on or off the landscape's ambient sounds.

Colour: Change the base colour of the map.

Fogging: Change the depth of fog. The menu is:

```

-----
>Custom >
>Slight >
>Heavy >

```



```
>Pea soup>
>Moderate>
>Evening >
-----
```

Custom leads to the following window:

```
-----
>Hue          -[]----->
>Saturation -[]----->
>Value        -[]----->
>Distance     -[]----->
>[OK]                >
-----
```

Spotlights 1-3: Add up to three special lights to the map. The window is as follows:

```
-----
>Location     [_____]>
>Lit Area     -[]----->
>Hue          -[]----->
>Saturation   -[]----->
>Value        -[]----->
>[OK]                >
-----
```

---

#### \$ R4 Scenario Editor

---

Ahhhh..... Now *\*this\** is the big one. This editor is used to specify how you want your characters in the game to behave; essentially, to write the plot of your game.

The scenario of the game consists of three sections: one Book, and some number of Sheets with attached Scripts.

A Script specifies a series of actions which occur (objects moving, effects, messages, etc.). Effectively it specifies a CG animation, although it can include user interaction.

A Sheet specifies the conditions under which an event occurs (eg, when you talk to someone, hit an object, etc.)

The Book specifies how Sheets link to each other.

Throughout the Scenario editor the controls are as follows:

Analogue/Digital	Move pointer
A	Click
X	Pan view
L	Zoom view
R	Unzoom view
B	Cancel
Y	Switch active map

#### \$ R4.1 Book

---

When you first enter the Scenario editor, the screen will display the Book. The Book consists of a number of sheets (square items) linked together by lines.

You create Sheets using the Book in order to create events in the

game. You control the flow of events in the game by linking the Sheets together.

Each sheet has four "input" points at the top, and four "output" points at the bottom; you connect input points to output points and vice versa. "Sparks" travel along the connections, from outputs to inputs.

For a sheet to activate, it must receive a spark on one of its inputs. A sheet that has none of its inputs connected will never activate. One spark may not be sufficient to activate the sheet (depending on how the inputs are set up), but at least one is *\*always\** necessary. When a sheet has the required combination of sparks, it becomes "ready". Sheets that are ready and whose triggers lie within the current Map are shown in the Book editor with a red box around them.

The sheet itself may then have an extra condition that needs to be met in the game. When that condition is also met, the sheet "fires", its associated script runs, and a spark appears on its left-hand output (unless something in its script makes it generate it somewhere else). A single output can be connected to several places, in which case sparks are sent on all the connections. An input can also have several things connected to it; how these are dealt with is looked at in the Sheet section.

There are two special sheets in the book at the start: the Start sheet (S) and the End sheet (E). The Start sheet has a single output which will generate a spark as soon as the game begins. The End sheet has a single input; if it ever receives a spark the game ends and Dream Studio returns to either the editor or the main menu.

(NOTE: The start sheet only generates a spark when the game is started for the first time - remember that Playtesting doesn't restart the game! So if you change any connections to the Start sheet, you must do a Game Reset for the connected sheets to receive their spark.)

Once a connection has sent a spark, the connection becomes "live". Live connections, like sparks, can make sheets become ready. A spark is only sent once every time the sheet fires, but a connection that has sent a spark remains live until something turns it off. Sparks can still be sent on live connections and they remain live as a result.

To create a new sheet, click in any empty space. To link sheets together, click and HOLD on one of the input/output ports of a sheet and drag the connection to the port you want to connect it to.

To edit a connection, point at the port to which it's connected and hit A. You'll then cycle through all the connections on that port with the digital joystick. When you find the one you want, hold A and drag to detach it from that port and reattach it elsewhere, or hit Y to kill it entirely.

When you click on a sheet, you'll get the following menu:

```
-----
>Edit Script      >  (Expert mode only)
>Edit Sheet      >
>Copy            >
>Copy to another map>
>Move            >
>.....>
>Delete          >
-----
```

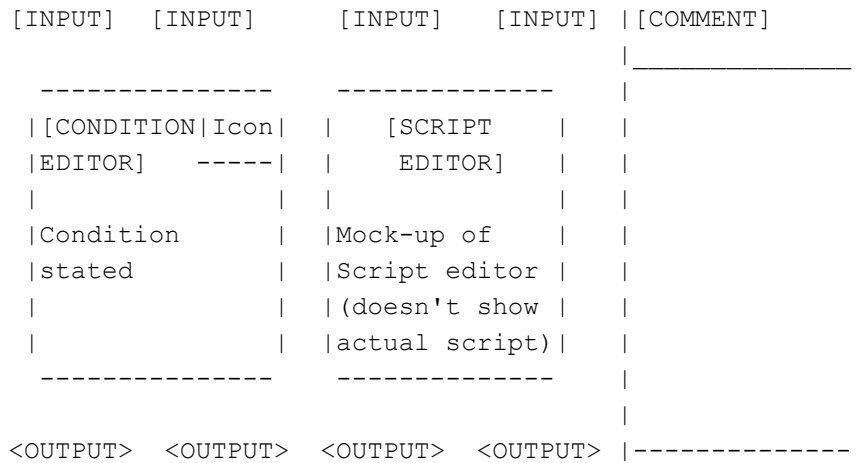
Edit Script and Edit Sheet take you to the appropriate editors. Copy and Delete are self explanatory; move will let you move the

sheet around on the book display (which you can also do by holding down A on it and dragging)

## \$ R4.2 Sheet

-----

The sheet editor looks like this:



To edit the comment, point at the COMMENT bar and hit A. You type the comment using the standard text interface. You can also edit the name of the sheet by moving the pointer up to the "Sheet Name" entry in the status bar (not shown on the diagram above) and pressing A.

To edit the condition, press on the Condition Editor box. The Icon shown in the Condition Editor box is the one that will be used to represent the sheet in the Book. It is determined by the nature of the Condition which will be looked at in R4.2.1.

To edit the script, press on the Script Editor box. You will go to the Script Editor as described in R4.3.

The four input boxes are used to edit the properties of the input connections. See R4.2.2.

### \$ R4.2.1 Condition Editor

-----

When you first choose to set the condition, you will be given the following menu of condition types:

```
-----
>System trigger          >
>Map hit area            >
>Map hit area and button > (expert mode only)
>Object hit area        >
>Object movement area   > (expert mode only)
>Object hit area and button > (expert mode only)
>Object and object collision> (expert mode only)
-----
```

System trigger: The sheet has no condition; it fires as soon as it is made ready by sparks. The icon will be an A in a box.

Map hit area: The sheet fires when the hero enters a location on the map. You will be taken to the World editor to specify the location. Use the analogue to move the area and the X button and triggers to set the size of the area. The icon will be a grid square.

Map hit area + button: The sheet fires when the hero enters a location on the map and the player presses a button. You will set the hit area

as above and you will also choose the button that will be required.

Object hit area: The sheet fires when the hero touches an object. You will be taken to the World editor to choose the object.

Object movement area: This means something like "fires when the hero moves away from the object", but I've never gotten it to fire in these circumstances.

Object hit area and button: The sheet fires when the player hits a button while the hero is touching an object. You will set the object as above and will also choose the button that will be required.

Object collision: The sheet fires when two objects crash into each other.

Remember that a sheet must be made ready (by receiving sparks) before it can fire regardless of the condition specified.

#### \$ R4.2.2 Spark control

-----

By clicking on the four inputs at the top of the box, you can choose how the sheet will behave with respect to input sparks. By and large the following rules always hold true:

- a) The sheet will NEVER be ready until at least one of its inputs receives an spark.
- b) The sheet will become ready if the specified condition is met on ANY of its four inputs. (For logic fans, inputs are ORed.)
- c) If unnecessary sparks are received (ie, one that would make it ready when it's already ready, or one that would disable it when it isn't ready), nothing happens.

When you click on an input, you will be given the following options for how connections should be treated on that input:

-----  
∨Any spark, once only ∨ (expert mode only)  
∨Any spark, multiple ∨  
∨All spark, once only ∨  
∨All spark, multiple ∨ (expert mode only)  
∨Any spark disables ∨ (expert mode only)  
∨All spark disable ∨ (expert mode only)  
-----

ANY SPARK, ONCE ONLY: If ANY connection to this port sends a spark, the sheet will become ready. Once made ready this way, the sheet will never become ready again unless it is explicitly turned off (by a spark on a "disabling" port) and then made ready again. The icon is the digits 01.

ANY SPARK, MULTIPLE: If ANY connection to this port sends a spark, the sheet becomes ready. The sheet will become ready whenever a spark is received here even if it hasn't been turned off in the meantime. The icon is the digits 01 with a looping arrow. This is the default.

ALL SPARKS, ONCE ONLY: The sheet will become ready if ANY connection sends a spark when ALL THE OTHER CONNECTIONS TO THIS PORT ARE LIVE.

Once it has done so, it will never become ready again unless it is explicitly turned off (by a spark on a "disabling" port) and then made ready again. The icon is the digits 11.

ALL SPARKS, MULTIPLE: The sheet will become ready if ANY connection sends a spark when ALL THE OTHER CONNECTIONS TO THIS PORT ARE LIVE. The sheet will become ready whenever this condition is met. The icon is the digits 11 with a looping arrow.

ANY SPARK DISABLES: If ANY spark is recieved on this port, the sheet will CEASE to be ready (if it was ready in the first place). In addition, if the sheet was a "ONCE ONLY" sheet, it will FORGET THAT IT WAS EVER ACTIVE; ie, the "ONCE ONLY" port will be able to activate the sheet again. Not only that, but all the outputs from this sheet will STOP BEING LIVE, if they were. This can happen any number of times. The icon is the digits 01 with a bar above.

ALL SPARKS DISABLE: If a spark is recieved from ANY OUTPUT connected to this port and ALL THE OTHER INPUTS ARE LIVE, the sheet will CEASE to be ready (if it was ready in the first place) and the same effect as above will happen. This can happen any number of times. The icon is the digits 11 with a bar above.

Note: be careful. "ANY SPARK MULTIPLE" means only that the sheet will become ready every time it recieves a spark. It does not mean that the sheet will be able to fire multiple times when it recieves one spark. To make a sheet that can fire multiple times, connect the output of the sheet to its own input port, so the spark it generates returns back to it and makes it ready again.

#### \$ R4.3 Script

-----  
The Script editor lets you specify what happens when a sheet triggers. This is where you will write all the events and dialogues in your game.

The Script editor's main display is a timeline: the left edge of the display is the time when the script begins to run. As you move the cursor left and right, the top left corner of the screen will display the point in time represented by the cursor's present location. Each entity involved in the script will have an activity line on the the display. Solid blocks on the activity line indicate the times at which the entity is performing actions within the script; the colour of the block indicates what action is being performed. The time resolution of the script is one frame; the actual amount of space representing this on the line will vary with the zoom level of the lines. (If you've ever used Shockwave Flash you'll know what's going on here.)

As an example of the appearance of the script editor with a few lines and actions added:

```
-----  
|Ic| Dragon | #####  
|On|         | #####  
-----  
  
-----  
|Ic| Hero   | #####  
|On|         | #####  
-----
```

```

-----
|Ic|Messages|          #
|On|         |          #
-----

```

```

-----
\           /
EVENT CONTROL
/           \
-----

```

This shows an event in which first the dragon performs an action for a few seconds, then a message appears, then the hero performs an action for a few seconds.

#### \$ R4.3.1 Script editor basics

```
-----
```

To add lines to the script: Hit the "event control" button and select the type of line you want to add. You can have only one of most lines; you can have several Object lines, provided they all refer to different objects, and you can have several Sound FX or Special Text lines. The list of lines offered is:

```

-----
>Messages    >
>Hero        >
>Object      >(you'll go to the World Editor to choose which object)
>Camera      >
>BGM         >
>Sound FX    >
>Letterbox   >
>Map         >
>Special text>
>Vibration   >
-----

```

Click on the box at the left hand side of a line to get the following menu:

```

-----
>New track    >(takes you to the menu above)
>Move track   >
>.....>
>Delete track >
-----

```

To create an event, click a point on a line where you want to create the event. You will be given a menu offering the type of event, and then the Event Properties sheet. These menus and options will vary based on the track, and on the event. To move an event, point to it, hold A and drag; to change the length of an event, use the handles attached to either end of the bar. Clicking once on an existing event will get you:

```

-----
>Event Properties >
>Copy             >
>Move             >
>Move forward +1 frame>
>.,.....>

```

```
>Delete      >
-----
```

Clicking on an area to the right of the headers where there is no track will get you the following menu:

```
-----
>Add time to all tracks      >
>Delete time from all tracks>
-----
```

Also, some of the icons that appear in the track labels have squares around them (Icon Lights). Clicking on the icon will light, or unlight, the square. When the square is lit, the state of the object represented by the track will be restored after the Event finishes. This means that any changes will only last the duration of the Event.

#### \$ R4.3.2 Event list

-----

This section will look at the list of possible events on each track and the options on their Event Properties sheets.

An important difference between events is in what is represented by the duration you assign them. The possibilities are:

**Effect duration:** The duration you set represents the period of time in which the effect you specified will take place.

**Transition duration:** The effect you specify will take place continuously and permanently AFTER the marked block of time. During the marked block, the transition will take place. In the vast majority of cases there is no standard transition and nothing whatsoever will happen in the marked time.

**Persistent Transition Duration:** Same as "transition duration", except that the effect you specify will continue applying AFTER the script has finished, UNLESS the Icon Light for the track is on.

**Special duration:** For most interactive events, you will be unable to adjust the duration on the script. When they are encountered, the entire scripts stops and waits for the user's response.

#### \$ R4.3.2.1 Messages track events

-----

The potential events on the Messages track are:

```
-----
>Message      >
>Menu         >
>String Entry>
>Button       >
>Timed button>
>Probability  >
-----
```

#### \$ R4.3.2.1.1 Message

-----

A single message is displayed in a pop-up box, dialogue style. The Event Properties box is:

```
-----
Message _____
[OK] [ADVANCED]
}=====
Position ___v
-----
```

MESSAGE: The message shown.

POSITION: The position of the message on screen. The symbol points up or down for the top or bottom.

Message has Special Duration.

#### \$ R4.3.2.1.2 Menu

-----  
The user is given a menu of options to choose from. The Event Properties is:

```
-----
Title _____
Option 1 _____
Option 2 _____
Option 3 _____
Option 4 _____
[OK] _____
-----
```

All the options represent the text of the appropriate part of the menu. When the script ends, the selection the user made will determine which output port emits a spark. Special Duration.

#### \$ R4.3.2.1.3 String Entry

-----  
The user is asked to input a string. The user must enter one of four option strings or the game cannot continue. The Event properties is:

```
-----
Prompt _____
Option 1 _____
Option 2 _____
Option 3 _____
Option 4 _____
[OK] _____
-----
```

The four options are those which must be chosen from. Use of this event is not really recommended. Special Duration.

#### \$ R4.3.2.1.4 Button

-----  
The script waits for the user to press a button. The Event properties is:

```
-----
Button 1 [_____]
Button 2 [_____]
Button 3 [_____]
Button 4 [_____]
[OK] _____
-----
```



-----  
Press A on any button option and then hit any button or direction on the digital stick to set which button will activate that option. Special Duration.

#### \$ R4.3.2.1.5 Timed Button

-----

Works in exactly the same way as Button, except it doesn't wait for the user. The user must hit a button within the time assigned to the event. The Event Properties sheet shows:

-----  
>Button 1 [\_\_\_\_\_]>  
>Button 2 [\_\_\_\_\_]>  
>Button 3 [\_\_\_\_\_]>  
>Button 4 [\_\_\_\_\_]>  
>Timeout <\_\_\_\_\_]>  
>[OK] >  
-----

The four button options work just like the standard RANDOM option, and the TIMEOUT choice is activated if the user doesn't press any button within the time assigned. The time allowed is decided by the length of the event on the track in the script. Effect Duration.

#### \$ R4.3.2.1.6 Probability

-----

Randomly picks one of four choices. The Event Properties is:

-----  
>Chance 1 -[ ]----->  
>Chance 2 -[ ]----->  
>Chance 3 -[ ]----->  
>Chance 4 -[ ]----->  
>[OK] >  
-----

The sliders determine the % probability of each of the choices being selected. No Duration.

On all the above items that give four choices, which choice is selected will determine which of the outputs from the sheet generates a spark when the script finishes.

#### \$ R4.3.2.2 Hero track events

-----

The potential events on the Hero track are:

-----  
>Serif >  
>Action >  
>Emotion >  
>Move >  
>Fly >  
>Turn >  
>Colour >  
>Visibility>  
>Model >

```
>Carrying >
>Switches >
>Map move >
-----
```

\$ R4.3.2.2.1 Serif

-----

Produces a floating bubble which can be used as a speech bubble or object indicator.

```
-----
>Serif _____>
>[OK] _____>
-----
```

Effect Duration.

\$ R4.3.2.2.2 Action

-----

Causes the hero to do an action. The property sheet is Action Properties, R3.2.4. Effect Duration.

\$ R4.3.2.2.3 Emotion

-----

Changes the hero's emotion (facial expression).

```
-----
>Emotion _____v>
>[OK] _____>
-----
```

\$ R4.3.2.2.4 Move

-----

Moves the hero automatically.

```
-----
>Movement _____v>
> X = <?> >
> Z = <?> >
>[OK] [ADVANCED]>
}====={
>Action ___v >
-----
```

Selecting Movement brings up the following menu:

```
-----
>Move to an Object>
>Move Absolute >
>Move Relative >
-----
```

Move to an Object: Select an object, and the hero will move over to it.

Move absolute: Pick a point and the hero will move there.

Move relative: Pick a point on the world editor. Dream Studio will remember how far away that point was from the hero and in which direction it lay. The script will then make the hero move by that amount, at the appropriate angle.

Action offers the following options:

```
-----  
>Default  >  
>None     >  
>Custom   >  
-----
```

Default: While moving, the Hero will do whatever action is appropriate for that model moving.

None: While moving, the Hero will do no action at all - this means it will appear to be sliding around on the ground.

Custom: Takes you to action properties R3.2.4 to specify an action of your choice.

Persistent Transition duration.

#### \$ R4.3.2.2.5 Fly

```
-----
```

This is exactly the same as Move, except the Hero can be moved around off the ground. Everything's the same, except that the Relative move now takes note of the angle with the ground as well as the angle along it.

WARNING 1: IF YOU MOVE THE HERO TO BE FLOATING IN MIDAIR (or similar location) THE PLAYER WILL \*NOT\* BE ABLE TO CONTROL THEM WHEN THE EVENT ENDS. (The Hero will run around, but there will be no motion.)

WARNING 2: FLYING TO AN OBJECT WILL PUT YOU AT THE ORIGIN OF THAT OBJECT'S COORDINATE SYSTEM. THAT MAY NOT BE ANYWHERE USEFUL.

#### \$ R4.3.2.2.6 Turn

```
-----
```

Turns the Hero. The controls are the same as Move, except there's no Action control (since you can't do an action while Turning) and the point specified in the World Editor will be the point that the hero turns to face. Hence, if you use Turn Absolute, the hero will turn to face a certain place on the map, and if you use Turn Relative, the hero will turn through the angle you specify.

(You need to be careful with this, because technically turning to a relative point turns the Hero to an absolute angle; and turning to an absolute point turns the Hero to a relative angle!)

#### \$ R4.3.2.2.7 Colour

```
-----
```

Respray your Hero, for fun and profit!

```
-----  
>Hue          -[]----->  
>Saturation   -[]----->  
>Value        -[]----->  
>Opacity      -[]----->  
>[OK]         >  
-----
```

All of these set the new values for the Hero. Persistent Transition Duration.

#### \$ R4.3.2.2.8 Visibility

```
-----
```

Changes whether or not the Hero is visible.

```
-----  
>Visibility      <OFF>>  
>[OK]           [ADVANCED]>  
}===== {  
>Effect         <____>>  
-----
```

Effect: Choose fade (4 symbols) or blink (2 symbols). Transition Duration. (Note: this is NOT persistent. If you really want the hero to be invisible, use Colour to turn his/her opacity to 0% and prepare for a very confused player.)

#### \$ R4.3.2.2.9 Model

Changes the model used for the hero.

```
-----  
>Model          !_L1_!>  
>[OK]           >  
-----
```

Persistent Transition Duration. During the transition, the Hero will disappear completely. Also note: if the Model isn't already used on the Map, it will be loaded when the event occurs.

#### \$ R4.3.2.2.10 Carrying

Changes one of the items the hero is carrying.

```
-----  
>Object         !__L1__!>  
>Hand           <____>>  
>[OK]           >  
-----
```

Hand: Choose left or right. The effect is clearly shown on the preview window.

Persistent Transition Duration.

#### \$ R4.3.2.2.11 Switches

Adjusts some miscellaneous parameters of the hero.

```
-----  
>Pose Lock     <OFF>>  
>Animation     <ON>>  
>[OK]         >  
-----
```

Transition Duration.

#### \$ R4.3.2.2.12 Map move

Moves the Hero to a new position on a new map.

```
-----  
>Location _____v>  
>[OK] [ADVANCED]>  
}===== {
```

>Effect <\_\_\_\_\_>  
-----

Location: Click here, then choose a map, then choose a location.  
The location for this type of move is always absolute.

Effect: Choose Fade or Blink, as with Visibility.

#### \$ R4.3.2.3 Object track events

-----

The potential events on the Object track are:

-----  
>Serif >  
>Action >  
>Emotion >  
>Move >  
>Fly >  
>Turn >  
>Colour >  
>Visibility>  
>Model >  
>Carrying >  
>Switches >  
>Behaviour >  
-----

All of these are identical to the corresponding events on the Hero Track, except that they affect the selected object. The one difference is: Visibility, for objects, has \*Persistent\* Transition Duration.

The one new event, Behaviour, brings up Ambient Behaviour Properties (R3.2.5) to change the object's behaviour. It has Persistent Transition Duration.

#### \$ R4.3.2.4 Camera track event

-----

There is only one potential event on the Camera track:

Camera

The Event Properties sheet is:

-----  
>Location \_\_\_\_\_v>  
>[OK] [ADVANCED]>  
}===== {  
>Follow Contour <OFF> >  
>End shift \_\_\_\_\_v >  
-----

Follow Contour: If this is ON (it doesn't use the words ON/OFF, but it's still an ON/OFF switch) then the camera will move along the contours of the map. It will also stop if it can't reach the point required because of an object in the way.

End shift: One of three options.

-----  
>None >

```
>Retain Composition>
>Pursue Target    >
-----
```

Unfortunately I don't know what these do.

Location brings up the following menu:

```
-----
>View an object>
>View absolute >
>View relative >
-----
```

VIEW AN OBJECT: Use the analogue joystick to select an object to focus on, as when selecting an object in the World Editor. Once selected, use the digital to position the camera for the angle you want.

VIEW ABSOLUTE: Positions the camera to an exact point. There are two options. If you choose the first option, you'll set the point by actually moving your viewpoint around the map and using the digital joystick to adjust the direction. If you choose the second, you can first place an object representing the camera in the world; then hit A and you will see the view from it and can choose its direction.

VIEW RELATIVE: Use the analogue and digital sticks and the L and R buttons as in the World Editor to move the camera. The camera will MOVE the same distance at the same angle as you set.

Transition Duration.

#### \$ R4.3.2.5 BGM track events

```
-----
The potential events on the BGM track are:
```

```
-----
>Start new>
>Stop all >
>Adjust    >
-----
```

#### \$ R4.3.2.5.1 Start new

```
-----
Starts a new piece playing. The properties are:
```

```
-----
>Tune      !__L5__!>
>[OK]      [ADVANCED]>
}=====
>Speed     -[]----->
>Volume    -[]----->
>Pitch     -[]----->
>Stereopos-[]----->
>Chorus    <__> >
-----
```

Most of these are self explanatory. Tunes are selected from the BGM library. "Chorus" has two choices: play all (two characters) or

play only melody (4 characters). Persistent Transition Duration.

\$ R4.3.2.5.2 Stop all

-----  
Stops all BGM. Has no properties. Persistent Transition Duration.

\$ R4.3.2.5.3 Adjust

-----  
Adjusts the properties of the current BGM. Properties are as above, excluding "Tune". Persistent Transition Duration.

\$ R4.3.2.6 Sound FX track events

-----  
The potential events on the Sound FX track are:

-----  
>Effect >  
>Stop all>  
-----

\$ R4.3.2.6.1 Effect

-----  
Plays a sound effect.

-----  
>Effect !\_\_L2\_\_!>  
>Loop <OFF> >  
>[OK] [ADVANCED]>  
)=====({  
>Volume -[]----->  
>Pitch -[]----->  
-----

Most of these are self explanatory. Effect Duration.

\$ R4.3.2.6.2 Stop all

-----  
Stops all sound effects. Has no properties. Transition Duration.

\$ R4.3.2.7 Letterbox track events

-----  
The Letterbox track has two events:

-----  
>Blackout >  
>Letterbox>  
-----

\$ R4.3.2.7.1 Blackout

-----  
Blacks out the screen. The Event Properties sheet:

-----  
>Blackout <OFF>>  
>[OK] >  
-----

The OFF/ON option doesn't show the words OFF/ON, but two Japanese phrases: 3 chars for blackout on, 2 chars for blackout off. When the screen is blacked out, only Special Text is visible. Transition

Duration.

#### \$ R4.3.2.7.2 Letterbox

-----  
Puts the screen into a letter box. You will be shown a preview of the current scene in the World Editor in a letterbox. Use the X button to move the cursor around between the corners of the view, and use the joystick to move that corner to create the letterbox. Transition Duration.

#### \$ R4.3.2.8 Map track events

-----  
The potential event on the Map track are:

-----  
>Sky >  
>Lighting >  
>Spotlights>  
>Fog >  
>Colour >  
>Switches >  
-----

#### \$ R4.3.2.8.1 Sky

-----  
Changes the Sky on the Map.

-----  
>Sky !\_\_L6\_\_!>  
>[OK] >  
-----

Persistent Transition Duration.

#### \$ R4.3.2.8.2 Lighting

-----  
Changes the lighting of the map. The Properties sheet is Map Lighting; see "Lighting" in R3.3. Persistent Transition Duration.

#### \$ R4.3.2.8.3 Spotlights

-----  
Changes the settings of one of the spotlights. The first property sheet asks you to select which of the 3 lights you want to change. The second is the standard spotlight control (see "Spotlights" in R3.3)

#### \$ R4.3.2.8.4 Fog

-----  
Changes the level of Fog in the landscape. For the property sheet, see "Fog" in R3.3.

#### \$ R4.3.2.8.5 Colour

-----  
Yep, changes the base colour. Yep, see "Colour" in R3.3.

#### \$ R4.3.2.8.6 Switches

-----  
Changes miscellaneous properties. To be specific, "Animation" and "Sound" in that order from.. you guessed it.. R3.3.



### \$ R4.3.2.9 Special text track events

-----  
The potential events on the Special text track are:

-----  
>Special text<  
>Narrative <  
-----

#### \$ R4.3.2.9.1 Special text

-----  
Generates text which appears on top of the regular game screen and can be subject to a number of effects. This text is also the only thing that can appear while a Blackout is in effect.

-----  
>Message \_\_\_\_\_>  
>Font size -[ ]----->  
>Location \_\_\_\_\_v>  
>[OK] [ADVANCED]>  
}===== {  
>Effect \_\_\_\_\_v>  
>Hue -[ ]----->  
>Saturation -[ ]----->  
>Value -[ ]----->  
>Opacity -[ ]----->  
-----

Effects are:

-----  
>None <  
>Slide from top <  
>Slide from bottom<  
>Slide from left <  
>Slide from right <  
>Grow <  
>Shrink <  
>Fade <  
-----

You can have several Special Text lines at once. In this fashion, you can do messages composed of bits in different colours and sizes or similar. Effect duration.

#### \$ R4.3.2.9.2 Narrative

-----  
Generates text that is typed out in a shaded box on the screen. Unlike Message, this has Effect Duration: you must make it long enough for all the text to type out or it will be cut off in the middle.

-----  
>Text \_\_\_\_\_>  
>[OK] [ADVANCED]>  
}===== {  
>Location \_\_\_\_\_v>  
-----

Location is the location of the box: Top, centre, or bottom. The

character will appear to point in the appropriate direction; the character for centre doesn't appear to point anywhere.

\$ R4.3.2.10 Vibration track event

-----

There is only one potential event on the Vibration track:

Vibrate

Vibrate makes the DC controller vibrate, assuming it's equipped with a PuruPuruPack.

-----  
>Power -[ ]----->  
>[OK]            >  
-----

Power sets the strength (1 to 7) of the vibration. While on the requester, you can press and hold Y to experience the vibration for yourself.

-----  
\$ X \*\*\*\*\* TECHNIQUES SECTION \*\*\*\*\*  
-----

There are lots of things you might want to do in Dream Studio. And a lot of them you probably can do, too, but they're a bit tricky first time around. Furthermore, the diagrams used by the Scenario Editor are almost impossible to read if it wasn't you who created them (thanks a bundle Kuusou.. :) ). So this section aims to explain some more advanced trickery.

-----  
\$ X1 Multiple sheets firing in one event  
-----

Most of the time, unless you're doing heavy wizardry, you'll only want to fire one sheet in one event. You might want to spark up several more, but they're at least going to have conditions, or need other sparks, so it's unlikely they'll fire right away, right?

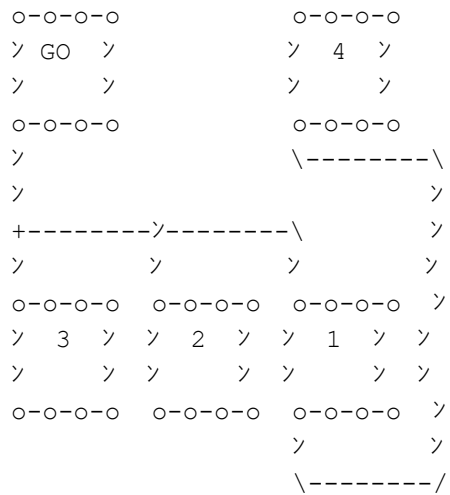
Well, this section's for heavy wizards. And being a heavy wizard is fun. When you're a heavy wizard, you do weird things, like linking up system trigger sheets that don't have any scripts attached to them. (You can do some marvellous things with this. No kidding.) And when you do that, it's useful to know a few things. Unfortunately, those few things may make your head spin the first time you encounter them. C'est la vie.

Ok, here's the first of these: which order sheets fire in. The rule is as follows, whenever sheets are ready, Dream Studio scans the grid in BOTTOM UP, RIGHT-TO-LEFT ORDER. That's the order in which sheets will fire. So say you have four sheets linked like this:

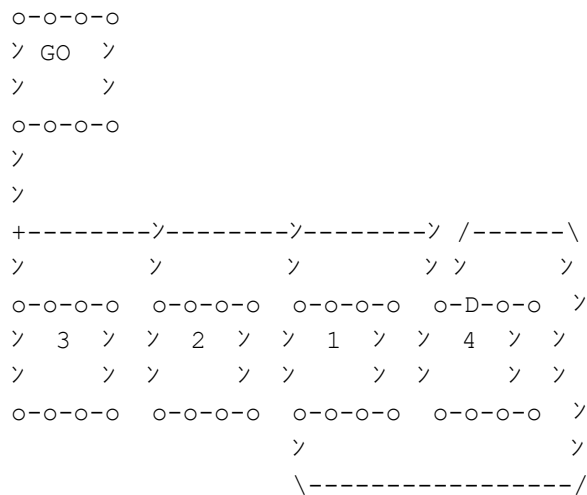




Then the firing order will be 1, 4, 2, 3. Why? We start scanning. 4 isn't ready, so nothing happens. 1 is ready and fires, and in doing so, sends a spark to 4. 4 is now ready, so the scan RESTARTS before 2 and 3 are reached! The new scan finds 4 ready, and it fires, and doesn't do anything. Then 2 and 3 are found by the scan, and they fire in that order. If, on the other hand, we have them like this:



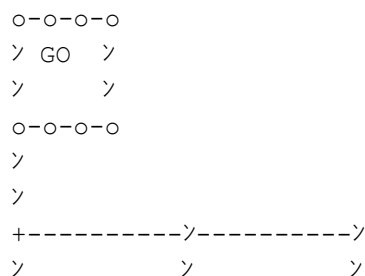
Then the firing order will be 1, 2, 3, 4. This gets especially important when you go around turning sheets off. Suppose that 4's SECOND input port is "ANY SPARK DISABLES". Then if we have:



What's the firing order? Answer: 4, 1, 2, 3. The scan reaches 4 first, and it fires. Then 1 fires, and turns 4 off... too late! It's already fired! On the other hand, if 4 is above 1, in the position shown earlier, everything is fine.

How's your head?

If it's not spinning around in circles yet, try this one. Set this diagram up in your editor and make something send a spark to the left-hand input of GO for a test run. Make the sheets numbered 1 to 3 just display a single message so that you know they're firing.



```

/--A-o-o-o /-A-o-o-o o-o-o-o
∟ ∟ 3 ∟ ∟∟ 2 ∟ ∟ 1 ∟
∟ ∟ ∟ ∟∟ ∟ ∟ ∟
∟ o-o-o-o ∟ o-o-o-o o-o-o-o
∟ ∟ ∟ ∟
∟ \-∟-----/
∟ ∟
∟-----/

```

The ports marked A are "All Sparks Multiple". So ALL the lines connected to them must be live to activate the sheet. Ok, so the firing order here is 1, 2, 3. It's just the same as it would normally be: the additional lines we've added don't do anything but follow the scan anyway. Now, let's make the SECOND input of go an "ANY SPARK DISABLES", and add an extra link...

```

/-----\
o-D-o-o ∟
∟ GO ∟ ∟
∟ ∟ ∟
o-o-o-o ∟
∟ ∟
∟ ∟
+-----∟-----∟ ∟
∟ ∟ ∟ ∟
/--A-o-o-o /-A-o-o-o o-o-o-o ∟
∟ ∟ 3 ∟ ∟∟ 2 ∟ ∟ 1 ∟ ∟
∟ ∟ ∟ ∟∟ ∟ ∟ ∟
∟ o-o-o-o ∟ o-o-o-o o-o-o-o ∟
∟ ∟ ∟ ∟
∟ \-∟-----+-----/
∟ ∟
∟-----/

```

What's the firing order now? Care to guess? Reset and run again. It's 1, 2. We'll explain that in a moment. For now, BRACE YOURSELF. Delete the link from 1 to 2's first input, then put it back again immediately. Then reset and run again.

EVEN THOUGH YOU HAVE NOT CHANGED THE LAYOUT IN THE LEAST, ONLY SHEET 1 FIRES. SHEET 2 DOES NOT.

THE ORDER THAT CONNECTIONS ARE ADDED HAS SIGNIFICANCE.

FEAR.

WHEN MULTIPLE THINGS ARE CONNECTED TO ONE OUTPUT, THE SPARKS ARE SENT IN THE ORDER THE CONNECTIONS WERE ADDED.

Ok, ready for the explanation? Here it comes. Before you made the change, here's what happened: Go sends a spark to all three sheets. All the connections go live, but only sheet 1 fires. It sends a spark to sheet 2, and since all its inputs are now live, it becomes ready. The scan does not restart immediately, since we're still in the middle of processing sparks. Sheet 1 sends a spark to GO, turning it off. This means that the connection to sheet 2 is no longer live, but we've already made sheet 2 ready - the connection changed too late! Now we're done with the sparks, we see if a sheet changed status, and it did, so the scan restarts. It picks up sheet 2. Sheet 2 fires and sends its spark to 3. 3 does not become ready, though, because the connection to GO is no longer live.

Now, when you made the change, here's what happens. GO sends a spark to all three sheets, and all those connection go live. Sheet 1 fires. It sends a spark to GO, turning it off. Then it sends a

spark to sheet 2. This time, when we come to check Sheet 2, we find that the connection from GO is no longer live, so it does NOT become ready this time! Since Sheet 2 is not ready, it doesn't fire, and doesn't make sheet 3 ready. So everything stops there.

If you want to see the order in which connections are going to activate, just click on the port in the editor. The order the connections are highlighted by the connection editor is also the order they were added in and the order they'll activate in.

The number of times this will actually be useful in anything you'll make can probably be counted on the fingers of one foot, but it's worth knowing, because it WILL bite you someday.

-----  
\$ X2 Hit Points (or counters in general)  
-----

Counters of all kinds, HP or otherwise, are generally useful things to have. But they seem a little tricky to create in Dream Studio. This section shows you how they're created.

Herewith the OFFICIAL KUUSOU HITPOINTS METHOD!

First of all, make a sheet for every HP the character has. So if the character has 10 HP, make 10 sheets. If your character has 1000 HP, change your damage scaling. :) You don't need a sheet for the initial number of HP, but you DO need one for the game over condition. These sheets should be System Triggers with no scripts. So, here's an example for someone with 6 HP:

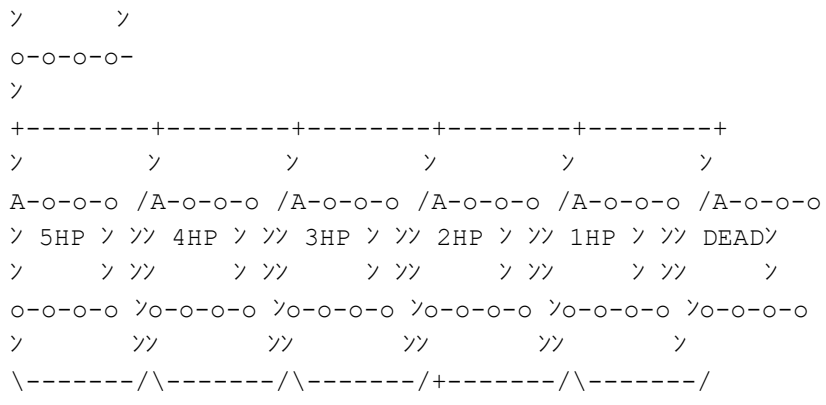
```
o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o
∟ 5HP ∟ ∟ 4HP ∟ ∟ 3HP ∟ ∟ 2HP ∟ ∟ 1HP ∟ ∟ DEAD∟
∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟
o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o
```

Ok. Now, the idea is that which of these sheets is ready is going to tell us what the player's hit point value is. We're also going to make a "damage" sheet. When we send a spark to the Damage sheet, a point of damage is done to the player. The first step is to connect the Damage sheet to all the hit point sheets, like so:

```
o-o-o-o-
∟DAMAGE∟
∟      ∟
o-o-o-o-
∟
+-----+-----+-----+-----+-----+
∟      ∟      ∟      ∟      ∟      ∟
o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o
∟ 5HP ∟ ∟ 4HP ∟ ∟ 3HP ∟ ∟ 2HP ∟ ∟ 1HP ∟ ∟ DEAD∟
∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟ ∟      ∟
o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o  o-o-o-o
```

Ok, that seems pretty pointless. So the Damage sheet activates all the sheets, which doesn't make sense. Ah, but we have an extra thing to do. First, set all of the first inputs on the HP sheets to be "ALL SPARKS, ONCE ONLY". Then link each sheet to the one behind it, so we now have this.

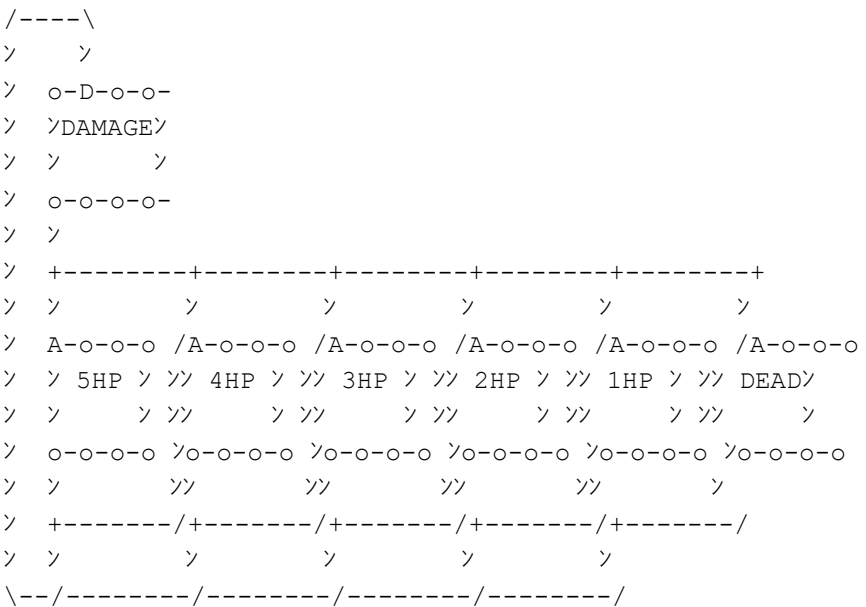
```
o-o-o-o-
∟DAMAGE∟
```



What does this do, apart from being an odd piece of abstract art? Well, the idea's is as follows: When damage is first done to the player, sparks are sent to all the HP sheets. But because all of them except 5HP depend on the previous HP sheet having fired AS WELL (because of "all sparks once only"), only the 5HP one actually becomes ready. Because it's a system trigger, it fires. When it fires, it sends a spark to the 4HP sheet. Then, next time the damage sheet is called, the 5HP one will no longer be able to become ready, but the 4HP one will; it will fire, and the 3HP will respond next time..

However, as you'll know from if you managed to read section X1 without your head falling off, this won't work. When 5HP sends a spark to 4HP, 4HP will find it has a spark from both 5HP and DAMAGE and will fire. Then it will send a spark to 3HP, which will find it has a spark from both 4HP and DAMAGE, and will fire.. and so on until the poor hero dies from a single point of damage.

To make it work, we have to make the HP sheets also disable the damage sheet. The port marked D is made "ANY SPARK DISABLES".



For reasons that were also explained previously, you ALSO need to make sure the link that turns off Damage is first in the order of activation. So you need to DELETE the link to the next sheet, THEN add the link to turn off Damage, THEN put the link to the next sheet back again. (If you don't, your hero will lose hitpoints two at a time.)

So, here's what happens: DAMAGE fires. It sends a spark to all the HP sheets. 5HP fires. It sends a spark back to DAMAGE, which turns it off. It then sends a spark to 4HP, which is not ready since

it no longer has a spark from DAMAGE. Next time DAMAGE fires, 4HP gets its spark, and fires. It sends a spark back to DAMAGE and turns it off, and sends one to 3HP.. and so on.

You now have HP sheets. Now, you can make them do interesting things. A fairly essential thing to do is to make them use a Special Text event to display the current hitpoint total.

-----  
\*\*\*\*\* LIBRARY INDEXES \*\*\*\*\*  
-----

-----  
\$ L1 - Model Library  
-----

The Model library is divided into categories. Scroll between categories by pushing left and right, and select an entry by pushing up and down.

Category: Men  
-----

Warrior  
Warrior 2  
Elf  
Priest  
Peasant  
Fat peasant  
Blacksmith  
Worker  
Sneering guy  
Old guy  
Guy with glasses  
Butler  
Knight  
"Master"  
Strongman  
Doctor  
Pauper  
Sweater guy  
Convict  
Wise man  
EDIT

(Note: Selecting Edit lets you make a new model out of any head and any body. The screen is:

-----  
>Head !\_\_L1\_\_!>  
>Head type \_\_\_\_\_v> ("type" = select alternate colour schemes)  
>Body !\_\_L1\_\_!>  
>Body type \_\_\_\_\_v>  
>[OK] >  
-----

You can select ANY head/body, not just those from the Men section.)

Category: Women and children  
-----

Warrioreess  
Warrioreess 2  
Elf



Priestess  
Cute girl  
Fat woman  
Grandma  
Goodwife  
Sneering gal  
Princess  
Maid  
Old maid  
Girl in skirt who gets used as villain a lot in sample games  
Water nymph  
Young boy  
Young girl  
Pauper child  
Rich child

Category: Monsters

-----

Angel  
Devil  
Skeleton  
Male zombie  
Female zombie  
Werebear  
Minotaur  
Golem  
"Boss"  
Dragon

Category: Architecture 1

-----

Wooden building 1-5  
Stone building 1-5  
Brick building 1-2  
Big Wood building 1-4  
Stone and thatch building 1-4  
Redbrick building 1-5  
Shed-type building 1-3

Category: Architecture 2

-----

Castle  
Cathedral  
Church  
Windmill  
Clock tower  
Tower  
Temple  
Raised house  
Ramshackle house  
Log cabin  
Log cabin w/snow  
Multiple Stable  
Single Stable  
Snowy single stable  
Igloo  
Tent  
"Bao"  
Boat

Category: Trees

-----

Tree  
Snowy tree  
Blooming tree  
Autumn tree  
Bare tree  
Thick bare tree  
Pine-type tree  
Pine-type snowy tree  
Fern tree  
Palm tree  
Cactus  
Big plant  
Small plant  
Small flowering plant

Category: Features

-----

Fountain  
Well  
Hedge  
Log  
Tree stump  
Bundle of logs  
Cannon  
Mounted cannon  
Grave 1  
Grave 2  
Grave 3  
Bones  
Rotten log  
Broken wall  
Rock 1  
Rock 2  
Rock 3  
Smooth Rock 1  
Smooth Rock 2  
Smooth Rock 3  
White Rock 1  
White Rock 2  
White Rock 3

Category: Signs

-----

Blank sign  
Point Left  
Point Right  
"INN"  
Beer symbol  
Gold symbol  
Weapons symbol  
Magic ball symbol  
First aid symbol  
Generic text  
Hanging sign: INN  
Hanging sign: Beer symbol

Hanging sign: Gold symbol  
Hanging sign: Weapons symbol  
Hanging sign: Magic ball symbol  
Hanging sign: First aid symbol  
Hanging sign: Generic text

Category: Fence

-----  
Iron gate  
Brick gate  
Fence gate  
Wood fence  
Iron fence  
Stone "fence"  
Ringfence + gate 1  
Prefab fence 2  
Raft/Bridge 1  
Raft/Bridge 1 snowy  
Raft/Bridge 2  
Raft/Bridge 3  
Suspension bridge  
Bridge curve  
Bridge straight

Category: Floor

-----  
Stone floor 1-5  
Brick floor 1-5  
Gravel floor 1-5  
Sand floor 1-5

Category: Dungeon parts

-----  
Enterance  
Wall  
Passage  
Corner  
T-Piece  
Cross-Piece  
Dead End  
Staircase  
Descent into rock  
Rock cave  
Big rock  
Rock passage  
Rock corner  
Rock T piece  
Rock cross piece  
Rock dead end  
Rock passage up

Category: Portals/Travel

-----  
Staircase up  
Staircase down  
Ladder  
Wooden door  
One-side door  
Grand door 1-2  
Strong door 1-2

Barn door  
Cell door  
Strong cell door

Category: Indoors 1

-----

Carpet Big  
Carpet Medium  
Carpet Small  
Circular table  
Square table  
Cross table  
Desk  
Draped desk  
Chair  
Bench  
Easy chair  
Sofa  
Throne 1-2  
Bed  
Bunk bed  
Princess bed  
Cramped bed  
Counter  
Counter pillar  
Chest of drawers  
Medicine cabinet  
Machine cabinet  
Bookshelf  
Empty bookshelf  
Fireplace  
Oven

Category: Indoors 2

-----

Treasure chest  
Coffin  
Crate  
Barrel  
Vase 1-2  
Pan  
Wicker basket  
Baby basket  
Ridge?  
Wheel/cog 1-4  
Pounder  
Stage stairs  
Stage curtain

Category: Indoors

-----

Altar  
Armor display  
Sword  
Shield  
Organ  
Picture  
Small picture 1-3  
Big Picture 1-3  
Tapestry

Statue  
Gold statue  
Small column  
Column  
Stone column  
Chandelier 1-2

Category: Indoor furniture

-----

Big pot plant  
Small pot plant  
Flower arrangement  
Empty vase  
Pendulum clock  
Mantel clock  
Grandfather clock  
Quill and letter  
Plaque  
Blank paper  
Blank plaque  
Book  
Open book  
Upright book  
Rope  
Food 1-3  
Candle  
Candle out  
Torch  
Candelabra 1-2

Category: Items

-----

Knife  
Frying pan  
Pot  
Vial  
Vial of liquid  
Bottle  
Chalice  
Cup  
Flask  
Syringe  
Lollipop  
Bouquet  
Guitar  
Swaddled baby  
Book  
Scroll  
Key  
Complex key  
Coin  
Crystal  
Bun  
French stick  
Cheese  
Ham  
Apple  
Sack  
Lamp

Category: Items 2

-----

Sword 1-2  
Big sword  
Small sword  
Sabre  
Spear  
Club  
Magic Staff  
Holy Staff  
Coloured staff  
Crooked Staff  
Bow  
Shield 1-2  
Axe  
Hammer  
Pick  
Spade  
Hoe  
Broom  
Standard spear

Category: Effects

-----

Transparant cube  
Warphole  
Checkpoint  
Glowing ball  
Energy ball  
Hit effect  
Slash effect  
Bloodstain  
Small heart  
Big heart  
Fire  
Inferno  
Water jet  
Burn  
Crater  
Ground patch  
Burned ground patch  
Glowing blade  
Small glowing blade  
Spotlight

-----

\$ L2 - SFX Library

-----

Cursor move  
Cancel  
Error  
Wheep!  
Window change  
Window close  
Wind  
Birds singing  
Crows  
Heavy wings  
Machinery  
Crickets

Bees  
Water dripping  
Water running  
Water running heavy  
Underwater  
Crowd  
Heavy machinery  
Rain  
Thunder rumble  
Thunder clap  
Horse  
Horse neigh  
Cockerel  
Dog Growl  
Dog Howl  
Dog Bark  
Cat  
Cheer  
Boo  
Male scream  
Female scream  
Male evil laugh  
Female evil laugh  
Female muted evil laugh  
Footstep  
Footstep echo  
Footstep on wood  
Footstep on wood spring  
Grass footstep  
Sand footstep  
Water footstep  
Switch  
Knock on door  
Door opening  
Door closing  
Door creaking  
Door slam  
Weird?  
Breakage  
Gong  
Alarm  
Phone  
Doorbell  
Big bell  
Ship's horn  
Piano  
Guitar C chord  
Guitar Cm chord  
Snare drum  
Cymbal  
Bass  
Boing  
Strum  
Wolf whistle  
Single clap  
Clapping  
Movement whoosh  
Impact  
Impact 2  
Metal impact

Sproing  
Wood on wood  
Breaking glass  
Fire  
Gun shot  
Low gun shot  
Magic warp  
Jump into water  
Rise from water  
Magic slide  
Whistle  
Eating  
Swallowing  
Flash  
Zap

---

\$ L3 - Sample game library

---

[Note: If you load the examples into the editor, do a Game Reset before playtesting, or the Sheets will not run.]

From left to right:

Initial Data

- This is the initial settings loaded when you start to create a game - the default landscape and trees and an empty Book.

Orphe's Wonderful? Country

- A complete sample game with UI.

Do your Best, Master

- A complete sample game with UI. Don't drink the blue bottle (unless you want to watch the intro twice)

Example: Opening

- A scripted opening sequence. Just watch.

Example: Meeting people

- How to encounter folks. Walk up to the people around the town.

Example: Map switching

- Walk into the cabin door and observe the location switch.

Example: Sky/weather

- Walk up the mountain and see the sky change.

Example: Getting items

- Look in the barrel and the crate.

Example: Items changing state

- Look in the treasure chest twice.

Example: Picking up and carrying items

- Get the apple.

Example: Dialogue, interaction, effects

- Talk to the angel. On the first two answers, he sends you to jail; on the third, he gives you an axe, and on the fourth, he runs off.

Example: Using items

- Try opening the door. Then look in the chest and try the door again.

Example: Menu based combat

- Take on the skeleton in FF style combat.

Example: Dungeon 1-3

- Explore the dungeons and see how to create them.

Example: Many monsters

- Destroy the skeleton without being caught by the zombies.  
[If a zombie touches you, you will "die", but the game does



not include game over so that you can see the final state of the sheets. Quit to the editor and do a game reset to play again.]

Example: Trap

- Get down the corridor avoiding the very unfair traps.

Example: Gimmick

- Push the crates around.

Example: Combat scenario

- Climb the mountain and save the priestess.

Example: World map

- Touch the sign to view the overall map.

Dream studio staff roll

- Watch the funky dancing girls and the credits for DreamStudio.

Orphe Demo

- A demo sequence from Orphe.

DreamStudio Demo

- Rolling demo showing off features of Dream Studio.

[If you leave the game idle on the title screen, one of the last three will automatically start running.]

-----  
\$ L4 - Map Library  
-----

Category: Exterior  
-----

Big mountain pass  
Glade with river  
Alternate glade  
Big house grounds  
Forest  
Snowy glade with river  
Sandy glade  
Desert  
Rocky desert  
Mountain climb  
Fire pit  
Lake glade  
Island in crater  
Waterfall  
River cutting  
Seaside  
Seaside+Jetty  
Small island in sea  
Boat at sea  
Prefab dungeon  
Rocky prefab dungeon  
Rocky cavern  
No map (must add objects)

Category: Interior  
-----

House  
Bigger house  
Tavern  
Throne room  
Church  
Double-level landing  
Stage

Mill interior  
Cells  
Circle house  
Abandoned shack  
Rogue's house  
Tiled room 1-2  
Empty wood rooms: L, M, S  
Empty stone rooms: 1-2, L, M, S  
Corner of room

---

\$ L5 - BGM library

---

[Note: I haven't translated the actual names but given a description of the piece]

Silence  
Town-type music  
MOON (introspective piano)  
Introspective wonderment  
"Lucky"  
Sirius (Strident)  
Next (Strident/Triumphant)  
Sky (Strident/V. Triumphant)  
Beat and undercurrent  
Harpsichord Landscape  
Quietly spooky  
Creepily spooky  
Sadness  
UNFINISHED [Sad, low]  
Shadows of evil  
Dreamstudio [Opening theme]  
Swoops to fanfarish  
Fast jumpy/slow  
BATTLE  
Last Fight  
"Level up music"  
"Fanfare music"  
"Joke music"  
"Morning music"

---

L6 - Sky library

---

Darkness  
Clouds 1  
Clouds 2  
Dark clouds 1-2  
Sunset  
Sunset to stars  
Stars  
Stars and clouds

---

\$ L6 - Human action library

---

Nothing  
Stand normally  
Stand innocently

Stand casually  
Walk  
Walk standard  
Walk hurriedly  
Amble  
Run  
Dainty run  
Short step run  
Tilting run  
Skip  
Drunken walk  
Zombie walk  
Sit  
Stay seated  
Eat (seated)  
Drink (seated)  
Read (seated)  
Sleep (seated)  
Play piano (seated)  
Sit on floor  
Kneel  
Lying prone  
Rising from prone  
Falling supine  
Rising from supine  
Falling prone  
Fighting with sword  
Punching  
Slapping round head  
Careful punch  
Chopping punch  
Football kick  
Groin kick  
Push  
Heavy chop  
Mid cross chop  
Overbear chop  
Jumping overbear chop  
Cautious palm  
Nocking bow  
Thumbs up  
Spear thrust  
Sssh pose  
Ready to fight  
Charge.. Fireball!  
Blocking with shield  
Lean dodge  
Getting slapped  
Getting hit in the head  
Getting hit in the midriff  
Stomach ache  
Laughing, hands on hips  
Laughing into sleeve  
Jumping  
Bracing  
Tapping foot (in time)  
Twitching foot  
Tapping foot (impatiently)  
Folded arms  
Crying

Distraught  
Wiping eye  
Wiping other eye  
Scratching head  
Patting stomach  
Posed, quivering  
Stumbling back  
Scared back  
Tough guy/gal  
Tapping sides of head  
Panting  
Abrupt panting  
Looking to sky  
Sigh  
Upward wipe motion?  
Shivering  
Sneezing  
Waving  
Saluting at chest  
Praying  
Praying on knees  
Blowing kiss  
Tapping other foot (to beat)  
Clapping either side  
Hands on knees (talking to dog)  
Bopping hands  
Calling  
Whispering  
Listening  
One clap  
Examining object  
Bracing  
Big chop down  
Rocking baby  
Playing guitar  
Blowing whistle  
Step Dancing  
Twirly dancing  
Reaching up  
Reaching down  
Throwing object  
Pushing object  
Pushing object via handle  
Failing to push object (trapped mime act)  
Knocking on door  
Opening door  
Bowing  
Deep bowing  
Set and salute  
Half Kneeling bow  
Curtsey  
Full kneeling bow  
Producing object  
Wave goodbye  
Big wave goodbye  
"Welcome"  
Waving welcome  
Engarde!  
Shooing  
Shooing hand on hip

Offering handshake  
Raising hand  
Offering hand low  
Offering hand to side  
Offering hand to other side  
Taunting  
Pointing to self  
Nod yes  
Shake head no  
Confused  
Look up, hand on hip  
Look down, hands on hips  
Look to side  
Look to other side

-----  
\*\*\*\*\* CREDITS \*\*\*\*\*  
-----

Mostly written by Mark Green, mark [at] antelope.demon.co.uk. Major kudos to Jim Breen for JDIC and to all the folks who wrote to me with encouragement, especially Colin Williamson.

And a big ganbatte! to anyone writing anything in Dream Studio at the moment.

This document is copyright MGreen and hosted by VGM with permission.