

# **WolfClone**

Alastair M. Robinson

|                      |
|----------------------|
| <b>COLLABORATORS</b> |
|----------------------|

|               |                             |                 |                  |
|---------------|-----------------------------|-----------------|------------------|
|               | <i>TITLE :</i><br>WolfClone |                 |                  |
| <i>ACTION</i> | <i>NAME</i>                 | <i>DATE</i>     | <i>SIGNATURE</i> |
| WRITTEN BY    | Alastair M. Robinson        | August 24, 2022 |                  |

|                         |
|-------------------------|
| <b>REVISION HISTORY</b> |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
|        |      |             |      |

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>WolfClone</b>                                       | <b>1</b> |
| 1.1      | Title page . . . . .                                   | 1        |
| 1.2      | What do you need to run the engine. . . . .            | 1        |
| 1.3      | Legal stuff regarding The program. . . . .             | 2        |
| 1.4      | The future... . . . .                                  | 2        |
| 1.5      | Currently implemented features . . . . .               | 2        |
| 1.6      | Controlling the `game' . . . . .                       | 4        |
| 1.7      | About the Author . . . . .                             | 4        |
| 1.8      | Other programs written by the author: . . . . .        | 5        |
| 1.9      | Notes for readers of a technical persuasion... . . . . | 5        |
| 1.10     | How to order the full version of TurnTables. . . . .   | 5        |

---

## Chapter 1

# WolfClone

### 1.1 Title page

A Wolfenstein 3D Clone  
(This title is becoming less and less appropriate!)

© copyright 1996 by  
Alastair M. Robinson.  
Written partly in AMOSPro, with the calculations in machine code ↔  
.

There are two versions of the demo, the only difference between them being the floor texture.

Legal stuff...

Requirements

Controlling the 'game'

About the Engine

Future plans

Other programs  
written by the author.

Please read this  
if you know how to program the blitter!

### 1.2 What do you need to run the engine.

This engine requires only the AGA chipset, and an '020 or higher. It has yet to be tested on anything faster than a 25MHz '030, so if you have an '040 or '060

---

please let me know how it runs.

FastRAM is not required but is highly recommended, since it makes the game much faster, and more playable. On a standard A1200 without fastram, the frame rate is acceptable if you turn the floors and depth-cueing off.

The Icons I have provided may look more than a little strange under a standard (Magic)WB palette. For this reason, I have provided my own personal favourite palette file, just so you can see what the icons are \*supposed\* to look like. Just use one of ShoveColors type utilities with the palette file.

### 1.3 Legal stuff regarding The program.

This demo may be freely distributed, provided no files are modified or removed, and no more than a nominal copy fee is charged for it. You may not add files other than BBS tags to the LhA archive.

You are allowed to decompress the files onto a self-booting disk, and to distribute that, provided no files are modified or removed, and no more than a nominal copying fee on top of media cost is charged. No files other than those needed for booting may be added. This last restriction obviously does not apply to CD distribution!

None of the graphics may be distributed separately from this program, or used in others. The sounds are not my work, and so are not subject to this rule.

THIS PROGRAM IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND. ALL USE IS AT YOUR OWN RISK, AND THE AUTHOR CANNOT BE HELD RESPONSIBLE FOR DAMAGES OF ANY KIND, ARISING FROM THE USE OR MISUSE OF THE PROGRAM OR ANY OF ITS ACCOMPANYING FILES.

Sorry about all that.

### 1.4 The future...

Future plans for this engine:

This engine will almost certainly become a game, hopefully reaching completion by the end of the summer, and will probably be shareware. I might also release a level editor to go with it.

I'm intending to make the gameplay as varied as possible, which will include the possibility of taking several different routes through each level. A chief criterion will be that the levels must make sense, i.e. everything must exist for a reason.

Since the 'Killer Stacy' is so enthralled with the game in its current form there will certainly be a version of the finished game which is suitable for kids to play. None of the existing texture mapped games are, in some cases more because of their sounds than their graphics!

Any comments or suggestions would be most welcome.

### 1.5 Currently implemented features

---

My 3D engine uses the ray-casting system, as described by John Corigliano in Steffen P. Haeuser's TMapFAQ, somewhere in the docs/misc directory of Aminet.

Until I found this document on CU-Amiga's cover CD, I wouldn't have known how even to begin programming a game of this type.

I have also used the copper-chunky screen described in the same document. I still think this looks better than blitter screens. Besides, it's a real treat to be able to use 12 bit High-Colour, even if the resolution is low.

Basic features and details of this engine:

**Textures:** The current design permits 32 texture blocks (which are 64 pixels square.) Of these, 16 are static textures, and the other 16 form two animated textures of 8 frames each (the first animation slot is currently used for the fans.)

**Objects:** The rendering of objects in this engine is a huge cheat; an object is drawn as though it is always facing the player. (This method is common to almost all games of this type.) This was actually the most difficult part of the entire engine to code in an efficient way. My solution to the problem is far from perfect, since the objects look like they're curved towards the player. I think that a better solution would require that the objects are taken separately from the map, but I'm not sure about this. The current method is certainly acceptable.

**Z-Buffer:** Black areas of textures are treated as transparent, so blocks with windows in them are possible. Normally a ray for each column on screen is cast into the world until it hits a block; if the block has transparency, the texture is stored in the z-buffer, and the ray cast further. This was necessary to allow objects to be rendered in front of walls, so it might as well be applicable to walls as well as objects. It is interesting that the only blocks you can 'see round' in Wolfenstein are pillars, which would have to be rendered as objects (i.e. always facing the player) anyway.

**Doors:** The engine allows 32 stages of opening to be rendered. Any texture may be used as a door, so secret doors are no problem. The door attribute is not mutually exclusive with any other, so it would be perfectly possible to have an animated door with a window in it! Doors can only slide sideways into an adjacent block; blocks which slide away from the player (à la Wolfenstein's secret doors) are not supported.

**Realtime movement:** All movements and animations in the demo are independent of processor speed. The fans may look a little jerky from a distance (especially without fast ram) but I figured this was preferable to them doubling in speed as you approach them!

**Floor Mapper:** This is really a totally separate routine, not handled by the ray caster - and it shows; the floor doesn't quite match up with the walls, but I think the overall effect is pretty good.

**Explosions:** When you shoot a barrel, it will explode and leave behind a charred shell. The explosion graphics were generated in a program of my own, which I'll be releasing onto Aminet shortly. I wrote it because I couldn't find an explosion generator on Aminet Set 2 which gave convincing results. I'll probably make a GUI for the explosion generator before I release it.

**Switches:** The switch handling code is now pretty well complete apart

---

from sound handling.

Enemies: The enemy handling code is about half done at the moment; currently supported are enemies which move towards the player, and static enemies (used for the explosions, since they need to be in front of the object which is exploding.) The 'enemy' included in the demo is just a digitised still. When I started writing the enemy handling code, I needed an image to try it out. After a tedious session of background removing, the 'Killer Stacy' was born!

Incidentally, Stacy (who's seven) is absolutely enthralled with this engine, and has given me more useful feedback than anyone else so far. I do have her permission (and that of her parents) to place this on the 'net, by the way!

## 1.6 Controlling the 'game'

Controlling the 'game'

Use the cursor keys or move the mouse horizontally to rotate left and right.

The cursor up key can be used to move forward.

The cursor down key will move you backwards.

The left mouse button can be used to walk in either direction; while holding down the button, move the mouse sharply towards you (about half an inch) to set the direction to backwards. The direction will be reset to forwards if you move the mouse sharply away from you, or if you let go of the button.

Use the right Amiga and Alt keys to sidestep left and right respectively.

Holding the shift key will increase the speed at which you walk.

Doors can be opened and switches operated with the space bar, or middle mouse button (if applicable).

Pressing the right mouse button or left alt key will play a machine gun sound effect, and decrease the ammo counter.

F1-3 sets the width of the screen, and F4-6 sets the Depth of field to match screen widths 1-3. Stand in front of a fan and play with these to see why I included them.

F9 turns on the Depth Cueing effect, and F10 turns it off (for a slight but noticable speed increase!)

F7 turns on the floor, and F8 turns it off.

You can't have a floor without depth cueing, and you can't turn off depth cueing without turning off the floor as well.

The Help key turns on the heads-up map, and the Del key removes it.

The '+' and '-' keys (keyboard or numeric keypad) will allow you to zoom in and out.

Finally, the Escape key is used to quit.

## 1.7 About the Author

Alastair M. Robinson

The Old Chapel  
Syderstone

---

Norfolk  
PE31 8SD  
United Kingdom  
  
(01485) 578 740

## 1.8 Other programs written by the author:

Lookout for these programs, coming soon to an Aminet site near you. ←

In game/misc, an educational game designed to help children learn their multiplication tables. The game is called TurnTables, should work on all Amigas (tested on A500, A600, A1200, A4000/030), and the archive should be called TurnTabs.lha or something similar (I'm not uploading these things myself...)

The version on the net is a demo version, but you can order the full version directly from me (for £5) if you wish.

In gfx/misc, a simple (but more effective than anything else I've seen on Aminet set 2) Explosion generator, also written in AMOS Professional. I needed a convincing explosion effect for my WolfEngine, so I came up with this. I'll probably put some work into a user interface for it before release though, so don't hold your breath.

## 1.9 Notes for readers of a technical persuasion...

To anyone who knows how to program the Blitter:

If you were to build a chunky buffer transposed in memory, so that consecutive bytes describe a COLUMN rather than a row, then used the blitter's line pattern feature to rotate it 90\textdegree{} into a new buffer, would the result not be 8 ← interleaved bitplanes? I don't know the blitter well enough to try this out myself, but I have some dim distant recollection of an example in the Hardware Reference Manual which rotated a bitmap by 90\textdegree{}. If it could be made to work, it could ← provide real asynchronous hardware based chunky to planar!

## 1.10 How to order the full version of TurnTables.

If you want to receive the full version of TurnTables, on an autobooting disk with a printed instruction leaflet, send £5 sterling, (or U.S.\$10 overseas) to:

Alastair Robinson  
The Old Chapel  
Syderstone  
Norfolk  
PE31 8SD



## ENGLAND

I can accept the following methods of payment:

- »» £5 sterling, in cash. (This is the preferred method.)
- »» A cheque for £5, drawn on a UK bank.
- »» A £5 postal order, purchased in the UK.
- »» U.S. \$10 dollars, in cash.
- »» An International money order.

Unfortunately, I cannot accept foreign cheques drawn on non-UK banks, or foreign postal orders. I can't accept Eurocheques either.

Please make sure that when sending cash, it is heavily disguised in the envelope. Wrapping the cash in a sheet of coloured paper should do the trick.

---