

Backbone-Manual

COLLABORATORS

	<i>TITLE :</i> Backbone-Manual		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Backbone-Manual	1
1.1	Backbone Manual	1
1.2	Updates since V2.00	3
1.3	The Future	7
1.4	Credits	8
1.5	Arexx Support	8
1.6	Various Useful Arexx Commands	9
1.7	Project Arexx Commands	9
1.8	Game Details Arexx Commands	10
1.9	Screen Setup Arexx Commands	10
1.10	Player Arexx Commands	11
1.11	Objects Arexx Commands	11
1.12	Blocks Arexx Commands	11
1.13	Levels Arexx Commands	11
1.14	Scenes Arexx Commands	11
1.15	Panel Arexx Commands	12
1.16	Controls Arexx Commands	12
1.17	Create Arexx Commands	12
1.18	Image Editor Arexx Commands	12
1.19	Sound Editor Arexx Commands	12
1.20	Block Maker Arexx Commands	13
1.21	Graphics Wanted!	13
1.22	Comparisons with other Game Creators	13
1.23	Introduction	14
1.24	Requirements	14
1.25	Tutorial	15
1.26	Limitations	23
1.27	Installation	23
1.28	Preferences	24
1.29	Common Problems	25

1.30	How to Contact Me or Register	26
1.31	The Backbone Mailing List	28
1.32	Example Projects	29
1.33	How Do I...?	29
1.34	Step By Step	30
1.35	Main Menu	32
1.36	Game Details	34
1.37	Screen Setup	35
1.38	Edit Player	36
1.39	Players Frames Settings - Side View	39
1.40	Players Frames Settings - Above View	40
1.41	Edit Objects	41
1.42	Editing a Patrolling Baddie	42
1.43	Editing a Collectable Item	44
1.44	Editing a Weapon Shot	44
1.45	Editing a Chasing Baddie	46
1.46	Editing a Baddies Weapon	47
1.47	Edit Blocks	48
1.48	Blocks Solidity	48
1.49	Blocks Animations	49
1.50	Blocks Foreground	49
1.51	Blocks Destroyability	50
1.52	Blocks Danger Levels	50
1.53	Edit Levels	51
1.54	Edit Scenes	54
1.55	Edit Panel	55
1.56	Simple Text Panel	55
1.57	Picture Based Panel	56
1.58	Edit Panel Text Element	57
1.59	Edit Panel Simple Bar Element	57
1.60	Edit Panel Object Bar Element	58
1.61	Edit Panel Still Object Element	58
1.62	Edit Controls	59
1.63	Edit Menus	60
1.64	Create	62
1.65	Image Editor	63
1.66	Sound Editor	64
1.67	Block Maker	65
1.68	Floppy Disk Tool	66
1.69	Pick Sound Effect	66
1.70	Pick Scene	66
1.71	Index	67

Chapter 1

Backbone-Manual

1.1 Backbone Manual

```
+-----+
| Backbone Manual for Version 2.08 |
|           30-July-98           |
|                               |
|           © Monolith Designs   |
|           malcolm.murray@virgin.net |
+-----+
      http://welcome.to/backbone
```

Pressing HELP at any time during Backbone will bring up this document.

Various Details

Recent Additions

The Future

Credits

Graphics Wanted!

Comparisons

General Information

Introduction

Requirements

Limitations

Installation

Common Problems

Contacting Me

Registering

Mailing List

Using Backbone

F.A.Q.

- Answers to common complex questions

Preferences

- Important Details on Changing Backbone Settings

Arexx Support

Example Projects

Step By Step Guide

Tutorial

Backbone Windows

The Main Menu

Game Details

Screen Setup

Edit Player

Edit Objects

Edit Blocks

Edit Levels

Edit Scenes

Edit Panel

Edit Controls

Edit Menus

Create

Backbone Tools

Image Editor

Sound Editor

Block Maker

Floppy Disk Tool

1.2 Updates since V2.00

Updates since V2.00

V2.08

- Added an option to have Panels above or below the main play screen.
 - Added an option to baddies that fire back at the player. You can now specify a distance at which they will keep away from the main player. This is included because you may want baddies who don't kill you by touch and thus might as well stay away and shoot you from a distance.
 - Adjusted the algorithm that allows chasing baddies to 'see' so that it is a bit more convincing when looking through platforms.
 - Backbone will now make it's own directorys on startup if it finds that some are missing. This helps with installation problems.
 - Changed the registration process. Now, if someone registers by post, I will no longer have to post them a disk with the key file on, instead I simply send them two codes which they enter into the registration entry form that appears at the start of Backbone (unless you have already got a Backbone key) and this program will automatically generate your key file. To keep hackers away, I will still send the actual Key File to users who register over the net.
 - Fixed bug where occasionally in the created games, when a new object appeared (usually a weapon shot or dead baddie) a wrong frame would be displayed for one frame.
 - Added option to allow screen shots of your game to be taken. In the Screen Setup menu there is now a checkbox. Activating it will mean that if you press 'Control' during your game, a screen shot will be saved to the RAM disk. Be careful not to do too many or to hold the key down for too long or you'll get too many screen shots and your memory will fill up fast.
 - Added option to the copper coloured backgrounds in levels. They can now be upside down so rather than going from Light to Dark they can go from Dark to Light. I also fixed a few problems with Copper colours so they work better with panels.
 - You can now use the same object file for different objects and they will not be duplicated in the created game. This means that instead of having to create a different object file for each item (or monster) you can put them all in a single object file now.
 - Added a merge option to the Image Editor. Now by clicking on Add Objects you can select a new object file and it will be added to the bank you are currently editing.
 - Fixed Arexx bug where commands would sometimes make Backbone freeze up.
 - If you load a project that has larger preferences settings than your current ones, Backbone will now give you the option of automatically changing to these new settings and reloading the project. This is done through Arexx. Backbone will adjust the preferences, quit and then restart before finally loading the project.
 - Added Fill option in the level editor. Pressing the new Fill button will fill the map between the top left corner and the the bottom right of THE CURRENT VIEW SCREEN with the block you have currently selected. The reason it does not fill the entire map is becuae when Backbone saves a map, it only save as much of it as there is blocks. If you filled the entire map
-

- with a block (unless it was the blank zero block), Backbone would save much more data than was really needed.
- Added Auto Scroll option. In the level editor you can specify details of how the scrolling will work. For example, for a space shoot em up you can make the level continually scroll in a single direction.
See
 Level Editor
 section for more details.
 - New baddie frames option added. You can now specify attack frames for any baddie. If you also click the 'Use Attack Frames' box, these frames will be used when the baddie is attacking the player. This means you can make a baddie eat or claw at the player.
 - Added Load/Save Panel option.
 - Backbone now allows you to create
 Menus
 . These are standard IFF pictures that have options attached to them. For example, you may have a main menu where you can pick to see Instructions, Start the Game, Quit the Game, Change Controls etc. Also you can have menus at the end of levels allowing the player to pick where to go next. So, if they pick "Go to the Dungeon" the game will send them to level 5 and if they picked "Go to the Forest" they would be sent to level 10 or whatever. This allows all sorts of special games to be created.
 - Normally Control+C is used to quit out of a game. Now that Menus can be used to quit out of the game, there is a new option in the screen setup to disable the Ctrl+C method.
 - Completely redesigned the Scene Editor. You can now view them all at once and also move them around to get them in some sort of order.
 - Added new sound effect option for the player. When selecting Powerhead frames in the edit player section, you can now select a 'powerhead SFX'. This sound will be played whenever the player hits a block with its head.
 - Teleporting routine slightly adjusted so that it appears the player stays still and the background changes behind him. This allows for the illusion of going through a door to look more realistic.
 - Added Cheat Mode option. Selecting the check box in the Screen Setup window will mean that you can use special keys to cheat in your game. Numerical Pad 1 will give you full lives and energy. Numerical Pad 2 will complete the level.
 - Completed redone all the example projects. They are now a bit bigger and use more of Backbone features such as menus. I had hoped to get more professional looking examples done but they are not yet ready.
 - Updated manual guide so that tutorial and example sections are more up to date.
 - Made many other various bug fixes and small adjustments to make Backbone run better.

V2.06

-Added

 Arexx

 interface along with new menu options, a special arexx window and some sample scripts.

-Added configurable

 Controls

 options.

-Changed the way weapons work. In screen setup, you now only have the

option between Weapons or No Weapons. When you edit a weapon, you get the choice there of whether it is a "left & right only" weapon or an "any direction" weapon. Make sure you check any weapons you have in your current projects and also check all the frames for both the weapons and the play firing them as they may be different.

*** IMPORTANT *** If your game had left and right weapons then the frames will certainly be wrong. You must reset the frames for the weapons bullet travelling right and also the frames of the player firing right.

- Fixed "Save Level As" menu option.
- Fixed Auto Grab function for the Player and the Baddies.
- Added more options to the preferences. You can now set some of the default paths to be used by Backbone.
- Shooting Baddies added. New options in edit baddie will allow you to make baddies that shoot at you. Patrolling baddies will fire whenever and wherever they're facing. Chasing baddies will fire at you.
- Fixed problem where the "Change to Object" option when a baddie dies was not saved properly.
- Fixed Make Object Events.
- Added Timer to the Create Process. You are now told how long it took for your game to be created.
- Image Editor now has a cycle gadget so you can choose the number of colours used by your objects yourself. This allows you to see hotspots better sometimes. Note that whatever no. of colours you view your objects in won't make a difference to the number of colours they are displayed in.
- Improved level editor by allowing you to select and paste groups of blocks at the same time rather than one at a time. Just hold and drag the mouse over a group of blocks when picking it.
- Made events more useful. Instead of being confined to a single block you can now drag an area for which they will take effect. This way you can allow big doors on the background where the player can walk through and teleport to another area in the level.
- The teleport event now instantly moves the screen to the point where the player appears.
- Loading and Saving of Objects fixed.
- Baddies no longer turn around upon hitting a ledge.
- Bug fixed where the baddies couldn't kill you if optimisation was on!
- Added full

panel

options. This was quite tough to do

so there may be some bugs. Let me know if you find any. You can now either select exactly what a simple text panel will show or alternatively use a picture panel with graphics, bars and fonts used to show energy, lives, score etc.

- Fixed Auto Decide in the Block Maker.
- New objects added in the level editor will now be visible straight away.
- Backbone now keeps the assigns it creates after quitting. They are BB: which is assigned to the directory Backbone is run in and BBTemp: which is assigned to the temporary work area (often RAM:). They are now kept because I think they are more useful to keep and don't get in the way of anything else so why remove them?
- Backbone Preferences will now ask you if you want to restart Backbone if it is already running. This is done through Arexx.
- Fixed various other small bugs.

V2.04

- Fixed problem where you couldn't click on the very top part of a screen

- such as in the block editor.
- Bug that meant you couldn't select Ghost Mode fixed.
- Adjusted the Create Game process slightly. Now uses less temporary space but may take longer to create.
- Fixed editing of Overhead baddies
- Overhead baddies now included. They should work just like the side viewed baddies.
- Adjusted example 4 to include some overhead baddies.
- Fixed level editor to allow overhead baddies to be correctly faced.
- Made arrow heads larger for those people who don't use MCP.
- Option to automatically play test a game as soon as it is created.
- Make level editing easier by being able to hold down mouse button as you draw the map.
- Changed Menu Layout slightly to incorporate more Create Options.
- Fixed optimisation as it wasn't working! Full optimisation should now make a noticeable difference to your games.
- Fixed stupid bug that appeared in V2.02 which meant baddies facing left could never hurt you! Whoops!
- Slider bars can now be seen properly even if you don't use the GadTools option in MCP.
- Fixed bug do to with baddies that could jump forever....
- Error messages in the actual game are re-enabled.
- Small screen sizes now work.
- Saving with large block banks now improved.
- Fixed bug to do with Baddies "Staying Whilst on Screen" when they die.
- 'Copy from Default' on the player frames section works better.
- Fall Speed not Blanked out for Baddies that still need it.
- Can no longer select zero object on level editor conditions.
- Set condition area on Level Ed gives more time between clicks.
- Frames can no longer be set to zero in edit player frames section.
- Preferences program changed a bit. A few fixes and more menu options. Now capable of saving to any preferences file.
- Extra option in the level editor where you can add coloured gradient effect called a rainbow in the background.
- Adjusted example 2 to show rainbows.
- Fixed bug that had appeared in V2.03 where scenes set to play before or after a level didn't work.
- Fixed a problem where 64 colours (EHB) mode wasn't being displayed properly in Backbone (although it was fine in the finished game).
- Changed preferences to allow you more blocks. 800 is now the maximum.
- Image Editor fixed a bit. You now press P to cut out a frame and a few other points are fixed too.
- Improved optimisation of overhead objects.
- Fixed stupid integrity check that would say "Object N File not Found!"

V2.03

- Installer program created.
- Bug to do with loading blank levels fixed.
- Bugs to do with Creating game with missing music fixed.

V2.02

- The Help Guide now works on the main Backbone screen.
 - Bugs fixed: Various Open and Save options fixed. Edit weapon shot fixed.
 - New Preferences option. DBL Pal mode means that all view screens will be opened in DBL Pal. Only of use to those people who set their main Backbone
-

- screen to a DBL Pal mode too.
- Special Map mode added to the preferences, this means that the level editor maps will be stored in a different way in memory and should free up space to allow you to have more blocks or bigger maps etc. The price is that it may be a little unreliable. I don't know yet.
- Baddies that die can now have the option to change to another object. This could allow you to have a baddie that morphs into another one when killed or also allow a baddie to drop an item when killed.
- Started a

Mailing List

devoted to the discussion of updates to Backbone.

- Added a new option to the baddies, there is now an option so that they will only hurt the player when they are on certain frames. This means you could have a baddie that say has spikes that come up and down and it is safe for the player to touch it only when the spikes are down. It also allows for still obstacles such as steam pipes (if you set the move speed to zero).
- Changed Image Editor so that you can press the Right Mouse Button on it without first clicking the left button. This meant getting rid of the pull down menus and changing them to buttons.
- The Block editor now shows "S" for a solid block.

V2.01

Backbone will now open any type of public screen due using the new preferences. This means the Graphics card owners can use their own custom screens with Backbone.

V2.00

An almost complete rewrite means that Backbone is now 80k smaller and runs as an OS friendly Workbench Application using Gad Tools. Also, the annoying tool type preferences have been replaced by a proper preferences program.

1.3 The Future

The Future

Here is a list of all additions I intend on adding in the foreseeable future, some of them will be done very quickly and some will be further ahead. They are collected from bug reports and suggestions sent to me. Join the

Mailing List

to discuss other future options.

Known Bugs to be Fixed as Soon as Possible

-No known bugs yet!

Intended Additions

- Allow option to change amount of scrolling lag.
- Ability to encrypt and powerpack game files.
- Ability to carry multiple Weapons and change between them.
- Ammo.

- Mouse control.
- More weapon types. e.g. Diagonal R-Type ones. Guided ones etc.
- More intelligent baddies that can climb ladders etc.
- Different death sequences.
- Allow Level Maps to be viewed.
- More Arexx support commands.
- Allow seperate GFX for when a player/object turns around so that text on the frame will not be mirrored.
- BBBasic. Backbone Basic will be a simple programming language to allow you to insert extra special features in your games.

1.4 Credits

Credits

Backbone was created by Monolith Designs.

All programming by Alastair Murray.

Thanks to the following graphic artists for help on the examples:

-Manual Tonatiuh Moreno

-Michael Mueller

And for the Backbone Logo Picture:

-Phil Price

And of course, thanks to all the users who have helped in testing and providing new ideas and support.

1.5 Arexx Support

Arexx Support

Backbone supports arexx to some extent. I don't know a lot about arexx but if anyone is willing to write their own functions then let me know and I'll add any functions you need into Backbone. Currently I've just got enough in there to do some basic functions. I can add quite complex ones if required.

Examine the current arexx scripts to get an example of what they do and how they work. Examples for the future could consist of automatically cutting out frames from a picture for example.

Commands

Each command will return a value. Typically this will be 0 if every thing went fine and 1 if there was an error.

Currently supported commands topics are:

Various Useful Commands

Project File Commands

Game Details Commands

Screen Setup Commands

Player Commands

Blocks Commands

Levels Commands

Scenes Commands

Panel Commands

Controls Commands

Create Commands

Image Editor Commands

Sound Editor Commands

Block Maker Commands

1.6 Various Useful Arexx Commands

Various Useful Arexx Commands

REQUEST <TEXT>!<OPTIONS> - Opens a requestor. The ! is used to separate the main body text and the options available for the user to click on. The & symbol can be used as a new line option. The | symbol is used to separate options.

e.g. "Line1&Line2!One|Two|Zero" will open a requestor looking like this:

```

-----
|      Line1      | Line1 and Line2 are the description.
|      Line2      | One,Two,Zero are the buttons to click on.
|-----|
| One | Two | Zero|
-----

```

This function returns zero if the right most gadget is clicked and then one, two, three counting from the left for the rest.

RESTART - Will quit Backbone and run it again. This is generally only useful if you have changed the preferences and want them to take effect.

1.7 Project Arexx Commands

Project Arexx Commands

PICKPROJECT_NEW - Opens a file selector to start a new project. Doesn't actually start it though. Use STARTNEW <FILE> to do this.
STARTNEW <FILE> - Starts a New Project as File Name.

PICKPROJECT_LOAD - Opens a file selector to load a project. Returns the selected file name with path.

OPEN <FILE> - Opens FILE as the project. Supply path name.

PICKPROJECT_SAVE - Opens a file selector to save a project. Returns the selected file name with path.

SAVE - Saves the currently opened project.

SAVEAS <FILE> - Saves the currently opened project as <FILE>. Note that Levels are not saved under the new project.

1.8 Game Details Arexx Commands

Game Details Arexx Commands

GAMEDETAILS OPEN - Opens the window

GAMEDETAILS CLOSE - Closes the window

1.9 Screen Setup Arexx Commands

Screen Setup Arexx Commands

SCREENSETUP OPEN - Opens the window

SCREENSETUP CLOSE - Closes the window

VIEW SIDE - Sets the project to a side viewed game.

VIEW ABOVE - Sets the project to an above viewed game.

BLOCKSIZE n - Sets Block Size where n is 16 or 32.

RES n - Sets Screen Resolution where n is 0-3 (0=Lowres)

HSCROLL n - Sets Horiz. Scrolling to Smooth/Chunky if n is 0/1

VSCROLL n - Same for Vertical Scrolling

COLOURS n - Sets the number of colours. n can be 8,16,32,64

SCREENSIZEX n - Sets the Horiz. Screen Size in pixels. Only use after setting Block Size.

SCREENSIZEY n - Sets Vert. Screen Size.

GUN OFF - Disables Weapons in the Game.
GUN ON - Enables Weapons.

1.10 Player Arexx Commands

Player Arexx Commands

PLAYER OPEN - Opens the window
PLAYER CLOSE - Closes the window

PLAYERMORE OPEN - Opens the extra options window
PLAYERMORE CLOSE - Closes the extra options window

1.11 Objects Arexx Commands

Objects Arexx Commands

OBJECTS OPEN - Opens the window
OBJECTS CLOSE - Closes the window

1.12 Blocks Arexx Commands

Blocks Arexx Commands

BLOCKS OPEN - Opens the window
BLOCKS CLOSE - Closes the window

1.13 Levels Arexx Commands

Levels Arexx Commands

LEVELS OPEN - Opens the window
LEVELS CLOSE - Closes the window

1.14 Scenes Arexx Commands

Scenes Arexx Commands

SCENES OPEN - Opens the window
SCENES CLOSE - Closes the window

1.15 Panel Arexx Commands

Panel Arexx Commands

PANEL OPEN - Opens the window
PANEL CLOSE - Closes the window

1.16 Controls Arexx Commands

Controls Arexx Commands

CONTROLS OPEN - Opens the window
CONTROLS CLOSE - Closes the window

1.17 Create Arexx Commands

Create Arexx Commands

CREATE - Normal create option as if you clicked on the button.

CREATE_NOREQ - Creates game with no requestors. Creates it straight to the
Temp Directory as directory Game.

RUN - Runs the last created program.

1.18 Image Editor Arexx Commands

Image Editor Arexx Commands

IMAGE EDITOR OPEN - Opens the window
IMAGE EDITOR CLOSE - Closes the window

1.19 Sound Editor Arexx Commands

Sound Editor Arexx Commands

SOUND EDITOR OPEN - Opens the window
SOUND EDITOR CLOSE - Closes the window

1.20 Block Maker Arexx Commands

Block Maker Arexx Commands

BLOCK MAKER OPEN - Opens the window
BLOCK MAKER CLOSE - Closes the window

1.21 Graphics Wanted!

Graphics Wanted!

I want to show off Backbone some more. As I am a very poor artist its hard for me to come up with the nicest looking example games. If there are any good graphics artists out there willing to help I'd appreciate it very much.

I already have people working on side viewed games that are going to look very nice, what I need now is an overhead viewed example. Something like the Chaos Engine would be great.

The graphics should be 16 or 32 colours, low resolution. I need blocks (16x16 or 32x32) and characters, baddies, weapon shots along with title screens etc.

Anyone who gives a significant contribution (i.e. enough complete GFX to make 90% of a game) will recieve a free registration and my thanks.

1.22 Comparisons with other Game Creators

Comparisons with other Game Creators

There are plenty of game creators around and basically the reason I made Backbone is because they are all rubbish! Addmittedly Backbone is still under development and doesn't have as many features as it should yet but I think it already has much more potential than most others.

Reality

I've had a lot of e-mail from people saying "I tried the reality game creator and it was so unfriendly that I gave up. Then I tried Backbone and it's much better." I tried Reality myself and it took hours to actually get the program up and running, even after that it crashed every time I quitted out of it. The games look nice but it certainly doesn't have the potential of Backbone. CU AMIGA said that Backbone games looked somewhat plain. Well I'd agree that my examples are plain but that is because I'm a terrible artist. There is nothing in the graphics department that Reality does that Backbone can't. I hope to get some much nicer examples going soon to show how nice the games can really look.

SEUCK

Errr. Not much to say about this one. Backbone can do far more. SEUCK just creates the same game over and over again with different graphics.

Click and Play/Click and Create/The Games Factory

These are all PC products made by the same person who made the excellent AMOS on the AMIGA. Its such a shame that hes reduced himself to making this sort of rubbish that is plainly intended for stupid kids. Basically, the fact that it runs in Windows is big problem, displaying that much graphics just kills all speed. The games it creates don't use proper blocks and therefore you end up with slow, jerky scrolling. I also think that the method it uses means it takes MUCH longer to create a level. Whereas with Backbone you make a baddie and then can put it anywhere in the level, with these programs you have to set up all sorts of things for EVERY object you place in a level!

1.23 Introduction

Introduction to Backbone

Backbone allows you to design scrolling or non-scrolling games. These games can be viewed from either above (e.g. Chaos Engine) or from the side. (e.g. Super Mario Brothers).

The games can have all sorts of aspects to them. You may choose the number of colours used and the screen resolution. You can define introduction sequences consisting of music, animations and pictures. You can define sounds to be used in your games. All sorts of different types of monsters (goodies and baddies) and items can be used.

How It Works

Backbone creates a main module file that caters for all of the different kinds of games that it supports. Within this file it inserts the necessary data to make your game specific to how you made it.

1.24 Requirements

Requirements for Backbone

Workbench 2 or greater is needed for the graphical interface.

A Hard Disk is almost essential, I suppose if you have lots of RAM and 2 or 3 disk drives you get away with running it on floppy but it would be nasty.

Backbone uses temporary disk space to hold files used. If you have plenty of memory you may set it to use the RAM disk as it's temporary storage. Otherwise, any place on your hard disk may be used.

A paint program is also required if you wish to draw your own graphics. This must be capable of saving IFF ILBM pictures. All amiga paint programs support this format usually as the default option. Deluxe Paint, Personal Paint and Brilliance are three common ones.

You will need for following libraries some of which are included here. The others should be available on your workbench disk.

Arp
Easylife
Explode
Mathieeedoubas
Mathtrans
Powerpacker
Reqttools

You will need several commonly used programs in your C directory. All of these should be included on your Workbench disk.

Copy
Assign
Run
More
Delete

As Backbone uses Workbench's standard Gadget Tools for its interface, the look and feel can be easily adjusted using programs like MCP or Visual Prefs.

If you have any troubles with missing files then
contact me
.

1.25 Tutorial

Tutorial

This tutorial will take you through all the steps to create the "Tutorial" example project. It will use the graphics and sounds as provided in the archive.

It is advised that you first play the Tutorial project to see how it will look. The game will be a side viewed platformer. Your character will have the "Powerhead" ability to destroy blocks once he collects a certain item. Upon collection he will change colour. There will be two baddies to kill before the level is completed.

First open Backbone.

First we must create the needed graphics and sounds for the project. Starting with the graphics. On the main menu go to the "Image Editor"

We will first pick up the graphics of the main player. As we will just pick up all the graphics in a row, turn AUTO MODE to ON.

Select "New Picture" to load the desired IFF picture containing the graphics. From the file selector, select the file "Tutorial_Objects.IFF" You will see that the picture contains various frames of the character. There is one still frame, 3 walking frames, a jumping and a falling frame, also 7 death frames. These are coloured both RED (for normal) and BLUE (for powerhead). All of these frames must be picked up.

Using the mouse, draw a box around the RED still frame (at the top left side). The box does not have to be accurate as it will automatically clip the object to size. Now you will see that Backbone has returned to the Image Editor and the Frame number has moved onto 2. Clicking on the left arrow will show the first frame that you just cut out. Make sure you move back to frame 2 before continuing.

Now, in the same way, cut out all the other frames of the character. Click on "Pick Image" to do it.

This order would be easiest:

Red Frames

1. Still
- 2-4. Walking
5. Jumping
6. Falling
- 7-13. Dieing

Blue Frames

14. Still
- 15-17. Walking
18. Jumping
19. Falling
- 20-26. Dieing

Once they are all cut out you must set their "hotspots". The place where their coordinates will be measured from. To do this go back through each frame and click on the image. Right mouse button will position the image in a convenient position. Left mouse button will set the hotspot. All hotspots should be set to the center of the base of the feet.

When this is done, select "Save Objects". Save the file as "MyPlayer"

Now we need to cut out the other graphics needed. On the same picture you will have noticed graphics of an item with "Pow" written on it. This is the powerhead item. There are 4 frames of it and 7 collected frames. There are also 5 frames of a small grey baddie (4 moving, 1 dead) and 5 frames of a coin (4 animated, 1 collected)

Each of these different objects should be held in its own object file. NOTE: Backbone doesn't mind if you use the same object file for different objects however it is wise to keep each object file to the same type of objects for clarity. For example, put all the Items frames in a file called "Items" and all the Baddies frames in a file called "Baddies". However, for this example we will store each object in a different file.

For each of these objects do the following:
Click on "Delete All" to erase any current images.
Click on "Pick Image" and pick up all the needed frames.
Set their hotspots to their bases.
Save the files under the following names:

Powerhead Item as "MyItem"
Baddie as "MyBaddy"
Coin as "MyCoin"

Repeat these steps until all objects have been picked up.

Now close the image editor.

Next we need to gather all the blocks that will be used for the levels. Go into the "Block Maker". Click on "New Picture" and select the file "Tutorial_Blocks.IFF". You will see all the blocks drawn. The block size is 16x16 pixels so set the cycle bar to "Block Size 16". There are no spaces in between the blocks so set the other cycle bar to "No Spaces". Now click on "Grab Blocks". As file name must be entered. Call the file "MyBlocks". You must now select the last block in the picture. Click on it (the furthest right block on the second row) The blocks will now be saved to disk. Close the Block Maker.

Next the sounds to be used must be made. Go into the "Sound Editor". First a sound to played when the player collects the Powerhead item must be loaded. Click on "Load Sample" and select the file "Halleluyah.smp". You can change the frequency by pressing the left and right arrows or listen to the sound by pressing "Play". The other sounds needed are as follows:

2. "AlienDie.smp" used for the baddie dieing.
3. "Ow.smp" used for the player when he is hurt.
4. "Collect.smp" used for when the player picks up a coin.
5. "HitBlock.smp" used for when the player uses his Powerhead ability.

The sample names can be changed. Give them all a suitable name before using the pulldown menu to save the sounds as "MySounds"
Exit the Sound Editor.

Now that the graphics and sounds are ready we must start on the actual game. From the pulldown menus select "New Project" In the file selector, enter the name of the project. "MyGame" is recommended. When this is done you will see the name of the project on the top bar of the main menu.

First we must set up the game details. Go into the "Game Details" menu. For the sound bank click on "Pick File" and select the sounds we just saved as "MySounds.Snd" As for the introduction etc. they can be created using the
Scene Editor
but will not be covered in this tutorial.

Now exit the Game Details and go into the "Screen Setup". All the game graphics used 32 colours, low resolution and 16x16 sized blocks. So select "Lowres", "32 colours" and "16x16 Block Size". Now click on "Full NTSC Screen" to set up the screen size to defaults.

Both the Horizontal and Vertical Scrolling should be "Slide Scrolling". The game should be set to "Side View" and "Gun: No Weapons at all".

We do not want the cheat mode or screen shot modes activated so ensure that there are no ticks besides them. However, we do want to be able to quit out of out game so ensure there is a tick next to "Ctrl+C Quits Game".

Now the game basics are set up we can start to edit the player. Exit the screen setup and enter the "Edit Player" menu.

We must first select the object file we saved earlier. Click on "Pick File" and select the file "MyPlayer.Obj". We must set the various speeds used within the game. They are usually found by trial and error. For this project use these values:

Moving Speed: 6

Falling Speed: 12

Jumping Speed: 12

Energy: 1 (it will only take one hit to kill him)

Lives: 3

Climb Speed: 0 (as the player cannot climb)

Max Jump Height: 48 (the player will be able to jump 2 blocks)

Crawling Speed: 0 (the player cannot crawl)

Crawling Height: 0

Acceleration: 2 (should be less than Moving Speed to add some sliding movement to the player)

Click on "MORE" to bring up more options.

As the game contains no weapons, most of these are not needed but you should select the "Cannot Crawl/Duck" option.

We must also set up what happens when the player dies. We will set it so that the player will lose a life but continue the game, being invincible for a short time.

Set the "After Death" option to "Continue" and the "Invincibility Timer" to 20. We don't want the player to START the game with the powerhead option (as he has to first collect an item) so set "Powerhead Off".

We will leave the player's height and width a moment and now set the player frames. Click on "Set Frames".

From here we must set each individual frame to be used in the game.

Clicking the "View" buttons allows you to see the frame in the screen below. You can position the image by clicking on it.

By selecting different radio buttons on the left, you set different frames on the right using the buttons.

First select "Still".

Set the "Still Frame" to whatever you cut out as your still frame (I will assume you used the recommended values which was 1 in this case)

There is only one still frame so set "No. of Still Frames" to 1.

Now select "Moving". Set "1st Move Frame" to 1 and "No. of Move Frames" to 4. Clicking on "View" beneath the "No. of Move Frames" will allow you to see the animation of the player moving.

Set "Jumping Frame" to 5, "Falling Frame" to 6 (presuming you used the recommended values, if not then adjust them appropriately)

Set "First Dieing Frame" to 7 and "No. of Dieing Frames" to 7.
The player cannot Crawl or Climb so ignore the other values.

Now we have set all the frames for the player with no weapon and no powerhead. As there will be no weapons in this game, we can ignore the weapons section but we must now set all the frames for "Powerhead On".

Click the "Powerhead On" button.
You must now go through all the frames again except selecting the BLUE coloured images this time.

Finally, we must set the sounds to be used with the player.
Make sure you have set Powerhead Off.
Click on "Still". You will see a button near the middle saying "Been Hurt SFX". This is the sound that will play when the player is hit by a baddie. Click on the button. Select the "OW!" sound effect and press OK.
You can set similar sound effects for Moving, Jumping, Falling etc. but we don't need those now.
We do need to set the Dieing Sound Effect to "OW!" also so do that.
Now set Powerhead to On. The Sound Effect will now show a "Powerhead SFX", set this to the "HitBlock" sample.

Now click on "BACK" to return to the previous menu.

Now we have set the frames we can get the computer to automatically detect the size of the player so we don't need to manually set it ourselves. Click on "Auto Grab". After a short time the values should be set.

Now exit the menu.

We have three objects to create yet. The baddie, the Powerhead item and the coin. Click on "Edit Objects"

We will define object 1 as the baddie.
Ensure you are on "Object Number 1" and select "PICK FILE". Pick the baddie we saved earlier as "MyBaddy.Obj". Click on the text gadget and give a suitable name to the baddie.
Click on the cycle gadget and change the TYPE to "Patrolling Enemy". Now we must set up the details of the baddie. Click on "Edit Details".
As in the Edit Player section we must now set the various items of the baddie. Set the following values:

Moving Speed: 3
Falling Speed: 12
When Hit Player: Zap Energy
Energy to Zap: 1
Energy: 0 (it will only take one hit to kill it)
After Death: Stay Forever - the body will stay on the map forever.
How it Dies: When Jumped On

We want the baddie to be a walking baddie, he should also fall off ledges (rather than turning around) but we don't want it to jump. Therefore set the top cycle gadget to "Walking+Falling".

There is a "Ghost" option where the baddie can walk through walls. We don't want this so set it to "Normal Mode".

Now lets set the frames. Click on "Set Frames".

As in the Edit Player section we need to do the same thing. Presuming you used the obvious values, set the frames to the following (if not then suitably adjust them):

```
First Frame: 1
No. of Move Frames: 4
1st Dieing Frame: 5
No. of Dieing Frames: 1
Jump Frame: 1 (not really needed)
Fall Frame: 1
```

Set the "Die SFX" to the sound effect of the baddie dieing.

Click on "BACK" and then "AUTO GRAB" to get the size of the baddie.

Now close the menu. We must now define the next object, the Powerhead Item. Click on "New Object" and then "Pick File" to be the "MyItem.Obj", as before change the items name and TYPE to "Collectable". Now click "Edit Details". Change the cycle gadget to "Powerhead On". Set the frames as follows:

```
First Frame: 1
No. of Frames: 4
1st Collected Frame: 5
No. of Collected Frames: 7
```

Collected SFX: Halleluyah!

Now close the menu. Next we must define the last object, the coin.

Click on "New Object" and "Pick File" as "MyCoin.Obj", change the name and TYPE to "Collectable". Now click on "Edit Details".

We want the coin to increase the players score so change the cycle gadget to "Add To Score" and "Amount" to 1.

Set the following variables too:

```
First Frame: 1
No. of Frames: 4
1st Collected Frame: 5
No. of Collected Frames: 3
```

Collected SFX: Collected

Now all the objects are defined, close the menu and return to the main menu.

Now we must define the blocks. Click on "Edit Blocks". Ensure you are on the 1st block bank and "Pick File" as "MyBlocks.Blocks" as we saved earlier.

Now click on Solidity. From here we can decide which blocks are solid (i.e. the player can't fall or walk through them) and which are not. There are further options but in this example we will just use Solid or Not.

Solid blocks are shown as a square with a cross through it.

Click on each block that should be solid. Counting from the left these should be blocks 1 to 10.

Ensure you only click once on each block. You can reset their status by clicking the right mouse button on them.

When you are done, click on "DONE".

We want some animated blocks so click on "Animations"

Near the end of the blocks are 4 blocks that are to be animated. They are in a row and are a sort of "fan" rotating.

Click on the first of these 4 blocks. Select "No. of Frames" as 4 and "Animation Type" as ordered.

Now click on Done. Whenever that first block is drawn in a level it will be animated as we defined it.

We want the player to be able to destroy some blocks using his "Powerhead". Click on "Destroyable". Decide on which blocks should be destroyable and click on them twice so that a small rectangle is shown at the bottom of the block. The blocks that should be destroyable are blocks 8,9 and 10. As you can see, block 11 is empty meaning that when you destroy block 8, it will change to 9, destroy it again to 10 and then finally to make it an empty block. This means it will take three hits to completely destroy one of these blocks.

These blocks can now be destroyed if the player bashes his head against them when owning Powerhead.

Click on Done.

We do not need to define any further block types so close the menu.

Lets quickly define the panel for the game too. Click on "Edit Panel" and change the panel type to "Simple".

Now we can create a level. You should first save the game though by selecting "Save Project" from the pulldown menu. After that, click on "Edit Levels".

Ensure that you are Level 1 and that it uses Block Bank 1 and then click on "Edit Level"

You should get a blank screen ready to draw the level on. If you've played the example 1 then you should know what the level is to look like.

Heres a simple diagram.

```
XXXXXXXXXXXXXXXXXXXXXXXXXX
X                               X
X      X                       X   X
X      XXXXXoooXXXXX          X
X                               X
X                               X
XXXXXXXXXXXXXXXXXXXXXXXXXX
```

where X is a solid block (it doesn't matter which)
and o is a destructable block.

Click on "PICK BLOCK" or pressing space will allow you to change the block you are using and the mouse is used to paint the blocks onto the screen. Use the arrows or cursor keys to move the map around. You shouldn't make the level much bigger than one screen for this one. You can also paint some background blocks in if you wish, some animated fans for example.

After you have painted the blocks we will position the player, baddies, powerhead item and coins according to this diagram.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                               X
X   X  B      B  X           X
X   XXXXXoooXXXXX          X
X                               X
X P       c  c  c           I X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

where P is the Player, B is a baddie, I is the powerhead item and c is a coin.

First lets position the player. Make sure the map view is in about the right position and click on "Start Pos" then use the mouse to position the player where we want him. In the bottom right corner you should see some information telling you whether the position is OK, In a solid wall or in mid air. You should only click on OK when it says "Position OK".

Now lets position the baddies. Click on "OBJECTS" and position the screen so you can see where we want to place the baddies. Click on "NEW OBJECT" and select the object that you want (the baddy), then click on "OK". Decide which way it is to face and click OK. Now position the baddie in the same way we did with the player. Now repeat the process for the second baddie.

If you make a mistake you can press "Del Object" to remove an object.

Do the same thing for the Powerhead item and the three coins.

Now the general level layout is complete we can set up the final details. First we'll set up the conditions for the player to win the level. Click on "Conditions"

We want the player to complete the level as soon as he's killed both baddies so we only need one condition. On Condition 1 set the cycle gadget to "Kill Baddies" and Object Number to 1 (it should say the baddies name on the screen) and the Amount to 2. Now click on "DONE"

You can also select some Music for the level by clicking on "Music" and then "Pick File". Note that you must pick a protracker music file.

Finally clicking on "DONE" will finish the level. When you are asked if you wish to save the level, click on "YES".

Your game is now complete. If you wish you can add introduction sequences etc. by using the

Scenes Editor

.

Click on "CREATE!" and select a destination for the game. When complete, iconify or quit Backbone and run the game.

Note that if the game is not fast enough you can increase the level of optimisation in the pulldown menu.

1.26 Limitations

Limitations with this Version of Backbone

The Registered Version:

This is a demo of Backbone to allow evaluation of it. Any game created with this version will have a message appear at the beginning to prevent distribution. You must

Register
to remove this limitation.

Further benefits of registration allow personal changes to Backbone. Contact me with ideas of improvements or even specific additions to your individual game and I'll do my best to help you.

Later Versions:

As time progresses, ALOT of improvements will be made and many additions will evolve this into a much better and very powerful product. But for this to happen I must have some sort of feedback. If you feel you would make good use of backbone then either

Register
now

for a cheaper price or just e-mail me to let me know you'll be interested once it develops further.

1.27 Installation

Installation

Run the installation program to install Backbone. It will automatically create a directory named Backbone. It will also update any previous version of Backbone as long as you select the correct directory. e.g. If you're current version of Backbone is in Work:Programs/Backbone then select Work:Programs in the installer.

Manual Installation

Backbone is quite easy to install manually if you have troubles with the installer program.

Copy all LIBS into your LIBS: drawer and all FONTS into your FONTS: drawer. Copy everything else into any other drawer (naming it Backbone would be a

good idea)

This is how the directory structure should look, you'll have to create some of the drawers yourself as LHA doesn't like empty drawers.

```
Backbone
|-Blocks
|-Code
|-Files
|-Icons
|-Libs (not needed)
|-Modules
|  |-Blocks
|  |-Controls
|  |-Details
|  |-Levels
|  |-Menus
|  |-Objects
|  |-Panels
|  |-Player
|  |-Scenes
|  |-Screen
|-Music
|-Objects
|-Pictures
|-Projects
|-Rexx
|-Samples
|-Sounds
```

1.28 Preferences

Preferences

Backbone has various Preferences settings that can be adjusted in order to set it up as you wish and help prevent crashes.

Run the Preferences program to adjust any of the settings. You must run Backbone again to make use of the new settings. Backbone will load the prefs file from Files/BB.Prefs so save your files there.

As a rule, decreasing any of the values will help protect against crashes. However, if you increase them too much, Backbone will fail to run. If this happens, try using the Special Map Mode. If all else fails a special version may be obtained from me for registered users that allows more of everything.

Another point to note is that if you reduce these values and then load a project that used higher values, problems may arise. There should be no crashes, but some objects, events or whatever may be missing. For the most part, unless you reduce them to very low values, there will be no problems.

You must save the preferences from the pull down menu before you quit if you wish them to take affect.

Your options are:

SCREENMODE By default, Backbone will run on a hires screen. This can be changed using this option. It is most useful for owners of GFX cards.
 NOTE: There is no need for screens with lot of colours, 4 will do.

REQUESTORS If this is clicked Off, then Backbone will only show requestors if they are very important.

VIEW SCREEN When Backbone opens view screens to see graphics they are in the normal AMIGA modes. However if you have selected the SCREENMODE option to be a DBL Pal mode then you should set this accordingly.

MAP MODE Set this to special will force Backbone to store the level maps in a different way. This will free up space for more blocks and events etc. but may cause crashes on some machines.

X X and Y set the size of the level maps in Backbone. Larger values
 Y will give you larger maps but will use more memory and will increase the chances of crashes.

EVENTS This sets the maximum number of possible events on any level.

WEAPONS This sets the maximum number of weapons that can be defined.

BLOCKS This sets the maximum number of blocks that can be used. Note that you can still use extra block banks to increase this number.

CONDITIONS This sets the maximum number of conditions used in a level.

OBJECTS This sets the maximum number of definable objects.

SCENES This sets the maximum number of definable scenes.

SET PATHS From this extra you can decide on some of the default paths that Backbone will use when looking for pictures, sounds etc.

1.29 Common Problems

Common Problems

 Firstly, if you find a bug in Backbone and have internet connection, check that you have the latest version. The bug may well have been fixed already. Check AMINET under dev/misc/Backbone2_main.Lha
 and dev/misc/Backbone2_ex.Lha

If you e-mail me about a problem, ensure you give a full description along with you machine spec. A snoop dos log sometimes helps as well.

Problem: Backbone will not run on my 060.

Solution: Some people have reported problems using Backbone with an 060. The solution that came up with was to disable all superscalar caches.

Problem: When I run the created game, either straight away or after a short time, it suddenly quits back to the Workbench.

Solution: 1. You are low on memory. Quit all applications and make sure you don't create the game and RAM and then run it again.
 2. You have come across a bug! E-mail me with the files needed and I'll solve it for you.

Problem: The colours are all wrong when I display blocks or objects.

Solution: 1. The setting for the number of colours may be wrong.
 2. The palettes of the blocks and the objects differ. Make sure that when you draw your blocks/objects that they use the same number of colours and the same palette.

3. You may be using AGA colours. Backbone does not yet support AGA. AGA stores RGB as values between 0 and 255. Non-AGA stores them as 0 to 15 so Backbone has to divide all the AGA colour values by 16 meaning they don't quite end up as you set them. To solve this, either use Non-AGA screens in your paint program (can be done in Brilliance or DPaint4) or simply ensure your colours are multiples of 16 starting from 0.
e.g. 0,15,31,47...

Problem: When I create the game, I get the error message "Error in Creating the Game, E-mail the Programmer".

Solution: 1. You may be low on memory. Shut down all other programs and windows and try again. Also, disconnected peripherals such as extra floppy disk drives can save some memory.
2. There may be a more serious problem with your project or with Backbone itself. Send all the files (the project, all levels and any object/block files) to
me
and will try
and sort the problem as quickly as possible.

Problem: When I create the game, I get the error message "Internal Files Missing".

Solution: There may be a serious problem with your project or with Backbone itself. Send all the files (the project, all levels and any object/block files) to
me
and will try
and sort the problem as quickly as possible.

Problem: I want bigger levels.

Solution: Edit the
Preferences
and increase the values of X and Y.
It is recommended that you do not exceed 200 for either values and are warned that you could increase the chances of a crash.

Problem: I need more Events/Weapons/Blocks/Conditions/Object/Scenes in my levels.

Solution: Adjust the
Preferences
of Backbone to higher
values.

Problem: Some elements of my game no longer work or are missing.

Solution: The
Preferences
of Backbone may have been reduced.
Reset them to a higher value.

1.30 How to Contact Me or Register

How to Contact Me (Alastair Murray)

1.E-Mail:

malcolm.murray@virgin.net (checked almost every day)
or: pmymagm@pmn1.maths.nott.ac.uk
or: agm96m@cs.nott.ac.uk (checked rarely)

2.Web Pages:

Backbone Support Site--> <http://welcome.to/backbone>
Monolith Designs-----> <http://surf.to/monolith>
Muzzasoft Productions--> <http://welcome.to/muzzasoft>

3.My Address:

Alastair Murray,
2 Warcop Close,
Nunthorpe,
Middlesbrough,
TS7 0QH
England.

What you get when you Register

Registering Backbone will remove all remove ALL limitations from it. It will give you complete updates to ALL later versions of Backbone and also allows the possibility of requesting specific elements to be added to new versions which will allow your projects to exactly as envisaged.

How much is Registration?

Currently, registration is 10 UK Pounds. This price will likely increase as Backbone is further developed. In other currencys that is:

16 US \$
30 German DM
32 NZ \$
100 French Francs

Ideally, a cheque made payable to Alastair Murray is the best method of payment. However, please do not send foreign cheques. Either go to your bank and get an international cheque for 10 UK Pounds or change your money to British currency and send that. If you must send foreign money, please double the amount because I have to pay half of it for the exchange.

Alternatively, a UK Postal Order or UK Cash is also accepted.

How to Register

Send the money to me at my address listed above. Include your name and address. If you have e-mail access then also send your e-mail address. With e-mail users, I will send you a small Key File that will activate all the features of Backbone. If you do not have an e-mail address then I will post you two special codes which you can use in the registration window at the start of Backbone to create your own key file.

I guarantee that I will send all key files within half a week of recieving the money.

Internet Registration

For those who prefer the security of paying over the internet by credit card or those who do not which to use foreign currency I have set up a method of paying for Backbone registration over the internet. This method is a bit more expensive than the standard method to cover costs of the service and currency exchange. This is how much the internet prices are:

25.00 US \$
15.60 UK Pounds
46.90 German DM
50.00 NZ \$
156.25 French Francs

The US \$ amount is accurate. The rest are based on current exchange prices and may change.

To pay over the internet. Point your browser to <http://www.swregnet.com> From there click on 'index' and select 'by platform'. You will see a list of AMIGA shareware titles displayed. Clicking on 'Backbone' will take you to a form where you can fill in the needed details. Your keyfile will then be sent within a day or two.

Distributing Key Files

Note that I can tell whether a game is made in Backbone and exactly who it was made by. This way, I will catch anyone distributing their key files. For anyone I find doing this, I will add a bit to the Backbone for its next release that prevents them from running it.

1.31 The Backbone Mailing List

The Backbone Mailing List

A mailing list has been set up so that users can discuss ways of getting the best out of Backbone and how the future updates should work.

If you don't know how a mailing list works, it's like this: Many people subscribe to the list and then they can send e-mails to it, the address is backbone@onelist.com. Any mail they send to the list will be passed on to everyone else who is also subscribed, this means that conversations can take place and people can help each other with problems.

Feel free to join. To do so go to:

<http://welcome.to/backbone>

or:

<http://freespace.virgin.net/malcolm.murray/backboneml.html>

or:

<http://cs.nott.ac.uk/~agm96m/backboneml.html>

and follow the instructions.

You have to register with a company called OneList first who simply take your e-mail address and computer details. You can then join any number of mailing lists or even create your own.

For further information go to this IRC channel:
#backbone on irc.demon.co.uk OR irc.bt.net
I can't guarantee to be there all the time but I try to pop in when I can.
I can also be sometimes found on #amiga or #amos
My nick is Muzza or Sheridan.

1.32 Example Projects

The Example Projects

There are several demo projects included for you to examine and learn from:

Note that I am NOT a graphics artist and therefore the graphics are quite poor (except for in one where some of the graphics were contributed to from my friend Manuel Moreno)

Each game has an instruction option in the main menu which gives information on what the game does, how to complete it etc.
To quit out of any game hold Ctrl+C.

Tutorial

This game is used in the tutorial section of the menu to show what your created project should look like. The game makes use of the powerhead option in backbone to destroy blocks.

Monkey

This example shows a fairly fast moving platform game. Only basic baddies are used but some of them (the tank) can fire back and some (the snail and flying fish) will chase the player.
The player can also pick up a weapon in some levels.
Menus are used in more detail here. There is a menu screen to allow the player to pick which level they start on.

Micro Fighters

This example is another fast moving platformer. This time the levels make use of auto scrolling. This means the player has to keep moving right to avoid being trapped by the moving screen.

Reality

Apologies to the people who made reality. I made this example using the graphics out of one of the example projects included in the Reality Game Maker. It was just used to see how the two compare.

1.33 How Do I...?

How Do I...?

Contained here is a list of various problems that you may come across when trying to add a complex part of a game.

For help on general running problems see the
Common Problems
section.

...add a weapon to the game?

First ensure that you can have weapons (as set in the screen setup).
Create a gunshot object in the object editor using bullets and explosions
as frames and then allow the player to use this weapon by either creating a
weapon item that the player can pick up or by setting the player to start
with the weapon from the player editor.

Thats it for now! I'll add more here as I get feedback as to what people
are getting stuck with.

1.34 Step By Step

Creating a Project, Step by Step -----

When you create a project of your own, you will always need to go through
some set steps. This is a simple guide so you don't forget anything.

1. Decide upon game details such as Overhead, Side View and what resolution
(usually lowres) and the number of colours to use (usually 16 for fast, 3
for slow moving games)
 2. Using a paint package, draw the background blocks to be used in the
levels, remembering how to use animated blocks, destroyable blocks etc.
 3. Using the same palette as for the blocks, draw all the frames that will
be needed for the player and any baddies or items. The blocks should all be
placed evenly next to each other, with or without frames around them.
 4. Draw any screens or animations needed for intro scenes etc. Also gather
any music and sound effects required and put them in the appropriate
Backbone drawers.
 5. In Backbone, go into the
Image Editor
and cut out all the images of the
player, setting the hot spots correctly and then save them.
 6. Delete all the images and repeat the cutting out process for each baddie
and each item, ensuring that you save them as different file names.
 7. Go into the
Block Maker
and cut out all the blocks saved earlier.
 8. Go into the
Screen Setup
and adjust all the options there.
-

9. Go into
 Edit Player
 and select the player object file saved earlier. Set all the needed parameters and frames.
 10. Go into
 Edit Objects
 and create an object for each baddie, item, or weapon type drawn earlier ensuring you correctly set their parameters.
 11. Go into
 Edit Blocks
 and select the blocks you saved earlier. Set all the parameters for them especially SOLID parameters.
 12. Go into the
 Menu Editor
 if any menu screens are required.
 13. Go into the
 Controls Editor
 and set desired controls.
 14. Go into the
 Scenes Editor
 and create any scenes needed for the introduction, game complete, game over and any in-between levels scenes required.
 15. Go into the
 Sound Editor
 and load any sound effects needed for your game. Save them before exiting.
 16. Go into Edit Details and set the scenes used for introductions etc. Also set any game music to be used and a sound bank as made in the previous step.
 17. Go into the
 Panel Editor
 and create the information bar.
 18. Go into the
 level editor
 . For level 1, ensure that the blocks are set to which ever blocks you wish to use and edit the level. (NOTE: You should Save the project before this)
 19. Create the background of the level using the blocks.
 20. Set the players starting point and add any baddies/items etc.
 21. Set the finishing Conditions for the level.
 22. Set any scenes needed before or after the level and ensure that the "Next Level" option is correctly set.
-

23. Finish editing the level and create any more as needed.
24. Create you game with no optimisation and test it.
25. Adjust as necessary and then repeat step 22 until happy.
26. Finally create the game will full optimisation.

1.35 Main Menu

Main Menu

The main menu is the link to all aspects of the game you are creating. Each game is called a Project. A project must be opened before you can begin creating any of it's aspects. To do this, you use the pulldown menus. Holding the right mouse button allow you to select various options. These are:

Project Menu

New Project : Starts a brand new, empty project.
Open Project : Loads a previously saved project.
Save Project : Saves the current project.
Save Project As : Saves the project under a different project name.
Delete Project: Deletes a selected project and all levels for it.
Iconify: Returns you to the Workbench and moves backbone to a small window.
Closing this window will bring Backbone back.
About: Shows registered information about Backbone.
Quit : Quit's Backbone and returns to the workbench.

Creation Menu

This menu is only needed when a project is about to be 'created'. That is, when all the game is finished and is ready to be put together to form the final product. When you 'create' a project, it will be written to a position of your choice, either hard drive, floppy disk or RAM drive. After it has been created you can then Run it. From this menu you can either choose 'Create and Run' which will create the current project and then auomatcially test run it or simply just 'Create' or 'Run Last Creation' which will automatcially run the game again.

Optimisation is when the game is adjusted so that it fits your parameters as best as it can so the final product runs as fast as possible. The problem with this is that it takes longer to create it. There are four different levels of optimisation from None to Full. No optimisation means the game will be created quickly but will not run at it's best speed (although it will still run at a reasonable rate for most games). Full optimisation means it will take a long time to create but will run at it's best speed. Normally, use No optimisation for checking the game and Full optimisation when it is completed.

Temporary Work

As Backbone uses alot of temporary files, plenty of space is needed to store them. If you have plenty of memory then your RAM disk may be used. Otherwise an area of your hard drive should be selected.

The option "Change Temp Directory" allows you to select where to use.

Arexx

The arexx menu allows various uses of the amigas built in arexx language to make Backbone do all sorts of custom processes. By default, all scripts are held the Rexx directory.

-Run arexx script will allow you to load and run an arexx script.

-Show commands will bring up a special window where you can set up to 10 different scripts to be used at the click of a button.

-Set command and Run command allow you to change and run defined scripts. For example AMIGA+SHIFT+1 will bring up a file selector for you to select what script to be assigned to button 1. From then on, AMIGA+1 will run that script.

-Load/Save Command Set allows you to save the 10 predefined scripts that you have chosen to be automatically loaded into Backbone on startup. Backbone will load the file Rexx/Default.Set as default.

Help

Selecting this menu option or pressing HELP at any time will bring up the Backbone manual. It simply runs multiview to do so.

The Buttons

There are several buttons available which lead to further menus to create the project. These are:

Game Details

- The Game Details menus

Screen Setup

- The Screen Possibilities

Edit Player

- Creating the Player

Edit Objects

- Creating Objects (Baddies,Items etc.)

Edit Blocks

- Creating Background Blocks

Edit Levels

- Creating the games Levels

Edit Scenes

- Creating Scenes

Edit Panel

- Creating the Game Energy,Lives Display

Edit Controls

- Setting up the Game Controls

Edit Menus

- Allowing Menus to give the player choices

Create

- Creating the Final Product

Tools:

Image Editor

- Create and Edit Object Files

Sound Editor

- Create Sound Effects

Block Maker

- Create the Background Blocks

Floppy Disk Tool

- For Projects that Will Span Several Disks

1.36 Game Details

Game Details

From here you can select:

Sound Bank : Select the sound bank made previously with the
Sound Editor

that contains the sound effects to be played in the game. ↔

If you see a button labelled SFX during parts of the editing,
clicking on it will allow you to select a sound effect to play
at the appropriate moment.

Introduction : Select a scene previously made with the Scene Editor to play at
the beginning of your game.

Title : Select a scene previously made with the Scene Editor to play during
the title screen. This will normally consist of just music playing
and a screen being displayed until the player carries on.

Completed : Select a scene previously made with the Scene Editor to play when
the player finishes the entire game.

Game Over : Select a scene previously made with the Scene Editor to play when
the player loses all their lives.

1.37 Screen Setup

Screen Setup

From here you may select various options to allow the actual game play to be as visualised.

The options selected here dictate how the game screen will appear.

Screen Modes:

There are four screen modes available. Lowres and Hires, Interlaced or not. Lowres is strongly recommended for anything but the least demanding or games as higher resolutions may have speed problems.

Colours:

8,16,32 or Extra Half Brite Mode can be used. Be aware that some resolutions do not allow more than 16 colours.

Block Size:

Blocks are the tiles that are used to create the background to levels. They generally consist of images of solid walls, solid floors, animated blocks and general background blocks. They have to be a certain size so that they will all fit together to make a level. They can be either 16x16 pixels in size or 32x32 pixels. Note that when using either Hires or Lowres Interlace screen modes the blocks will not be perfect squares.

Full PAL Screen:

Click this button to create a default screen setting that will fill up a PAL display.

Full NTSC Screen:

Click this button to create a default screen setting that will fill up an NTSC display. This will leave a black border on PAL machines but has the benefit of being playable by any amiga owner in the world.

Recommended Settings:

Click this button if you are a novice to backbone. It defaults to a NTSC 16 colours, lowres screen. This allows games a reasonable speed on low grade computers. If you are writing the game to run on high spec AMIGAs then more colours will be OK.

Screen Size:

From here you may change the exact size of the screen display.

Screen Shots:

There is a check box here that, if activated, means that during your game you can press the 'Control' key and a screen shot of the current play will be saved to the RAM disk. Useful for showing off your games. Be careful not to hold down the key for very long.

Ctrl+C Quits Game:

If this option is selected, you can press Control+C at any point to quit your created game. However, if you have created a menu that allows an

option to quit then you may want to disable the Ctrl+C option.

Cheat Mode:

If you select this box, it allows you to use two special keys to cheat in your game. This is mostly used for when you're testing your game. During your game, pressing 1 on the numerical pad will give the player full lives and energy whereas pressing 2 will complete the current level.

Scrolling:

There are two buttons which allow you to specify the type of scrolling used. You can use Chunky or Slide scrolling. Chunky is where the screen moves in blocks. This has the effect of sometimes looking unsmooth but moves very speedily. Slide scrolling moves more smoothly but slightly slower. A recommended method is to set the vertical scroll to chunky and the horizontal to slide.

View:

Perhaps the most important initial setting for any project. You must first decide whether the game is to be viewed from above (as in games such as The Chaos Engine) or from the side (as in platform games such as Mario Brothers or Rainbow Islands)

Gun:

You have an option of how the weapons and bullets in your game will work.

1. No Weapons. The game will contain no guns or bullets whatsoever.
2. Weapons Enabled. The game can use weapons of different types as specified when you create a Weapon Shot object.

1.38 Edit Player

Edit Player

From here, all the details about the actual player can be adjusted.

First of all the set of images used to animate the player (called an object file) must be loaded by clicking on the PICK FILE button. To create these images, uses the

Image Editor Tool
from the main menu.

Moving Speed:

Select the number of pixels per frame that the player will move when the joystick is pushed. For side viewed games, this value only applies to left and right movement but for overhead viewed game, up and down movement is also involved.

Typical values: 4 to 12 depending on the size of the player.

Acceleration:

For side viewed games only. If you want your player to "slide" as in games such as the Mario series then this is the option to use. Setting to zero or to a value more than the moving speed will have no effect but setting it somewhere in between (typically about one third of the value of the move

speed e.g. Moving Speed=9 so Acceleration=3) gives a feeling of acceleration to the player. They will have to speed up to reach their full speed and will not stop instantly when letting go of the joystick.

Falling Speed:

For side viewed games only. The number of pixels per frame that the player will move downwards when not standing on a solid platform.

If this value is set to zero then the player will not ever fall. This allows for such games as sideways spaceship shooting.

Typical Values: 8-16

Jumping Speed:

For side viewed games only. The number of pixels per frame that the player will move upwards when the player jumps.

If this value is set to zero then the player is unable to jump.

Typical Values: 8-12

Max Jump Height:

The maximum number of pixels that the player can jump in height.

So if using 16x16 sized blocks. A max jump height setting of about 56 would allow the player to jump up three blocks but no more. You should always set this value a few more pixels than you would seem to think to allow for various other variables.

Climb Speed:

For side viewed games only. The number of pixels per frame that the player will move upwards or downwards when the player climbs a ladder.

If this value is set to zero then the player is unable to climb any ladder.

To set up ladders, use the

Block Editor

.

Typical Values: 8-12

Crawling Speed:

For side viewed games only. The number of pixels per frame that the player will move left or right when the player is crawling by pushing the joystick down is pushed.

You must set the "Can Crawl" option to use this value. If it is set to "cannot crawl" then this value has no use.

If this value is set to zero then the player may not crawl but will only duck without moving.

Typical values: Either the same or a bit smaller than normal moving speed.

Crawling Height:

When a player is crawling, its height will be reduced so that it may crawl underneath platforms which would previously get in the way. The crawling height value is the height of the player in pixels when crawling.

So for example if the player image is about 64 pixels high normally, it may half to 32 when crawling. You should also take into account block sizes when selecting the height. If for example, you are using blocks of size 32x32 and want the player to be able to crawl under one block then you should set the value to 31 or less.

Energy and Lives:

These values decide on the amount of energy and the number of lives your player has.

Player Height and Width:

As for the crawling height, you must give the size of the player (as measured from the hot spot, see the

Image Editor
for details)

This value will be used to calculate whether the player can pass under solid platforms above them and also when jumping.

The width is used to calculate how close to a solid wall the player can get without having to stop.

Auto Grab:

This option will automatically grab reasonable Width and Height sizes for the player. Note that you must have first selected the object file and selected the correct frames to use this option.

Powerhead:

For side viewed games only.

This is an option which is seen in games such as Super Mario Brothers or the much better Great Giana Sisters. It means that when the player jumps up and hits a certain type of platform, it will be either destroyed or changed into another type of block.

If Powerhead mode is ON then the player always has this ability, otherwise there is the option to create collectable items which will give the player this ability or even take the ability away. See the Object Editor for more details on collectable items.

After Death:

When a player dies, but still has some lives left, there are two options in what happens next. Either the player can "continue" in which case the game goes on except the player is invincible for an ammount of time or the player can "Restart the Level" in which case the level is ended and the player must do it again from the beginning.

Invincability Timer:

Once the player dies and if the After Death setting is set to Continue then the player will be invincible for a set number of turns as changable here.

Typical Values: 20+

More:

Pressing this button will flick between two screens to allow you to edit further details of the player.

Start Weapon:

Use this to select the weapon that the player will start the game as created previously using the
object editor

Selecting "None" means the player will start without any weapon.

Selecting "Not Defined" means that the weapon has not been created and you should make another choice.

Shot Height:

For side viewed games only.

Every time your player fires a weapon (assuming they have one) the bullet for it must appear at a position near the player (at the end of its gun) so that it can move away in a realistic manor.

Shot Height tells the computer how far up from the players hot spot (usually its feet) the bullet should come from. This value is usually about half the value of the "Players Height".

Shot Crawl Height:

Exactly as for "Shot Height" except used when the player shoots their gun while crawling.

Can Crawl/Duck:

For side viewed games only.

This option can either be set to "Can" or "Cannot". If set to "Can" then if the player pushes down then they can duck or crawl depending on your value set in the "Crawl Speed" field.

If set to "Cannot" then pushing down has no effect.

Set Frames:

The set frames button allows you to specify which animation frames are used for which parts of the player.

[Click here to see details of the frame selection process.](#)

For Side Viewed Games

For Above Viewed Games

1.39 Players Frames Settings - Side View

Player Frames For Side Viewed Games

From here you select the exact frames to be used from the previously selected object file created with the Image Editor Tool

.

NOTE: All these frames should be of the player facing RIGHT. They will automatically be switched when the player faces left. Also note that the death frames remain the same whatever weapon the player has and whether or not it has powerhead so they only need setting once.

There are several radio buttons detailing possible actions for the player. If your player cannot do these actions (e.g. they cannot jump) then you do not need to set them.

These include:

Still: The set of frames where the player is still. Normally this will just be one frame but the option is there for animations.

Moving: The set of frames in which are cycled when the player moves.

Jumping: The frame when the player jumps.

Falling: The falling frame.

Dieing: The set of frames which are moved through when the player dies.

Climbing: The set of frames in which are cycled when the player climbs a ladder.

Crawling: The set of frames in which are cycled when the player crawls.

Shooting: The set of frames showing the player firing. If you have not

selected a weapon (see below) then this option will not be available.
 Five direction buttons will be in the top right hand corner. These are used to set the frames for the player when it is firing Up, Up and Right, Right, Down and Right, Down. For firing left, these images will automatically be flipped. If the weapons to be used will only fire left or right then you do not need to set the frames for up/up+right/down+right/down.

Powerhead On/Off:

If your game uses the Powerhead option at all then you must set the same frames that will be used for when the player has the powerhead, this is done by selecting the Powerhead On option and changing all the frame values. You may well want those frames to be the same as without Powerhead, if so you can use the Copy from Default option or just manually set all the frame numbers.

Weapon:

You must set different frames for when the player is holding a weapon. If you have no weapons in your game then you do not need to adjust these, otherwise you must cycle through all the available weapons and set the frames. If you do not wish them to be any different from normal then use the Copy From Default option. This option allows the player to see for themselves what kind of weapon they have at a glance.

Copy From Default:

This will copy all the values for all the frames of the original frames to the current set of frames.
 e.g. If you are editing the frames for the player holding Weapon 2 then clicking on this button will copy all the frames from the player without any weapon.

SFX:

Sound can also be selected from this menu to be played at appropriate times. For example, whilst editing the Still Frames you can select what sound effect (from the sound bank selected in the
 Game Details
 Section)

to play when the player is hurt. Similarly when editing Jumping frames, you can decide on a sound to play when the player jumps.
 When editing Powerhead frames you can pick a 'powerhead' sound. This is the sound that will be player when the player uses its powerhead ability to destroy a block. The rest are self explanatory.

The View boxes allow you to view the frame (or animation) in the screen below. Clicking on the screen will move the image so that it is viewable.

1.40 Players Frames Settings - Above View

Player Frames For Above Viewed Games

From here you select the exact frames to be used from the previously selected object file created with the

Image Editor Tool

.

Moving Direction:

There are several radio buttons detailing possible directions for the player. There are 8 directions the player can face (up,down,left,right and the diagonals)

For each direction you must set a still frame and the walking frames.

The View boxes allow you to view the frame (or animation) in the screen below. Clicking on the screen will move the image so that it is viewable.

Weapon:

You must set different frames for when the player is holding a weapon. If you have no weapons in your game then you do not need to adjust these, otherwise you must cycle through all the available weapons and set the frames. If you do not wish them to be any different from normal then use the Copy From Default option. This option allows the player to see for themselves what kind of weapon they have at a glance.

If you select weapon zero (no weapon) then you can set the death frames for your player, otherwise you must set the shooting frames for that particular weapon.

Copy From Default:

This will copy all the values for all the frames of the original frames to the current set of frames.

e.g. If you are editing the frames for the player holding Weapon 2 then clicking on this button will copy all the frames from the player without any weapon.

SFX:

Sound can also be selected from this menu to be played at appropriate times. For example, whilst editing the Still Frames you can select what sound effect (from the sound bank selected in the

Game Details
Section)

to play when the player is hurt. The rest are self explanatory.

1.41 Edit Objects

Edit Objects

Objects is a term used for all sorts of thing in the game.

Objects can be monsters (good or bad) or items for example, anything that uses image files.

NOTE: At this early stage of the development of backbone there are a sever lack of object types. Many more are planned including inteligent creatures and goodies who will help the player along with all manor of other creatures.

For each object (identified by its Object Number) you must select a name for it by clicking on the "NEW NAME" button.

Objects can be added or deleted by clicking on the appropriate buttons. You must also select an object file for the object as created using the

Image Editor Tool

.

The Object Type can be selected using the cycle tool.

This will cycle between:

Patrolling: A simple back and forth moving baddie.

Collectable: An item that can be picked up by the player by touching it.

Clicking on the "Edit Details" button will give you a further menu from which the exact details of the object can be adjusted.

Click on the following buttons to see instruction for the details of these item types.

Patrolling Enemies

Collectable

Weapon Shot

Chasing Enemies

1.42 Editing a Patrolling Baddie

Editing a Patrolling Baddie

Moving Speed, Falling Speed, Baddie Height, Baddie Width, Jump Speed and Jump Height work just as for the player in the

Edit Player
section.

Auto Grab also works as in the player section, grabbing the width and height of the object from the defined frames.

Walking/Falling/Flying/Jumping:

For side viewed games only: Patrolling baddies can have set options to them, some will not fall off ledges, some will. This is how you set which option to use. If a baddie encounters the edge of a ledge or platform and they are instructed NOT to fall off it then they will turn around. Alternatively, your baddie can be flying in which case ledges have no consequence whatsoever.

Baddies that are not flying can have the option of jumping, when they encounter a wall they will attempt to jump over it.

For overhead viewed games, patrolling baddies will simply stick to the approximate area that they were created in, randomly moving around it.

Normal/Ghost Mode:

If Ghost Mode is set then the baddie will go through walls. Note that they will not fall through floors however.

When Hit Player:

When the baddie touched the player there are two options of what can happen. The player can either have some of its energy zapped (the exact amount detailed in the "Energy To Zapp" box) or the baddie can cause "Instant Death" to the player.

Energy:

The amount of energy the baddie has. When this reaches zero the baddie will die.

How It Dies:

There are four options of how the baddie can be killed.

1. When Jumped On - The player must jump on the creatures head (each time will take off one point of energy) to kill it. NOTE: Setting the Baddie Height field is vital for this to work.
2. Indestructable - The baddie can never be killed. This can be used for inanimate object such as falling blocks which will kill the player but cannot be destroyed.
3. When Shot - The baddie will die when it is shot by the player or a friendly object.
4. Jump On or Shot - The baddie will die if it is either jumped on OR shot.

After Death:

There are currently three options available when the baddie is killed.

1. Disappear - The baddie will go through its death frames and then completely disappear.
2. Stay Forever - The baddies last death frame will remain forever on the level where it died. For this you should ensure the last frame is something that will look alright. i.e. a small corpse or puddle of blood. This method has the disadvantage of slowing the game down if too many dead creatures are laying around.
3. Stay While On Screen - The baddies last death frame is "pasted" onto the screen. This is like the Stay Forever option except once the player moves off the screen the corpse will disappear.
4. Change to another Object - The baddie will change into another previously defined object. This could be another baddie (so when killed, it would morph into another sort of creature) or an item (so it drops a special item when killed)

Set Danger Frames:

This brings up an option window where you can select which frames of the baddie are dangerous to the player. Normally this will be all the frames but sometimes you may want the baddie to only be dangerous at certain times.

Weapon:

This option will bring up a further window where you can decide upon options to do with the

Baddies Weapon

.

Set Frames:

This works as in the

Edit Player

section.

1.43 Editing a Collectable Item

Editing a Collectable Item

Frames:

"First Frame" and "Number of Frames" just indicate the frames to cycle through when the item is just sitting there waiting to be collected.

If you just want a single frame for the item, set "Number of Frames" to 1.

"First Collected Frame" and "No. of Collected Frames" indicate the frames to go through when the item is picked up by the player. If you just wish the item to disappear then select the same frame for "First Collected Frame" as you did for "First Frame" and set "No. of Collected Frames" to 1.

There are various types of items you can select to dictate what the item actually does when collected:

Powerhead On: Gives the player the powerhead ability to allow them to destroy certain types of platforms by jumping at them with their head. See

Edit Player
section for details.

Powerhead Off: Takes away the powerhead ability.

Extra Energy: Gives the player some more energy, the amount dictated by the "Amount" field.

Extra Lives: Gives the player one or more extra lives.

New Weapon: Gives the player the weapon number specified as previously created with the
object editor

.

Add To Score: Increase the player score. (NOTE: Score is not actually implemented yet, sorry! But you can still use this item, it just won't do anything.)

1.44 Editing a Weapon Shot

Editing a Weapon Shot

Moving Speed works just as for the player in the
Edit Player

section.

When Hit Object:

When the weapon shot touches an object (e.g. the player or a baddie if the player fired it) there are two options of what can happen. The object can either have some of its energy zapped (the exact amount detailed in the "Energy To Zapp" box) or the weapon can cause "Instant Death" to the object.

Range:

The distance in pixels that the weapon shot will travel before exploding.
Typical Values: 120-200

Weapon Number:

Each weapon must have a unique identification number assigned to it. A value will automatically be entered in here but it may be manually adjusted as well.

When telling Backbone about this weapon (e.g. saying that the player has this weapon to start with) you will refer to it by this number.

Shot Delay:

The number of game frames before the player (or creature) with the gun can fire another shot. A value of zero means a continuous burst of shots.

Typical Values: 1 for a fast reloading gun.
20 for a very slow reloading gun.

Travel Through Wall:

This can either be set to "CAN" or "CANNOT" and simply decides whether the weapon shot can travel through walls or not. If not then it will explode upon contact.

Move When Firing:

This can either be set to "CAN" or "CANNOT". If they CAN then the player will continue to move when firing if the player presses fire. If they CANNOT then the player will stop moving for the instant they press fire.

Fire When Jump or Fall:

Only applicable for side viewed games. If your player cannot jump or fall (i.e. it is a spaceship based game) then this does not matter, otherwise setting it to CAN means that the player can shoot their weapon whilst jumping or falling, otherwise they must be on solid ground to do so.

If you are using "Any Direction" weapon type as set in the screen setup section then this option will not be available as there is no way of firing down and left while jumping up and right for example.

Type:

Sets the type of the weapon. Currently the options are:

Left and Right Only: (Only used in side viewed games) The player (or baddie) will only be able to fire the weapon left or right.

Any Direction: The player (or baddie) will be able to fire the weapon in any direction.

Set Frames:

This works as in the Edit Player section.

1.45 Editing a Chasing Baddie

Editing a Chasing Baddie

Moving Speed, Falling Speed, Baddie Height, Baddie Width, Jump Speed and Jump Height work just as for the player in the Edit Player section.

Walking/Falling/Flying/Jumping:

For side viewed games only: Chasing baddies can have set options to them, some will not fall off ledges, some will. This is how you set which option to use. If a baddie encounters the edge of a ledge or platform and they are instructed NOT to fall off it then they will turn around. Alternatively, your baddie can be flying in which case ledges have no consequence whatsoever.

Baddies that are not flying can have the option of jumping, when they encounter a wall they will attempt to jump over it.

Normal/Ghost Mode:

If Ghost Mode is set then the baddie will go through walls. Note that they will not fall through floors however.

(un)Limited Sight:

Normally, a chasing baddie will have a limited sight meaning that it can only see the player if it is within their line of sight and will then move towards it. However, if you set this option to "Unlimited Sight" the baddie will always know where the player is and will always move towards it.

When Hit Player:

When the baddie touched the player there are two options of what can happen. The player can either have some of its energy zapped (the exact amount detailed in the "Energy To Zap" box) or the baddie can cause "Instant Death" to the player.

Energy:

The amount of energy the baddie has. When this reaches zero the baddie will die.

How It Dies:

There are four options of how the baddie can be killed.

1. When Jumped On - The player must jump on the creature's head (each time will take off one point of energy) to kill it. NOTE: Setting the Baddie Height field is vital for this to work.
2. Indestructible - The baddie can never be killed. This can be used for inanimate objects such as falling blocks which will kill the player but cannot be destroyed.
3. When Shot - The baddie will die when it is shot by the player or a friendly object.
4. Jump On or Shot - The baddie will die if it is either jumped on OR shot.

After Death:

There are currently three options available when the baddie is killed.

1. Disappear - The baddie will go through its death frames and then completely disappear.
2. Stay Forever - The baddies last death frame will remain forever on the level where it died. For this you should ensure the last frame is something that will look alright. i.e. a small corpse or puddle of blood. This method has the disadvantage of slowing the game down if too many dead creatures are laying around.
3. Stay While On Screen - The baddies last death frame is "pasted" onto the screen. This is like the Stay Forever option except once the player moves off the screen the corpse will disappear.

Set Frames:

This works as in the
 Edit Player
 section.

Set Danger Frames:

This brings up an option window where you can select which frames of the baddie are dangerous to the player. Normally this will be all the frames but sometimes you may want the baddie to only be dangerous at certain times.

Weapon:

This option will bring up a further window where you can decide upon options to do with the
 Baddies Weapon
 .

1.46 Editing a Baddies Weapon

Editing a Baddies Weapon

From this window you can decide if the baddie is to have a weapon, and if so, the details of using the weapon.

First, a

Weapon Shot
 must have been created as a standard object.

You can cycle through which weapon to use on the slider bar. A value of zero means the baddie has no weapon. You can also set Aggression and Accuracy of the baddie. Aggression means how often it will fire and Accuracy means how good the shot will be aimed at the player.

You can also set the Shot Height which works as when

Editing the Player
 .

This option is not needed in overhead games.

There is an option to keep the baddies a certain distance away from the player. Normally they will just move towards the player, firing as they move until they reach it, however if you activate the 'Keep Player Away'

then you can set a certain distance that the baddie will keep from the player. This means the baddie will simply run to the player until it gets in range and then will stop and shoot. NOTE: For this to be effective you must set the distance value to be just under the range of their weapon. This option is only available (and useful) for chasing baddies.

The final button is Edit Firing Frames and is used to set what frames the baddie will go through when firing a shot. Note that generally a weapon has a delay meaning that it will only take one frames to fire the weapon.

1.47 Edit Blocks

Edit Blocks

From here you can edit all the blocks used in the levels of your game. You can have several block "banks" which use different images to allow variety throughout the levels.

First you have to pick a block file as created using the
Block Maker

You can edit various properties of all the blocks including:

Solidity

: Whether the player can pass through blocks.

Animations

: Background blocks can be animated.

Foreground

: Some blocks can be displayed OVER objects so the player passes behind them.

Destroyable

: Some blocks can be destroyed through the players actions.

Danger Level

: Some blocks can be dangerous to the touch.

1.48 Blocks Solidity

Blocks Solidity

Clicking on a block with the left mouse button will cycle its type. Clicking with the right mouse button will reset it.

Side View:

Cycles through 4 types of block.

1. Completely Solid. Nothing may pass through this block or jump or fall through it. Seen as a square with diagonal lines going through it.
2. Platform. Nothing may FALL through it but the player may jump through it. Seen as a rectangle at the top of the block with diagonal lines through it.
3. Ladder. Can be climbed but the player will also fall through it. Seen as a series of lines shaped like a ladder.
4. Top Ladder. Used as the top block to a ladder. Can be climbed but the player will NOT fall through it if it walks over the top of it. Viewed as a mix between the ladder and a platform.

Above View:

Can either be solid or not.

1.49 Blocks Animations

Blocks Animations

Clicking on a block with the left mouse button will select a block to be animated. You will then be asked for the number of frames the block will have, these can be between 2 and 4 and must be consecutively placed. You will next be asked whether the animation will be ordered or random. Ordered means that the animation block frames will be cycled through in order forever, Random means that frames will be displayed in a random order.

If a block shows an "S" on it, that means that you have previously defined the block to be solid. This is simply used for reference so you know what block does what.

Clicking with the right mouse button will reset it.

1.50 Blocks Foreground

Blocks Foreground

Clicking on a block with the left mouse button will select a block to become a foreground block. You will be asked which block should be placed over it.

The foreground system works like this. The original block is always drawn as part of the background and the block selected to be drawn over it is always placed over it on top of any foreground objects. This means that the foreground block (the one to be placed over it) must have a background colour of zero if anything behind it is to be displayed.

If a block shows an "S" on it, that means that you have previously defined the block to be solid. This is simply used for reference so you know what block does what.

Clicking with the right mouse button will reset it.

1.51 Blocks Destroyability

Blocks Destroyability

Clicking on a block with the left mouse button will cycle its type.
Clicking with the right mouse button will reset it.

If a block shows an "S" on it, that means that you have previously defined the block to be solid. This is simply used for reference so you know what block does what.

NOTE: When a destroyable block is destroyed by any method what actually happens is the block is moved to the next consecutive block. So if block number 12 was blown up, it would be replaced by block number 13. If you wanted the block to be replaced by an empty space then block 13 should be empty.
This method allows for several effects including blocks that need to be blown up several times to be destroyed.

Side View:

There are 3 types to cycle through:

1. Normal Destroyable. Can be blown up by various methods.
Seen as a box with diagonal lines across it.
NOTE: Not yet implemented in this early version of Backbone.
2. Powerhead Destroyable. Can be destroyed by a player with the powerhead ability when they jump up and strike the block from below.
Seen as a rectangle at the bottom of the block with diagonal lines through it.
3. Totally Destructable. Can be destroyed through either of the above methods. Seen as a mix between the above descriptions.

Above View:

Blocks can either be destroyable (by being blown up) or not as powerhead has no use for games viewed from above.

1.52 Blocks Danger Levels

Blocks Danger Levels

Clicking on a block with the left mouse button will cycle its type.
Clicking with the right mouse button will reset it.

If a block shows an "S" on it, that means that you have previously defined the block to be solid. This is simply used for reference so you know what block does what.

There are 4 types of block (2 for above viewed games)

1. Hurt. The block will take off energy upon contact. Shown as a small
-

- cross on the editor.
2. Death. The block will kill the player upon contact. Shown as a large cross on the editor.
 3. Hurt When Fallen On. The block will take off energy when the player falls on it (side viewed only). Shown as a small cross with a base to it on the editor.
 4. Death When Fallen On. The block will kill the player when they fall on it (side viewed only). Shown as a large cross with a base to it on the editor.

1.53 Edit Levels

Edit Levels

From here you can edit all the levels that your game will use. Each level must use a block bank as created by the Block Editor these are what will make up the background to your level.

The pulldown menus allow you to save or load levels from other sources (e.g. other projects) Note that unless they use the same blocks and objects then the levels will be completely corrupted.

Click on the Edit Level button to begin editing. When you finish editing a level you will be asked whether you wish to save it or to lose it.

When editing the level:
 Use the cursor keys or press the arrow buttons to move the map. Shift and cursor will move the map faster.
 The 9 buttons labelled Position Map will allow you to move the position that you are viewing the map from (e.g. top left etc.)

The X and Y fields show the position of the viewpoint on the map in Block coordinates.

The left or right mouse buttons on the map will paste selected blocks onto it. To pick new blocks either press the "Pick Block" button or press space. Selecting the block with the left mouse button will mean the left button will be used to paste it and the same with the right button. You can select chunks of blocks together by holding the mouse button as you click on them.

Scenes:
 You can set the project up to play a previously designed scene using the

Scene Editor

at different points. You can play them before the level or after the level and more than that you can have different scenes for both whether the player has completed or failed the level. This allows you to design games which will take unique paths with unique storylines depending on how well the player is doing.

Next Level:

Here you can select which level the player will go onto next if they either

complete it or fail it. For the majority of games, the completed field will be set to the next level and the failed to the current level so the player has to try again if they die.

Start Pos:

This button allows you to select where the player should start the level. The computer will warn you if you try to place the player with a solid wall.

Conditions:

The computer has to know what it is that the player has to do on the level to complete it. There can be one or more conditions but if you do not set any then the level will be instantly completed by the player as soon as they start it.

Conditions can be:

1. Kill Baddies. The player must kill a certain amount of a specific object type. e.g. If object type 2 was an alien, you could set the condition to be Kill 10 Baddies of Type 2. Then the player would have to kill 10 aliens to continue.
2. Collect Items. Works as Kill baddies except instead of killing them, you are collecting them.
3. Get To. The player must get to a certain area of the map (usually at the end of the level) Click on the "Set Area to Get To" button will set up this area. NOTE: The player only has to have got to the area at ANY point throughout the level, not just at the end.
4. End At. As Get To except that the player must finish the level at this position having completed all previous conditions first.

Objects:

Clicking on this will bring up a sub-menu containing the options:

Add Object:

When you either press the "Add Object" button or press N (for New) you will be able to add objects to the level that have been previously made with the

Object Editor

First select the object you wish to add and press on OK.

Next you will have to select any needed details about the object.

e.g. If it is a monster then you must select which way it is facing.

Finally you must place it on the map.

Delete Object:

This will cycle through all the objects on the screen. When the one you want to delete is flashing, press OK to erase it.

Events:

Note: You can change the maximum number of events that Backbone has by editing the Preferences of Backbone.

Clicking on this will bring up a sub-menu containing options for creating events. You will also be able to view event positions on the map as crossed squares. These are your options:

Add Event:

You can set up certain events which occur when the player moves over specified blocks. When adding an event, you must first specify which

block it is that will activate the event.

Also select whether the event is to be automatically activated when the player passes over the area or whether the player must manually activate it like a button or switch. (The player does this by pressing Space)

Next you must decide what the event is to do from these options:

1. Teleport the player. You must select where the player will be teleported too.
2. Change a block. You must select the position off the block to be changed and then select the block to change it to. This option can be used for opening up secret areas, or opening doors etc.
3. Add an object. You must select the object and position for it to appear.
4. Remove an object. You must select the object that is to disappear.
5. Play a scene. You must select the scene to be played.

Del Event:

This will cycle through all the events on the screen. When the one you want to delete is flashing, press OK to erase it.

MUSIC:

Each level can have it's own unique music being played. Clicking on "Music" will bring up a menu telling you what music is currently selected and giving you the option of deleting it or changing it. You must select Protracker music files here.

BACKGROUND:

Each level can have it's own special background. They can be plain colour zero using the colours you used with your blocks (usually Black) or you can set up a special colour gradient effect called a rainbow or copper effect. There is a choice of colours that your copper can be. They can add the effect of a sky to the level. You can also decide how the gradient will progress. e.g. Whether it will shade from Light to Dark or vice versa.

FILL MAP:

This option will fill a part of the map with the currently selected block. Every block in a rectangle between the very top left of the entire map and the bottom right of your current view screen will be changed to this new block. It is advised that you don't overwrite areas of the map which are not really required in your level as Backbone can only tell where the right and bottom borders of your level is when it comes to empty blocks.

AUTO SCROLL:

From this menu you can decide on how the scrolling will work in your level. Normally the scrolling will simply be set to follow the player but in some games such as space shoot em ups you may want to make the scrolling so it continuously moves in one direction. You can do this by setting the required speed and direction. A speed of 2 would be quite slow whereas a speed of 16 would be very fast. However, larger values will make for more jerky scrolling.

If the player is forced into a solid block by the scrolling you can decide whether or not the player is killed instantly or merely loses energy. There is also an option to allow backwards scrolling or not. This would mean that if the player starts moving in a backwards direction to that of the scrolling, the scrolling will stop moving forwards and will move backwards.

There is one further option. You can decide whether or not the player is

moved at the same speed as the scrolling or not. For space shoot em ups the answer is usually yes but for platformer game that want to force the player to keep moving, this answer would usually be no.

When your level is complete, close the window.

1.54 Edit Scenes

Edit Scenes

From here you can create the scenes played during the introduction, in between levels, when you complete the game and in other situations.

A complete scene is made up of several individual scene numbers. So for example if you wanted a short introduction scene that went like this:

1. Some music starts playing.
2. A still logo picture is displayed for a short time.
3. An introduction animation is played.
4. Music Ends.

Then you would create 4 scenes each linked to the next one which would do exactly as detailed. The last scene would have no link to let Backbone know that it is the end of the complete scene.

The scene editor is made up of a list view of all the scenes. This list can be scrolled up and down to view any scene. The view shows the scene number, name and the number of scene it is linked to (if any).

Clicking on a scene will select it and you can then edit it's features. It's name can be changed and type from the cycle gadget. You can also use the slider bar to change the next scene to link to. If you select zero, this means the scene will end. Clicking on the 'Auto Next' button will set the next scene value to the one after the current one for convenience.

You can also manipulate the positioning of the scenes in the list by using the Up and Down buttons to relocate them. Delete will erase the current scene. Note that if you move a scene that has been selected elsewhere to be played (e.g. as the introduction) then you must adjust this.

Clicking on Edit Details will allow you to decide on the precise aspects of the scenes. Most noticeable the files that will be used (e.g. a Picture file, Animation file or Music file)

Scene Types:

1. Show IFF Picture. Select an ILLM picture to be loaded and displayed.
NOTE: Currently AGA pictures can NOT be used. (this should change in later versions of Backbone)
In the Edit Details section you can select whether the picture should fade in, fade out or just appear and disappear. NOTE: HAM Pictures can not be faded in.
You can also set the number of seconds that the picture should be shown for. Finally, the "able to skip scene" cycle gadget specifies
-

whether the player pressing fire on the joystick should skip the scene or whether the player is forced to watch it.

2. Show Animation. Select a standard ILBM type 5 animation (as used by Dpaint and most other art packages) to be played. You can select the number of seconds to wait each frame (which will normally be a fraction of a second) and whether or not the player can skip the scene. Also select the number of seconds that the animation will pause when it is finished before closing itself.
3. Play Music. Loads and plays a PROTRACKER music file. No further details required.
4. Stop Music. Stops any music playing. No details or file required.
5. Show Menu. This allows you to display a menu screen previously defined using the

Menu Editor

to give the user a choice of whatever you want.

1.55 Edit Panel

Edit Panel

The panel is what is used to display the details of your game. e.g. Lives, energy, objects collected etc.

1. None - No panel at all is used.
2.
 - Simple Text
 - The panel contains simple text information telling you the amount of energy etc.
3.
 - Picture Based
 - This uses an IFF picture background and displays the game details over it. This mode allows you to use graphics and bars as well as text.

You can also select where the panel is displayed, above or below the main play screen. Usually, below looks better but not always. It is essential (most noticeably so with above panels) that you get the sizes correct. For example if your play screen is 200 pixels in height, you want the panel to be 56 pixels maximum (for a PAL screen) otherwise some of the screen will be obscured.

Click on Edit Details to set the particulars of your panel.

1.56 Simple Text Panel

Simple Text Panel

This mode will show a simple black panel below the main playing screen with white text used to display the details.

You can have the panel in Lowres or Hires. Lowres will only allow about 40 characters across the screen whilst Hires will allow 80 but the text will be smaller.

You can fill in the text box exactly how you wish the text to appear. Special characters starting with ! are used to represent individual values.

e.g. !e = energy
 !l = lives
 !s = score
 !a = ammo

So typing in the text box "LIVES=!l" will show in the panel:
 LIVES=10 (or however many lives the player has)

1.57 Picture Based Panel

Picture Based Panel

A picture based panel consists of a picture along with a series of 'elements' to represent different characteristics of the game (e.g. Energy Left)

The first thing that must be picked is Background IFF Picture. This picture should be of a reasonable size. For example, if your game screen is 320x200 then you may want the panel to be 320x50 (so that the game screen and the panel will fit on the whole screen together). If your paint package doesn't support smaller screens then you can simply cut out panel and save it as a brush.

Now you have to add elements to your panel. You can cycle through each element using the slider gadget. Each element can display a different aspect of the game. e.g. Energy, Lives, Score, Ammo or Weapon.

Each element can also be displayed in a different manor. These are:

Text

- Simple text using a specified font can be written to the panel.
 Commonly used for Score or maybe Lives.

Simple Bar

- A simple coloured bar can be used to represent the value.
 These are commonly used for Energy or Ammo.

Object Bar

- This uses objects cut out using the
 Image Editor

and displays them as a bar. For example, you could have an image of the players face and an image of a gravestone. Then using an object bar you could display the number of lives left as a face and every time the player loses a life, a face is replaced by a gravestone.

Still Object

- This mode using a still image that changes as the value specified changes. For example you could have a picture of the players face at various point from changing from being healthy to being a skull. As the player loses energy, the face on the panel could change.

For each of these modes (except Simple Bar) you are required to pick a file. In the case of Text mode, you must pick a font to be used. In the case of Object Bar or Still Object you must pick an object file saved previously using the

Image Editor

.

1.58 Edit Panel Text Element

Edit Panel Text Element

When editing a text element, you must set the Maximum Value. This is the highest value that will be shown. So for example, if the player can have a maximum of 4 lives, enter 4 in this box.

In the text box you must enter what text is to be displayed. Use the character ! to denote where the value must be placed. Using an empty box except for ! will just show the value and nothing else.

You also have the option represented by a cycle bar of having the value displayed as actual or percentage. Actual is obvious, 4 lives will be displayed as 4 but with percentage the value is shown as a percentage of the maximum value. For example if you enter the max energy value to be 40 and the player has 10 then using '!%' in the text box will display 10% on the panel.

You can also set the colour of the text using the Set Colour button and the position on the panel using the Position button.

1.59 Edit Panel Simple Bar Element

Edit Panel Simple Bar Element

When editing a bar element, you must set the Maximum Value. This is the

highest value that will be shown. It is used for calculating the percentage value. So for example, if the player starts with 20 energy you would probably want to set this value to 20 unless they are likely to collect energy that will take them above this value.

You must also set the direction in which the bar moves, Up, Down, Left or Right. Generally they are right or up. For example an energy bar will probably want to move to the right as it fills up and ammo move up.

You can also pick the colour of the bar.
When positioning the bar, you must drag over the area to be used.

1.60 Edit Panel Object Bar Element

Edit Panel Object Bar Element

When editing an object bar element, you must set the Maximum Value. This is the highest value that will be shown. It is used for calculating the percentage value. So for example, if the player starts with 20 energy you would probably want to set this value to 20 unless they are likely to collect energy that will take them above this value.

You must select the frames used for the ON and OFF bar. For example in the case of lives, the ON frame may want to be set to a picture of the players face and the OFF frame to a skull. Another example would be for energy. The ON frame could be a bright bar and the OFF a faded one so that as energy is reduced the bars go faded in turn.

You can position the first and last bars using the buttons. Note that the bars will progress from the first to the last in that direction.

Max Objects is to set the number of objects that will be used in the total bar. e.g. If you have 4 lives this will want to be set to 4. In the case of energy or ammo where a precise value is not required it can be set to whatever value fills the gap best.

1.61 Edit Panel Still Object Element

Edit Panel Still Object Element

When editing a still object element, you must set the Maximum Value. This is the highest value that will be shown. It is used for calculating the percentage value. So for example, if the player starts with 20 energy you would probably want to set this value to 20 unless they are likely to collect energy that will take them above this value.

You must set different frames for different percentage values. In the case of energy, the lower the percentage the more battered the image of the players face. There are other uses too.

Finally you must set the position of the object on the panel.

1.62 Edit Controls

Edit Controls

In a normal game, the player simply uses the joystick left, right, up, down and fire to control the on screen action. For most games, this is fine. However sometimes it is better to configure your controls differently. Amiga CD32 Joypads allow more fire buttons and of course, we have a whole keyboard of buttons available.

An example would be for a simple platform game where all you can do is walk left, right and jump. In this case it might be better to assign Jump to the Fire button rather than the Up button.

You can configure three separate sets of controls named Set A, Set B and Set C. Having three sets allows the player to choose their own controls from a menu screen.

For each set there is a series of possible commands that a key can be assigned to. The commands available are:

For Side Viewed Games

Up: Used for climbing up ladders or shooting above you.
Down: Used for climbing down ladders or shooting below you.
Left: Move left.
Right: Move right.
Shoot: Fire Weapon.
Jump: Makes player jump.
Duck: Makes player duck or crawl.
Press: Makes player activate special event block.
Cycle Gun < : Cycles through available weapons <not yet available>.
Cycle Gun > : Cycles through available weapons in the other direction.
Pause: Pauses the game. Pressing the key again will unpause it.
Quit: Kills the player forcing him to lose a life.

For Above Viewed Games

Up: Move Up.
Down: Move Down.
Left: Move Left.
Right: Move Right.
Shoot: Fire Weapon.
Press: Makes player activate special event block.
Cycle Gun < : Cycles through available weapons <not yet available>.
Cycle Gun > : Cycles through available weapons in the other direction.
Pause: Pauses the game. Pressing the key again will unpause it.
Quit: Kills the player forcing him to lose a life.

Clicking on the PICK button will bring up a further menu to allow you to change the control button to anything you want. All CD32 Joypad buttons are

included too. Note that the Red Button acts as a normal fire button and the second fire button on some joysticks acts as the Blue Button.

You can load and save sets from the pull down menu.

1.63 Edit Menus

Edit Menus

Backbone allows menus to be created within your games. These can have a variety of uses. The most common use is as a 'Main Menu Screen'. Most games show some sort of menu at the start of their game. This usually has several options, Start Game, Quit Game, Select Controls and View Instructions are all common ones.

Other uses for menus are in games that take different routes depending on the users choice. For example, the player may come to the end of a level be told that they have two choices of where to go next, say the forest or the dungeon. Using a menu they can then decide which direction to take and Backbone will send them to the appropriate level.

To use a menu, they must be called as a scene using the
Scene Editor

Clicking on edit menu will bring up a window allowing you to choose which menu to edit. They each have a number ranging from 0 to 20 and a name which you can enter yourself.

Once the name and number have been selected, clicking on Edit Menu will allow you to decide exactly how the menu will work.

First you need to understand how a menu works.
It is a good idea to examine the example menus included with the example projects to get an idea of how they work.

A menu must always have a picture which is used as the general background of the menu. This picture should contain anything that remains constant in the menu. i.e. Controls and menu pointers change but background pictures and words remain constant.

For all the changing elements in the menu, an object file must be used with each of these elements cut out as a frame.

Each menu can also have a number of 'menu objects' to it. A menu object is something that is simply used to give some information to the user. At the moment, the only object type allowed is Control Type. This will display a different frames from your object file for each controls that are picked. For example you could set Controls Set A to be Joystick, Set B to be CD32 Joypad and Set C to be Keyboard. The user could then use the menu to cycle through which control set they would like to use. The menu object would show a different picture for each set. i.e. A picture of a joystick, joypad or keyboard.

Most importantly, each menu has a number of 'options' of which there must be at least one. These options represent the choices that the player can make. e.g. If the only choices were to be Start Game or Quit Game then there would only be two options.

Each option can have a specified action attached to it. For example, Start Game would call the 'Continue' option to carry on the scenes whereas Quit Game would call the 'Quit' option. All the options are listed in more detail below.

For each of these options, a frame must be displayed from the object file. This is usually just an arrow pointing to the option word but other uses are easy to work too. You could make it so each option is highlighted as they are selected for example.

Now let's look at how to edit the menu:

First you pick an IFF Picture to use as the background and also an object file. You must also set the total number of options available which must be at least one using the slider bar.

There are two options of how the menu options can be cycled through by the player. Up/Down or Left/Right. This will depend on how your menu is laid out. If, for example, you had each menu option on top of each other, you would select Up/Down and the player would have to move the joystick up and down (or use the up and down arrow keys) to select which option they wanted.

Next we must set any Menu Objects that are to be used. If your menu has the option of changing the controls, you will want to select one of the objects to have type 'Controls'.

Then clicking on 'Edit Object' will allow you to select which frame to be used for each control set. It also allows you to position the object on the menu screen.

Now all options must be defined. Go through each option at a time (using the slider bar) and select its type. For example, if your menu looked like this:

```
Start Game
View Instructions
Quit Game
```

then you would set option 1 to type 'Continue', option 2 to type 'Goto Scene' (the instructions would consist of another set of scenes which would then return to the main menu) and option 3 to 'Quit Game'.

Clicking on Edit Option allows you to set the details of each of these options. For each one you will have to select the frame to use and where to display it. If you were using an arrow, it should be displayed next to the appropriate option. For example, in the above menu, for option 2 an arrow would be displayed like this:

```
Start Game
> View Instructions
Quit Game
```

Here is a list of all available options, including what they do and the details needed:

Continue: Simply continues on from the scene that displayed the menu. If that was the last scene then the game would start.

Quit Game: Will completely quit out of the game and return to the Workbench. This is also what happens when you press Ctrl+C, however that option can be turned off in the Screen Setup menu.

Pick Controls: This will allow the user to decide which control set to use. If the user presses this option, one of several things will happen depending on how the cycle bar is set.
Use Control Set A : Will change the controls to set A. Similarly for Sets B and C.
Cycle A,B: Will change between sets A and B.
Cycle A,B,C: Will cycle between all the different sets. Everytime the controls are changed the appropriate object images will also be updated.

Goto Scene: This will play the scene as set in the details. If you wish the scene to return to the menu when complete, you must set the last scene to go to the menu display scene.

Change Level: This option will change the current level and then continue, quitting the menu. The player will then start on the level designated in the details.

1.64 Create

Create

When you wish to play your game, whether as the final product or just as a test you should click on the Create button.

Note that this is where the optimisation comes in. The more optimisation, the faster the game will run but the longer the creation process will be.

You will be asked where you wish the game to be created. You should select a path and then type in the name of a directory to use where all the game files will be placed. You should have plenty of room wherever it is. If you have plenty of RAM then the RAM disk is a possibility.

You will also get the option to automatically run the game once it has been created.

NOTE: In this early version of Backbone, the created games cannot span over several disk so either they must be played on hard disk or they should be small enough to fit on one floppy disk.

A window will be displayed and you will be kept informed of what Backbone is doing. When finished you should receive a message informing you that the project creation was a success.

If you get an error message, some common reasons for errors may be:
 Lack of memory or hard disk space.
 Not filling in all the details required (e.g. entering filenames for the
 players object file etc.)
 A bug(!) in backbone. If you believe this is the case, send all the
 project files to
 me
 and I will sort it out.

The next step is to quit out of backbone and run the game by clicking
 on its icon in the drawer you created.
 If the game does not run then you should go through the steps as above to
 find the problem.

To quit out of the game when running press Control and C.

1.65 Image Editor

Image Editor

NOTE: Creating Images in the Image Editor does not automatically mean
 they will put into the game. This is just a general tool that can be
 used for any project. To use the files you must select it at the appropriate
 point in your project.

ALSO: The Number of Colours displayed depends on the colours gadget
 setting. If your images have the wrong colours then adjust this.

All of the foreground objects in Backbone use Object files.
 Object files contain a series of animation frames used to animate these
 objects.
 For example, say for a Mario type character, its object file would look
 like this.

Frame	Description
-----	-----
1	Mario Standing Still Facing Right
2	Mario Starting to Walk Right
3]
4]----- Mario Continuing Walking Right
5]
6	Mario Jumping
7	Mario Falling
8	Mario First Frame of Dieing
9]
10]----- Mario Continueing Dieing
11]

This is a simple example but shows how the frames must be in a consecutive
 order when they need to be animated.

Hotspots:

Each image must have a hotspot, an individual pixel on the frame that is used to measure its location from.

For example, if the mario character had its hotspot set at its feet, then when it was displayed on the screen at coordinates 100,100 it would be its feet that was displayed there and the rest of the image would be displayed above it.

Also, the hotspot decides what objects should be displayed in front of other objects.

Here is a guide as to where hotspots should be positioned on frames.

People/Monsters: Positioned at the base of the feet in the horizontal center.

Explosions/Bullets: Positioned in the center of the image.

Spaceships: Positioned in the horizontal center and just below the vertical center.

Image files can be loaded or saved from the buttons Open Objects and Save Objects. An object file can be added to the end of the objects you are currently editing by clicking on Add Objects. Note that the palettes should be the same for both banks.

You can cycle through the various images and view them on the screen below. Pressing the RIGHT mouse button on this screen will move the position of the image so you can fully view it. Pressing the LEFT mouse button will set the hotspot of the image, shown as a crosshair over the image.

To create frames you must make an IFF picture as used by most art packages and grab the individual frames off it.

To do this click on the "Pick Image" button, if not already loaded, you will have to load an IFF picture. Clicking on the "New Picture" button allows you to change this picture at any time.

When clicking on the "Pick Image" button will bring you to the IFF picture you selected.

Then you can use the left mouse button to make a box around the frame you wish to cut around, the image will automatically be clipped which may take a short amount of time.

The right mouse button will reset the image.

The "Auto Mode On" setting means that when you cut out an image, the next image is moved onto. This is so you can cut out a whole series of images in a row.

The "Delete All" button will erase all images.

1.66 Sound Editor

Sound Editor

NOTE: Creating Sounds in the Sound Editor does not automatically mean they will put into the game. This is just a general tool that can be used for any project. To use the files you must select it at the appropriate point in your project.

From here you can edit the sounds that your game will use.

These will range from gunshots to victory shouts and can be played at specific points in the game.

From the pulldown menu you can load and save sound banks. You can cycle through all the sounds and adjust their given names for reference (8 characters only) You can also adjust the frequency that the sounds will be played at. Clicking on the play button will allow you to hear the sample. Both IFF and RAW sounds can be loaded by clicking on the Load Sample button.

1.67 Block Maker

Block Maker

NOTE: Creating Blocks in the Block Maker does not automatically mean they will put into the game. This is just a general tool that can be used for any project. To use the files you must select it at the appropriate point in your project.

Blocks are what are used to display the background to your levels. They are created from a IFF picture as created by most art packages. The blocks must be positioned in order and together in the picture. Examine the included example files to understand more fully. The picture file can either have borders around the blocks or not. The blocks go in a consecutive order. So if you were using 16x16 blocks then:

Block 1	would have its top left corner at coordinates	0,0
2		16,0
3		32,0

and so on. When you reach the end of a horizontal line, move down to the next one. Screens can be of any height but must be either 320 (lowres) or 640 (hires) pixels wide.

To pick the picture from which the blocks are to be grabbed, click on the New Picture button

The cycle button allows you to manually tell the computer whether the file has borders around it or not. If you do not understand this then simply use the Auto Decide option.

You must also tell the computer what size the blocks are, either 16x16 pixels or 32x32 pixels.

Finally click on the Grab Blocks button and you will be asked for a filename to save the blocks as. Now you must tell the computer how many blocks there are to extract. Do this by clicking on the last block when the picture is displayed.

Your blocks will now be extracted and saved.

IMPORTANT NOTES:

When creating your block picture file you should be aware of the methods used by some of the inbuilt routines to take full advantage of them.

1. Animated Blocks. Background blocks will usually just be displayed as normal but if you wish they can be animated using 2,3 or 4 frames. These frames must appear as consecutive blocks
2. Foreground Blocks. Most blocks will be put into the background but some blocks can be overlaid on top of objects. These blocks must have an extra block with a background colour of zero to be overlaid. This allows for blocks that are partly see through, partly solid.
3. Destroyable blocks. Most blocks will stay forever but some can be set to be blown up or destroyed in some manor. When this happens the block will change to the next consecutive block. So if you want block 12 to be destroyed and set to a blank block then set the block 12 to be destroyable and make block 13 a blank block.
You could make blocks 12,13,14 as blocks with more and more destruction apparent in them and then block 15 blank. Setting 12,13,14 to be destroyable would mean the player must blow up the block 3 times to completely destroy it. Each time it would look more damaged.

1.68 Floppy Disk Tool

Floppy Disk Tool

When you create your final project, you will usually do so onto your hard drive. However, if you wish to distribute your game you will likely wish to do so on floppy disk. This tool will do all this for you, formatting the disks, copying the files etc.
It will also allow your project to span across several disks.

NOTE: Not implemented yet! Register and then pester me to get it done!

1.69 Pick Sound Effect

Pick Sound Effect

Select the sound effect, created previously using the
Sound Editor
and put into your project in the
Game Details
menu.

From here you can pick the exact effect to player to be played at the particular point in the game, depending on how you reached this menu.

e.g. if you clicked on the SFX button when selecting the dieing frames for your player then this sound effect will played when your player dies.

1.70 Pick Scene

Pick Scene

Select the scene, created previously using the Scene Editor to be played at the particular point in time depending on how you reached this menu. ←

e.g. if you clicked on the Intro button in the Game Details section then the scene you select will be played for the games introduction sequence.

1.71 Index

A

Animations

Arexx Support

Arexx

B

Baddies Weapon

Block Editor

Block Maker Commands

Block Maker

Blocks Commands

C

Chasing Enemies

Collectable

Common Problems

Comparisons

contact me

Contacting Me

Controls Commands

Controls

Create Commands

Create

Credits

D

Danger Level

Destroyable

E

Edit Blocks

Edit Controls

Edit Details

Edit Levels

Edit Menus

Edit Objects

Edit Panel

Edit Player

Edit Scenes

Editing the Player

Example Projects

F

F.A.Q.

Floppy Disk Tool

For Above Viewed Games

For Side Viewed Games

Foreground

G

Game Details Commands

Game Details

Graphics Wanted!

I

Image Editor Commands

Image Editor Tool

Image Editor

Installation

Introduction

L

Level Editor

Levels Commands

Limitations

M

Mailing List

me

Menu Editor

Menus

O

Object Bar

object editor

Object Editor

P

Panel Commands

Panel Editor

panel

Patrolling Enemies

Picture Based

Player Commands

Preferences

Project File Commands

R

Recent Additions

Register

Registering

Requirements

S

Scene Editor

Scenes Commands

Scenes Editor

Screen Setup Commands

Screen Setup

Simple Bar

Simple Text

Solidity

Sound Editor Commands

Sound Editor

Step By Step Guide

Still Object

T

Text

The Future

The Main Menu

tutorial

V

Various Useful Commands

W

Weapon Shot
