

Backbone-Manual

COLLABORATORS

	<i>TITLE :</i> Backbone-Manual	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		October 30, 2022
		<i>SIGNATURE</i>

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Backbone-Manual	1
1.1	Backbone Manual	1
1.2	Additions Since V1.00	3
1.3	Introduction	5
1.4	Example Projects	6
1.5	Requirements	6
1.6	Tutorial	7
1.7	Limitations	14
1.8	Tool Types	15
1.9	Common Problems	16
1.10	How to Register	17
1.11	How Do I...?	19
1.12	Step By Step	19
1.13	Main Menu	21
1.14	Game Details	23
1.15	Screen Setup	23
1.16	Edit Player	24
1.17	Players Frames Settings - Side View	27
1.18	Players Frames Settings - Above View	29
1.19	Edit Objects	29
1.20	Editing a Patrolling Baddie	30
1.21	Editing a Collectable Item	31
1.22	Editing a Weapon Shot	32
1.23	Editing a Chasing Baddie	33
1.24	Edit Blocks	35
1.25	Blocks Solidity	35
1.26	Blocks Animations	36
1.27	Blocks Foreground	36
1.28	Blocks Destroyability	36
1.29	Blocks Danger Levels	37

1.30 Edit Levels	37
1.31 Edit Scenes	40
1.32 Edit Panel	40
1.33 Create	41
1.34 Image Editor	42
1.35 Sound Editor	43
1.36 Block Maker	43
1.37 Floppy Disk Tool	45
1.38 Pick Sound Effect	45
1.39 Pick Scene	45

Chapter 1

Backbone-Manual

1.1 Backbone Manual

Backbone Manual by Alastair Murray

Pressing HELP at any time during Backbone will bring up the help screen appropriate to the current screen.

NOTE: I'm not too great at writing manuals. If you come across any problems which you don't understand then please contact me and I will fully explain it to you and also modify this document for future releases.

Additions Since V1.00
- Updates and Bug Fixes

Introduction
- General information about Backbone

Requirements
- Specifications for running Backbone

Limitations
- How the full version differs

Common Problems
- Help Guide to Problems

Tool Types
- Various Settings to prevent crashes.

How to Register
- How to get the full version

The Example Projects
- How to use them and what they are.

How Do I....?

- Answers to common complex questions

Step By Step Guide

- Several steps to go through when making a project

Tutorial

- How Example Project 1 was made.

The Main Menu

- Commands Available from the Main Menu

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

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
-

1.2 Additions Since V1.00

Additions Since V1.00

V1.20

Fixed stupid file selector bug and open project bug that would cause Backbone to lock up.

V1.19

Made it a less CPU intensive, multitasking with AMIGA+A should work better and I'm hoping (again) that this will solve some of the crashes that some people seem to get.

V1.18

Reverted back to the old way of making level maps which means there's less flexibility in the map sizes but I hope that some peoples crashes will be solved (although I was completely unable to find any reason for a crash - it worked fine on mine and many others!)

V1.17

Fixed a problem to do with changing the number of blocks on a level. Got rid of a stupid error message.

V1.15 & V1.16

Fixed a very embarrassing problem that Backbone actually randomly wrote over memory in the level editor which caused crashes on peoples machines.

V1.14

Fixed a couple of stupid bugs. Particularly to do with painting level blocks.... whoops!

V1.13

Added tutorial for how to create example 1.
Fixed Auto Grab bug for getting the size of the player and objects.
Changed the way variables works so there should be less crashes and you should be able to have larger map sizes with no problems.
File Requestors remember where they were last at.
Fiddles with the level editor a bit to make it less sensitive.

V1.12 and V1.11 and V1.10

Fixed another Image Ed bug. Fixed a few File Req things. Fixed a NEW Project bug. Fixed an Edit Level bug.
Added AUTO option on the player's width and height bit. This will look at your still frame defined for the player and work out a suitable width and height for it. Also added this option for the baddie's heights and widths.
Added the option for different music for each level. In the level editor you will have the option of selecting music. The general "Game Music" in

the details menu has now gone.
Updated the example files because they were earlier versions and were causing all sorts of problems.
Fixed another stupid File Req bug.
Added Danger Level on blocks. Blocks can either 1.Hurt You on Contact 2.Kill You on Contact 3.Hurt You when Falling on them 4=Kill you when Falling on them.
Added No. of Still frames option so that the player can constantly go through an animation when not moving.
Added test for write protected Temp directorys so that CD-ROM version go OK.
Adjusted Example 3 to use the steam as a "danger block"

V1.09

Fixed stupid Image Editor bug.
Added a check to ensure you don't lose any blocks when saving a project.

V1.08

Fixed problem with Map Sizes being incorrect in a game.
Added loads of Tool Types: See
 Tool Types
 for details.
Reducing these values will help stop crashes.

V1.07

IRC channel. Go to #backbone on irc.demon.co.uk OR irc.bt.net for help on Backbone. Obviously there's not always someone there but keep trying!
Fixed Image Editor bug. Fixed Save Game bug. Reorganised the game code to help with incompatibility problems.
Added a new tooltype. EVENTS sets the maximum number of events allowed in a level. 30 is the default. Be warned that if you change this value, games may run funny if you don't keep it constant each time you edit it!

V1.06

Most crashing bugs appear to be gone! If people still have trouble with the iconify option (I think it's to do with PAL/NTSC) then use the AMIGA+A option for now. Its far from perfect because the OS will go very slow using this and you won't be able to run the games at the same time as Backbone.
Baddies can now jump! Added jumping/falling images for the baddies too.
Added Chasing Baddies of all kinds, jumping, flying etc.
Fixed a bug to do with dieing baddies staying in mid air and one to do with flying baddies getting stuck in blocks.
Added the SIGHT option into chasing baddies so they can either only see the player when he's in view OR see him anywhere on the map.
Changed the level editor to give seperate menus for Objects and Events.
Added Events, walking over specific blocks will activate the event which can be: 1.Teleporting the player, 2.Creating a new object, 3=Removing an object, 4=Playing a scene, 5=Changing a block.
Events can be either activated when the player presses SPACE or automatically.
Fixed bugs that prevented some games from working with Full Optimisation.
Added Score to Panel.
Changed example2 to include some of the new features.

Added a new tooltip. Lockout=No means that you CAN use AMIGA+A to flick between Backbone and the Workbench.

V1.05

Fiddles with various things to improve compatability.
Removed bug to do with bouncing off baddies into a solid wall (I think)
Made an option for "Ghost" Baddies where they can walk through walls.
Changed the Iconify option back to how it was before V1.01 as I just found out it doesn't let you run you programs at the same time as Backbone <sigh>
Renamed Backbone.Library To Bone.Library (for important reasons I assure you), removed AMOS.Library as you don't need it any more.
Also removed the GUI drawer as it is not needed either.

V1.04

Removed bug to do with map sizes. Added the ability to flick between workbench and backbone using AMIGA+A to help the people who had problems with the iconify option. Fiddled with one or two other things.

V1.03

Reduced memory consumption by a further 150k. Fiddled with a few things in the hope of making it more compatable. Added two new tooltips, X=200 and Y=200. These set the maximum sizes of the levels in blocks.
If you are having troubles with crashes then try reducing these values. It is not recommended that you do not exceed 200. Values of 80-120 are usually enough.

V1.02

Fixed a bug to do with shooting enemies. Fixed a bug to do with loading projects. Added a NTSC/PAL mode in the tooltips of both games and Backbone.

V1.01

A couple of things fixed. Now uses 300k less memory which should help stop a few crashes. Also changed the iconify option as some people had problems with it and I wasn't happy either.

1.3 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.

1.4 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 example 1 where some of the graphics were contributed to from my friend Manuel Moreno)

Example1:

This project shows the basics of a platform game using the Powerhead option (as seen in games such as Mario Bros and Giana Sisters), it also shows how background blocks can be animated and be destructable.

You can see how this project was made by following the step by step tutorial

Example2:

This project is mostly to show you how the scenes work, it has an introduction scene comprising of an animation and picture. There are also scenes before and after each level including music and pictures. As for the game, it will show how to make a basic platformer. It includes normal and flying baddies and also a weapon. One of the baddies (the Tank) is a jumping baddie and one of the flying baddies (the Fish) is a chasing baddie. The others are simply normal patrolling baddies.

Example3:

This project shows how different elements of platform games can work in Backbone. In this one your character can crawl and climb ladders as well as having a weapon to start off with and being able to fire it in any direction.

Example4:

A simple project showing how an overhead viewed game works.

1.5 Requirements

Requirements for Backbone

You will need a Hard Disk. 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.

NOTE: If the program unexpectedly quits, it may be due to lack of memory or lack of temporary disk space.

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
More
Delete

If you have any troubles with missing files then
contact me

.

1.6 Tutorial

Tutorial

This tutorial will take you through all the steps to create the first example project. It will use the graphics and sounds as provided in the archive (not in the minimum download version though)

It is advised that you first play the example 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 load up Backbone. You can flick between this guide and Backbone by either iconifying Backbone, Using the flick option (see
tool types
)

or by opening the help guide from within 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 "SmallMan.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 6 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
- 7-12. Dieing

Blue Frames

13. Still
- 14-16. Walking
17. Jumping
18. Falling
- 19-24. 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, from the pulldown menu select "Save Objects". Save the file as "TutorialMan"

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 3 frames of it. There are also 8 frames of a small grey baddie (4 moving, 4 dieing) and 5 frames of a coin (4 animated, 1 collected)

Each of these different objects must be held in its own object 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 "TutorialItem"

Baddie as "TutorialBaddie"

Coin as "TutorialCoin"

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 "SmallBlocks3.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 "TutorialBlocks"

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.

The sample names can be changed by clicking on "New Name". Give them all a suitable name before using the pulldown menu to save the sounds as "TutorialSounds"

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. "Tutorial" 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 "TutorialSounds.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".

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 "TutorialMan.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: 8
Jumping Speed: 8
Energy: 1 (it will only take one hit to kill him)
Lives: 2
Climb Speed: 0 (as the player cannot climb)
Max Jump Height: 42 (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 Dying Frame" to 7 and "No. of Dying Frames" to 6.
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.

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 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 "TutorialBaddie.Obj". Click on "NEW NAME" 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: 4

Falling Speed: 10

When Hit Player: Zap Energy

Energy to Zap: 1

Energy: 1

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: 4
Jump Frame: 1
Fall Frame: 5

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 "TutorialItem.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: 1
1st Collected Frame: 1
No. of Collected Frames: 3

Collected SFX: Halleluyah!

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

Click on "New Object" and "Pick File" as "TutorialCoin.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 "Ammount" to 1.

Set the following variables too:

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

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 "TutorialBlocks.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 12-21 and 24-25 and 36 and 41, the last block. 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. 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.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                               X
X      X                       X   X
X     XXXXXoOOoXXXXX         X
X                               X
X                               X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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     XXXXXoOOoXXXXX         X
X                               X
```

```
X P      c c c      I X
XXXXXXXXXXXXXXXXXXXX
```

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 baddie), 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 Ammount 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.

1.7 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:

This is the first release of Backbone and is severely underdeveloped. At the moment there is little options in using the Overhead Viewed game. It has been mainly developed for the side view games. One of the main updates will be a move to a full Workbench GUI application.

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.

1.8 Tool Types

Tool Types

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

To change the tool types, in workbench, select the Backbone icon and from the pulldown menus select Icons and then Information. Alternatively just press AMIGA+I. You can now adjust any of the settings. You must run Backbone again to make use of the new settings.

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.

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.

Your options are:

MODE= This can be set to PAL or NTSC with obvious consequences.

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.

FLICK= This can be set to YES or NO. It is used for people who have
troubles using the iconify option in Backbone. Setting it to YES

means that you can use AMIGA+A to flick between WB and Backbone.
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.

1.9 Common Problems

Common Problems

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.

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: Backbone crashes at random times, often when Creating the game.

Solution: 1. You may be lacking memory, close other programs you have running.
2. You may be running a program in the background that interferes with the running of Backbone. Try loading Workbench without any of the extra programs you may have.
3. Try editing the

Tool Types
and reducing some of the values.

Problem: I can't run Backbone with a Graphics Card. The display is corrupted.

Solution: This is a problem. I know there are some software fixes available that will solve the problem for certain types of graphics cards but other than that, I hope to solve it myself sometime.

Problem: When I go into a certain menu, Backbone suddenly quits with an error message.

Solution: There may be a file missing or corrupt. For example, if you try to edit the players frames when the file you have selected does not exist, then problems will occur.

Problem: I want to run the program and games in NTSC mode.

Solution: Edit the
Tool Types
of the icon to read MODE=NTSC instead of
MODE=PAL.

Problem: I want bigger levels.

Solution: Edit the
Tool Types
of the Backbone icon and increase the values
of X=*** and Y=*** where *** are numbers. 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: The iconify option crashes my computer.

Solution: Try using AMIGA+A instead. You must change the
Tool Types
of the
Backbone icon to FLICK=YES first though.

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

Solution: Adjust the
Tool Types
of Backbone to higher
values.

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

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

1.10 How to Register

How to Register

How to Contact Me (Alastair Murray)

1.E-Mail:

I have several e-mail addresses any of which may be available or unavailable at various times throughout the year.

Try these addresses in this order (or all at once)

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

2.Web Page:

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

3.My Address:

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

As Backbone is currently at an underdeveloped stage and at its first release, this version will be much cheaper than later versions. It will eventually move onto a full Workbench GUI and therefore the price will increase.

Registering will give you complete updates to later versions of Backbone and the possibility of requested personal specific elements to projects to allow your projects to exactly as envisaged as well as removing all limitations from it.

You can receive the registered version by e-mail or by post. If you wish me to post it then either include a disk in a jiffy bag with a stamp or simply add a bit to the cheque to cover the costs of a disk and posting.

Register now by sending to the above address a cheque for:

8 UK pounds

Please don't send foreign cheques or foreign cash, you can get your money changed at your bank. If you must send a foreign cheque then please DOUBLE the amount because I have to pay 6 pounds for the exchange!

This would make it:

40 German DM

30 US \$

40 NZ \$

or equivalent in another currency.

Please make it payable to Alastair Murray.

Include also your name, address (if needed) and e-mail address (if needed).

If you can e-mail me in advance then I can prepare the key file in advance and send it quicker. Tell me your name and e-mail address.

I will e-mail or send by post a small "key file" that can be placed in the Backbone directory or in the S directory and will automatically enable all available options within Backbone and disable the demo messages that appear both in Backbone and in the final product.

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 keyfiles.

For anyone I find doing this, I will add a bit to the Backbone for its next release that prevents them from running it.

1.11 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.

...test the game while still running Backbone?

If you have enough memory you can iconify Backbone, run your created game and then quit it with Control+C and return to Backbone for modifications.

...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.

1.12 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
Scenes Editor
and create any scenes needed for the introduction, game complete, game over and any in-between levels scenes required.
 13. Go into the
Sound Editor
and load any sound effects needed for your game. Save them before exiting.
 14. 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.
 15. Go into the
Panel Editor
and create the information bar.
 16. 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)
 17. Create the background of the level using the blocks.
 18. Set the players starting point and add any baddies/items etc.
-

19. Set the finishing Conditions for the level.
20. Set any scenes needed before or after the level and ensure that the "Next Level" option is correctly set.
21. Finish editing the level and create any more as needed.
22. Create you game with no optimisation and test it.
23. Adjust as necessary and then repeat step 22 until happy.
24. Finally create the game will full optimisation.

1.13 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.

Optimisation 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. 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.

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

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
-

1.14 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.

1.15 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.

NOTE for *AGA* machine owners: In this early version of the game, AGA is
not supported. I hope that, as long as sufficient demand is made, AGA
support will implemented in the near future.

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.

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 soemtimes 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. All directions. The player and baddies can fire bullets in any of the eight available directions. This allows for shooting enemys above or below you. However you will have to draw more animation frames and the program and control method will be more complex.
3. Left and Right Only. Things can only fire in the direction they are facing. Most commonly used for simple platform games. This option is only available for side viewed games.

1.16 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 spacehips 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 ammount 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

1.17 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. If your gun types (as set in the screen setup

section) are set to be able to fire in any direction ↔ then five

direction buttons will appear 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.

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.

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.

1.18 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.

1.19 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

1.20 Editing a Patrolling Baddie

Editing a Patrolling Baddie

NOTE: In this early stage of Backbones creation, Patrolling Baddies will only function on Side Viewed Games.

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.

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 three options of how the baddie can be killed (more coming in later versions).

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.

Set Frames:

This works as in the
Edit Player
section.

1.21 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.)

1.22 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.

Set Frames:

This works as in the

Edit Player

section.

1.23 Editing a Chasing Baddie

Editing a Chasing Baddie

NOTE: In this early stage of Backbones creation, Chasing Baddies will only function on Side Viewed Games.

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 three options of how the baddie can be killed (more coming in later versions).

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 baddie's 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 baddie's 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.

1.24 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.

1.25 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.

1.26 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.

Clicking with the right mouse button will reset it.

1.27 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.

Clicking with the right mouse button will reset it.

1.28 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.

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.

1.29 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.

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.

1.30 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.

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 Tool Type of Backbone. (Select it's icon in WB and press Amiga+I). Edit the EVENTS=30 to a larger value.

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.

When your level is complete, click on DONE.

1.31 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.

To make an individual scene:

Select the scene number.

Give the scene a name by clicking on the empty large box.

Use the cycle gadget to choose what you wish this scene to do.

Now select the file concerned (picture, animation, music or whatever)

Then click on edit details to set the exact details of the scene depending on its type.

Finally set the "Next Scene" field to point to the number of the scene to play next. e.g. for the previous example scene 1 would point to scene 2 and scene 2 to 3 etc. Scene 4 would point to 0 as it is the last one.

Scene Types:

1. Show IFF Picture. Select an ILBM 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.

1.32 Edit Panel

Edit Panel

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

At this early stage of Backbone's development the panel option is very limited. There are only currently two options.

1. None - No panel at all is used.
2. Simple - The panel contains simple text information telling you the amount of energy and lives left.

1.33 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.

NOTE: In this early version of Backbone, the created games cannot span over several disks 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:

In this early version of Backbone, full error checking is not implemented.

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 player's 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.

NTSC/PAL

You can change what mode Backbone runs in and the games created run in by adjusting the tooltype of the icons. If you wish to revert to default then miss out the tooltype altogether.

It should read: MODE=PAL or MODE=NTSC

1.34 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.

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 pulldown menus. 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.

1.35 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.

1.36 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.

1.37 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!

1.38 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.

1.39 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.
