

# **CBlocksWin Version 1.0 User Guide**

This User Guide, and the programs 'CBlocks' and 'CBlocksWin' are Copyright (C) 1997-2005 by Will Stevens (william@stevens93.fsnet.co.uk). Do not distribute the software without this User Guide.

## **Contents**

1. Introduction
2. Menu Commands
3. Input file format

Appendix A. Tile Types

Appendix B. Tile Codes

## **1. Introduction**

CBlocksWin is a windows version of the CBlocks simulation software. CBlocksWin allows you to open, run and save simulations as well as create your own machines interactively. You can navigate around the universe by clicking the mouse at the point in the universe that you want to move to the centre of the active window. You can also zoom in and zoom out.

When you first start CblocksWin, you must resize the toolbar by dragging its bottom edge so that all of the buttons are visible.

## **2. Menu Commands**

File

New

Open...

Prompts for a .grd or .gr2 file.

Close

Save

Saves the state of the universe in the active window as a .gr2 file.

Save as...

Prompts for a filename then saves the state of the universe in the active window as a .gr2 file.

Exit

Edit

## Select

Set the cursor to 'select' mode, in which you can drag across an area of the universe to select it.

## Save Selection...

Saves the current selection as a `.gr2` file.

## Import

Import a `.grd` or `.gr2` file at the selected location. The imported file will be located at the bottom left corner of the selected region and will overflow the selected region if the region is not large enough.

## View

### Zoom in...

Zooms in on the centre of the active window.

### Zoom out...

Zooms out from the centre of the active window.

## Run

### Go

Start running the simulation.

### Stop

Stop running the simulation.

### Step

Run a single simulation iteration step.

## Connect

### Fuse

Fuse tiles together. Click on two neighbouring tiles one after the other and they will become fused.

### Unfuse

UnFuse connected tiles. Click on two fused tiles one after the other and they will no longer be fused.

## 3. Input file format

CBlocks supports two file formats: `.grd` and `.gr2`.

`.grd` files are supported for backward compatibility only and are not described here.

### 3.1 `.gr2` input files

`.gr2` input files use a whitespace-delimited text file format and can easily be created using a text editor or by an automatic script.

`.gr2` input files begin with a pair of positive integers which specify the width and height

of the area described by the rest of the file.

For every tile being read in (including empty ones), the file contains a record of with the following structure:

```
<type> <orientation (0-3)>
```

```
<connect-n> <connect-e> <connect-s> <connect-w>
```

```
<output-n> <output-e> <output-s> <output-w>
```

```
[Pr=<property>]
```

Here, the 'Pr=<property>' part of the record is optional, and only applies to some tiles that have an internal state.

These records are arranged in the file as follows:

```
record for x=0,y=0
```

```
record for x=0,y=1
```

```
record for x=0,y=2
```

```
...
```

```
record for x=1,y=0
```

```
record for x=1,y=1
```

```
record for x=1,y=2
```

```
...
```

```
record for x=width-1 y=height-1
```

## Appendix A. Tile Types

The following tile types are implemented in CBlocksWin Version 1.0:

Wiring and Logic		
Tile	Name	Function
	insulator	none
	wire	$\text{output}[\text{north}] = \text{input}[\text{south}]$
	inverter	$\text{output}[\text{north}] = \text{!input}[\text{south}]$
	set/reset	if input[east], set output[north] and output[south] to 1 if input[west] set output[north] and output[south] to 0
	delta	$\text{output}[\text{north}] = \text{input}[\text{south}]$ $\text{output}[\text{east}] = \text{input}[\text{south}]$ $\text{output}[\text{west}] = \text{input}[\text{south}]$
	and gate	$\text{output}[\text{north}] = \text{min}(\text{input}[\text{east}], \text{input}[\text{west}])$
	or gate	$\text{output}[\text{north}] = \text{max}(\text{input}[\text{east}], \text{input}[\text{west}])$
	nand gate	$\text{output}[\text{north}] = \text{!(input}[\text{east}] \ \&\& \ \text{input}[\text{west}])$
	nor gate	$\text{output}[\text{north}] = \text{!(input}[\text{east}] \    \ \text{input}[\text{west}])$
	equal	$\text{output}[\text{north}] = (\text{input}[\text{east}] == \text{input}[\text{west}])$
	crossover	$\text{output}[\text{north}] = \text{input}[\text{south}]$ $\text{output}[\text{east}] = \text{input}[\text{west}]$
	pulser	if a rising edge is detected on input[south], output[north] = input[south], otherwise output [north] = 0
	store	if output[north] is 0 and input[south] is non-zero then set output[north] to input[south]. if input[east] && input[west] then set output[north] to 0

<b>Wiring and Logic</b>		
	multiplier	$output[north] = input[east] * input[west]$
	adder	$output[north] = input[east] + input[west]$
	negator	$output[north] = -input[south]$
	less-than	if $input[west] < input[east]$ then $output[north] = 1$ otherwise $output[north] = 0$
	greater-than	if $input[west] > input[east]$ then $output[north] = 1$ otherwise $output[north] = 0$
	toggle	if $input[south]$ then invert the state of $output[north]$

<b>Forces</b>		
<b>Tile</b>	<b>Name</b>	<b>Function</b>
	thruster	if ( $input[south]$ ) apply force on self in southward direction
	pusher	if ( $input[south]$ ) then apply northward-directed force on the tile to the north (if any)
	right slider	if ( $input[south]$ ) then apply eastward-directed force on the tile to the north (if any)
	left slider	if ( $input[south]$ ) then apply westward-directed force on the tile to the north (if any)
	rotator	if ( $input[south]$ ) then rotate the tile north clockwise by ninety degrees

Connections		
Tile	Name	Function
	right fuser	if there are tile to the north and north-east, connect them together when input[south] is non-zero
	left fuser	if there are tiles to the north and north-west, connect them together when input[south] is non-zero
	right unfuser	if there are tiles to the north and north-east, disconnect them when input[south] is non-zero
	left unfuser	if there are tiles to the north and north-west, disconnect them when input[south] is non-zero

Others		
Tile	Name	Function
	creator	<p>if the value of input[south] is not zero, and there is no tile north, create a tile north.</p> <p>The type and orientation of the newly created tile are determined by the value of input[south]. See appendix B for details.</p> <p>Type = (input[south]) / 4  Orientation = orientation of creator tile + ((input[south]) % 4) * 90 degrees</p>
	detector	<p>if there is a tile south, output a signal north determined by the tile's type and relative orientation, otherwise output zero.</p> <p>output[south] = type * 4 + orientation / 90 degrees</p>
	direction detector	if there is a tile south, output[north] = relative orientation of tile
	keypress	output[north] = 1 if 'k' is pressed on the keyboard, 0 otherwise.

## Appendix B. Tile Codes

The following encoding is used by the creator and detector nodes.

1 wire	10 pusher	19 right_slider	28 store
2 crossover	11 keypress	20 left_slider	29 dir_detector
3 delta	12 thruster	21 equal	30 less_than
4 inverter	13 right_fuser	22 pulser	31 greater_than
5 and_gate	14 left_fuser	23 set_reset	32 toggle
6 or_gate	15 right_unfuser	24 creator	
7 nand_gate	16 left_unfuser	25 multiplier	
8 nor_gate	17 rotator	26 adder	
9 insulator	18 detector	27 negator	