

SWOS Picture Editor User Manual

v0.99.2023

Karakaš Zlatko <zlatko.karakas@gmail.com>

December 15, 2022

1 Intro

Thank you for using **SWOS Picture Editor**, or **SWPE**. It is a tool capable of editing/replacing (almost) any graphics in the game, by reading and writing original **SWOS** files.

The first iteration of the program sprang to life back in 2002., and in some way could be thought of as a piece of **SWOS** history. It was written primarily as a **SWOS** exploration and experimentation tool, especially for data files, and codifies the stuff I've discovered over the years. Over the time it has evolved randomly in many directions, has had pieces of **SWOS** code and various kludges slapped in and as a consequence user interface suffered: it's mostly keyboard based, although this version introduces mouse support for pitch editing. **PITCH** mode is the epicenter of the most recent additions and this is where you will probably get the most value out of the software.

This manual will list all the commands and show useful tips and ways of using the program effectively. In addition, there's also a [short demo video](#) demonstrating the new pitch editing features.

Be aware not to start **SWOS** while using **SWPE**, as it locks the graphic files and prevents **SWOS** from accessing them.

2 Program modes

The program is divided into 5 different modes: **SPRITE**, **PICTURE**, **PITCH**, **TILE** and **REPLAY**. Each mode has different use and commands. Press **F1** in any mode to see which keys are valid.

Commands available on all screens are:

F1	– toggle help
F5	– switch to sprites mode
F6	– switch to pictures mode
F7	– switch to pitch mode
F8	– switch to tile mode
F9	– switch to replay mode
ALT+ENTER	– toggle fullscreen mode ¹
ESCAPE	– end program
A	– about program

Main window can be moved by dragging it with the left mouse button (unless in pitch edit mode). You can minimize it by double clicking on main window while holding **CONTROL** key.

¹Fullscreen is broken, and there are no plans to fix it

2.1 Sprite mode

Sprite mode was all there was in the first version. The program used to be called **SWSPR** – **SWOS Sprite Viewer** in those days. Now it's just one of program modes used for viewing and replacing sprites. There are 1334 sprites in **SWOS**. Index file is **SPRITE.DAT**, and data files are **CHARSET.DAT**, **SCORE.DAT**, **TEAM1.DAT**, **TEAM2.DAT**, **TEAM3.DAT**, **GOAL1.DAT**, **BENCH.DAT**.

Commands:

SPACE/BACKSPACE	
ARROWS RIGHT/LEFT	– next/previous sprite
G	– go directly to sprite
PAGE UP/DOWN	– 10 sprites forward/backward
0..9	– change background color
HOME/END	– first/last sprite
+/-	– zoom/unzoom
F2	– save current sprite to bitmap
SHIFT+F2	– save all sprites
I	– toggle sprite information on/off
INSERT	– replace current sprite with corresponding bitmap
SHIFT+INSERT	– replace all sprites with corresponding bitmaps from current directory
CONTROL+R	– revert sprites to saved state
CONTROL+T	– switch between team2.dat and team3.dat

Arrow keys and space/backspace are provided for navigation within sprites. Use **G** for quick jump to specific sprite.

F2 saves sprite to disk in bitmap format for easy editing. If the sprite has center coordinates a text file with the same base name will be created, containing x center coordinate in the first line, and y in the second. If the sprite center is at (0, 0) text file will not be created.

INSERT key will cause replacement of an in-game sprite with external bitmap. Mass save and insert sprite features are provided by holding shift. **SHIFT+F2** will save all sprites to disk, and **SHIFT+INSERT** will read in as many sprites as possible from external files.

By default, sprites 644 – 946 are mapped to team2.dat. By pressing **CONTROL+T**, sprites 644 – 946 will show graphics from team3.dat, so it too can be edited. Pressing same keys again returns to team2.dat.

2.1.1 Replacing sprites

When inserting sprite, the program looks for bitmap filename in the form of **sprnnnn.bmp**, where **nnnn** is number of current sprite (e.g. **spr0007.bmp**). Bitmap must be 8-bit 256 color, without compression. Dimensions must be: $0 \leq \text{width} < 320$, $0 \leq \text{height} < 200$.

Recommended way of altering sprites is:

1. Find sprite you wish to change (you should see it in front of you)
2. Note its number (shown in the upper left corner)
3. Press **F2** – now the sprite is saved on your disk as a bitmap
4. Find the bitmap, which is in the same directory, and begins with “spr”, followed by four digit number of sprite
5. Load bitmap into some graphics editing program and make your changes
6. Go back to **SWOS Picture Editor**, find the same sprite again and press **INSERT** on your keyboard
7. Press **ESCAPE**, press **Y** when asked whether to save changes
8. Start **SWOS** and enjoy your new graphics...

Be careful with resizing sprites – this could have unpredictable results, especially when enlarging. I have tried it out, and with some sprites everything went okay, but with others **SWOS** tended to crash at random times.

When editing sprite bitmap in graphics editor there are couple things that you should be aware of. Transparent color will be color number 17 in palette. If you'd like to add some transparent areas to sprite, just draw with that color. You can also “delete” sprite by making it 100% transparent. Transparent color will be background color in the moment of saving, so you should pick some color not used in the sprite.

Also note that colors will be converted according to skin color of player and dress color of teams that are playing:

- 0 – no change
- 1 – no change
- 2 – no change
- 3 – no change
- 4 – skin color (light shade)
- 5 – skin color (normal shade)
- 6 – skin color (dark shade)
- 7 – turned to zero
- 8 – no change
- 9 – hair color (normal shade)
- 10 – shirt basic color
- 11 – shirt stripes color (swapped with 10 when vertical stripes)
- 12 – hair color (dark shade)
- 13 – hair color (light shade)
- 14 – shorts color
- 15 – socks color

2.2 Picture mode

Primary purpose of this mode is watching *.256 pictures. It is possible to save them to bitmaps by pressing **F2**.

Commands:

SPACE/BACKSPACE	
ARROWS RIGHT/LEFT	– next/previous picture
F2	– save picture as bitmap
F3	– reload picture
+/-	– brighten/darken picture

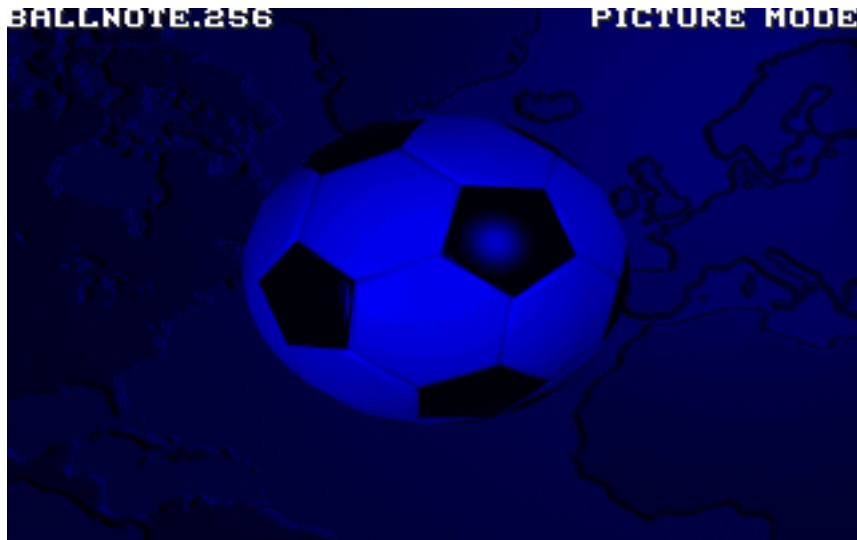


Figure 1: Picture mode

2.3 Pitch mode

This mode shows the pitch, can scroll through it and also edit it. Pitch is composed of 42 x 53 tiles, dimensions 16 x 16 pixels. By pressing **CONTROL+E**, edit mode is activated, and flashing cursor shown. See more in [2.3.1](#).

To show tile number for each tile press **CONTROL+N**. Press **CONTROL+N** again to turn it off. Pressing **CONTROL+U** shows usage count of every tile – the number of times it appears on the pitch. Pressing it again turns it off.

When saving and loading pitches, the filename pattern is simply:

`pitch<pitch number>.bmp`, e.g. `pitch3.bmp`.



Figure 2: Pitch mode



Figure 3: Pitch mode with tile numbers on

Commands:

ARROWS	– scroll pitch
PAGE UP/DOWN	– scroll up/down by about half of screen
SPACE/BACKSPACE	– go to next/previous pitch
HOME/END	– go to up-left/down-right corner of pitch
+/-	– change pitch type (frozen, muddy etc.)
CONTROL+N	– show/hide tile numbers
CONTROL+U	– show/hide tile usage count
CONTROL+Z	– undo
CONTROL+Y	– redo
CONTROL+S	– save changes
CONTROL+SHIFT+S	– force save (even if no changes)
CONTROL+R	– reload from disk
CONTROL+O	– optimize pitch tiles
CONTROL+E	– toggle edit mode
F2	– save pitch to bitmap
INSERT	– read pitch from corresponding bitmap

In edit mode:

CONTROL+ARROWS UP/DOWN	– change tile number plus/minus one
G	– input current tile number from keyboard
C	– change all instances of selected tile

2.3.1 Pitch editing

Pressing **CONTROL+E** switches pitch view into edit mode. The most notable visual change is display of tile grid, which makes it easier to see where the tiles are.

Key concept in edit mode is the cursor. It is flashing and designating the tile under it as the current tile. Many operations use current tile as a destination. The cursor can be moved around the pitch using the mouse or arrow keyboard keys.

Note that in this mode we are only changing which tile gets displayed within the pitch, and not the contents of the tiles themselves (pixels). For that see [Tile mode](#).

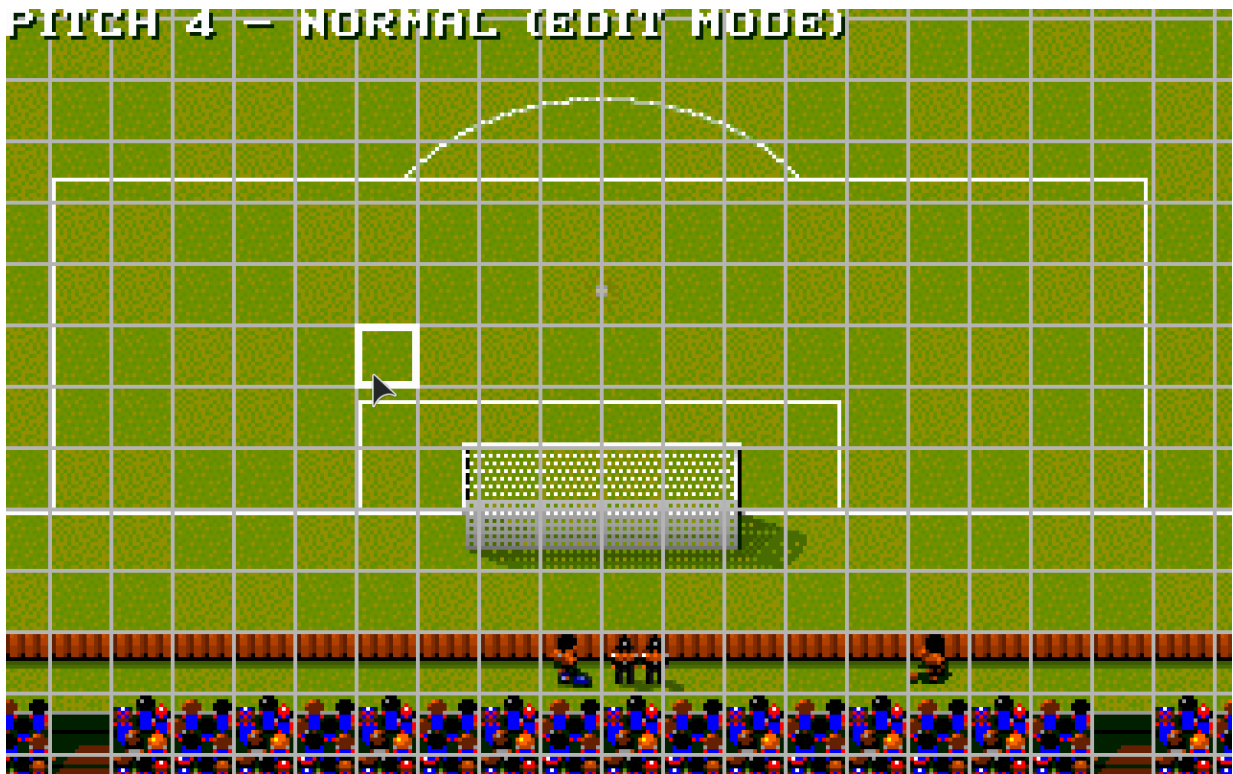


Figure 4: Pitch edit mode with cursor highlighting the current tile

2.3.1.1 Mouse navigation

Latest program version uses mouse for pitch mode navigation and editing. Left clicking the pitch and dragging will scroll it. Mouse wheel will scroll up/down.

When left clicking a tile while holding **CONTROL+SHIFT**, the tile becomes highlighted. It will be drawn with darker colors and outlined with a blue rectangle. Every tile having the same number on the pitch will also become highlighted. This can be very useful to differentiate between similar tiles, and to discover the leftover ones. To dismiss the highlight **CONTROL+SHIFT** left click on an already highlighted tile.

To temporarily show tile number of a single tile without turning on numbers for all tiles, click it with the middle mouse button. It will only display while the button is down. Same effect can be achieved by holding **TAB** key on the keyboard.

By **ALT** left clicking on the tile, tile mode opens with that tile displayed.

These operations, except for pitch dragging, work outside the edit mode as well.

2.3.1.2 Changing tiles

There are several options available to change tile numbers. In all cases except tile painting the first step is always to move the cursor to the tile you want to change.

- Cycling through tiles: **CONTROL+UP/DOWN** or **CONTROL**+mouse wheel
- Entering the tile number directly: **G** or mouse left button double click

- Change tile en mass **C**
- Tile painting: mouse right click or **SHIFT+ARROW** keys

CONTROL+UP/DOWN goes through available tiles at the selected spot one at a time. **UP** increases tile number and **DOWN** decreases it. This can be very useful if the tiles are sorted by the order of appearance. Same effect can be achieved with the mouse, by holding **CONTROL** and scrolling with the wheel. Scrolling up increases tile number, scrolling down decreases it.

If you need to a set tile to an exact number you can press **G** on the keyboard. **SWPE** will prompt for a number, and if it's valid assign tile with that number to the slot. Double clicking the tile with left mouse button achieves the same effect.

Pressing **C** will make **SWPE** ask for a tile number, and if valid, replace every occurrence of the selected tile with the newly entered number. For example, if the current tile number is 40, and entered number is 41, this action will cause every tile #40 in the pitch to change to #41.

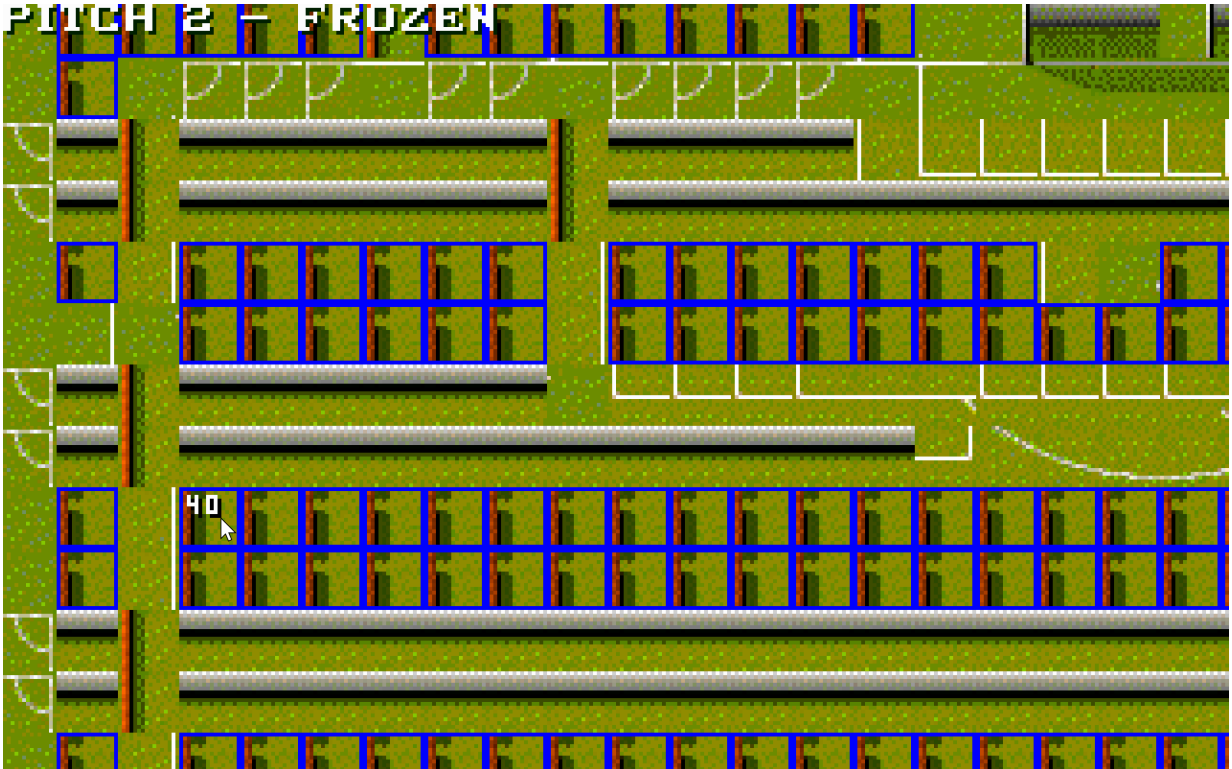


Figure 5: Tile #40 that's about to be mass-replaced highlighted

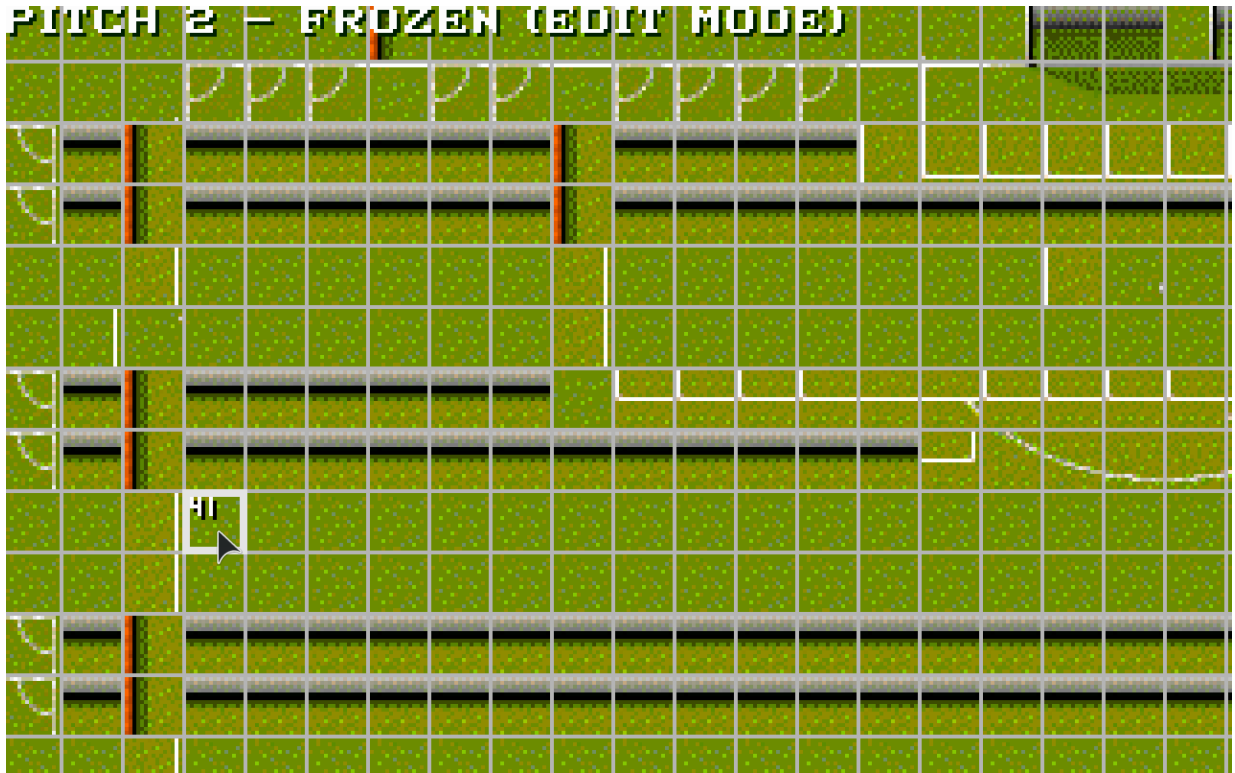


Figure 6: After replacing tile #40 with #41

2.3.1.3 Tile painting

Using the mouse you can do “tile painting”: click with the right mouse on a tile you want to duplicate, and drag. Initially clicked tile will get a red outline, specifying it as a master or “brush” tile. Any dragged upon tile becomes the same as the brush tile. Holding **SHIFT** key while dragging restricts “painting” to straight lines.

Alternative way of setting the brush tile is by **CONTROL**-left clicking on it. This is useful when the brush tile isn’t adjacent to the intended painting area.

Same operations can be done via the keyboard:

- pressing **CONTROL+ENTER** toggles tile brush
- **SHIFT+ARROWS** keys move the cursor and duplicate the master tile



Figure 7: Tile painting, note the brush tile highlighted with a red frame

2.3.1.4 Deleting tiles

Selected tile can be deleted by pressing **DELETE** key on the keyboard. Tile will be completely erased from the tile map. Every occurrence will be replaced with tile zero.

2.3.1.5 Pitch optimization

Pressing **CONTROL+O** activates pitch optimization. This will reassign tile numbers in the order of appearance on the pitch, starting from top left. Any left-over tiles not appearing on the pitch will be dropped. Note that animated tiles are ignored (effectively dropped).

This process creates minimal representation of the current pitch and usually comes at the end, when you're satisfied how the things are looking.

2.3.1.6 Saving changes

To save pitch changes to the disk press **CONTROL+S**. In case no changes have been detected it will not do anything. To force the save even if there are no changes press **CONTROL+SHIFT+S**.

2.3.1.7 Undo/redo

Any pitch operation can be undone and redone practically unlimited number of times. **CONTROL+Z** undoes and **CONTROL+Y** redoes last action.

It's a recent addition to the program and hasn't been extensively tested. Save your work often to be on the safe side.

2.4 Tile mode

This mode is for viewing and replacing individual tiles. Tiles are displayed in the center of the window with their number, pitch number and usage count shown top-left. Unused patterns are marked with ****UNUSED****.

To edit tile pixels use the following workflow:

- navigate to the desired pitch and tile
- use **F2** key to export the tile to bitmap
- edit the bitmap in a graphics editor
- import the bitmap back into the tile via **INSERT** key

The exported filename follows the pattern: **pt**<*pitch number*>-<*tile number*>.bmp, where <*pitch number*> is a single digit and <*tile number*> is four digits zero padded, for example: **pt2-0023.bmp**.

Same filename pattern is used when importing. There is no undo command for importing tiles, so use it with care. For more details see [Replacing tiles](#).

“Insert all tiles” command will look in game directory, and insert first continuous range of bitmaps following the forementioned filename pattern.

In case you want to add more tiles than the pitch is currently containing you can use **CONTROL+INSERT** to add new empty tile, which can then be used as an import tile target. This is meant for adding a small number of tiles, to add large quantity use **SHIFT+INSERT** command.

When highlighting a tile, it will also be highlighted in pitch mode, and vice versa. When changing tile number, old and new tile simply swap places. Tile deletion works the same way as in **PITCH** mode and can be undone.

Commands:

ARROWS RIGHT/LEFT	– next/previous pitch
ARROWS UP/DOWN	– next/previous tile
HOME/END	– first/last tile
PAGE UP/DOWN	– 10 tiles forward/backward
SPACE/BACKSPACE	– change pitch type
F2	– save tile to bitmap
SHIFT+F2	– save all tiles to bitmaps
G	– go to tile
C	– change tile number
H	– highlight current tile
DELETE	– delete current tile
INSERT	– replace current tile with corresponding bitmap
SHIFT+INSERT	– insert all tiles in game directory
CONTROL+INSERT	– insert new empty tile at the end
CONTROL+Z	– undo
CONTROL+Y	– redo

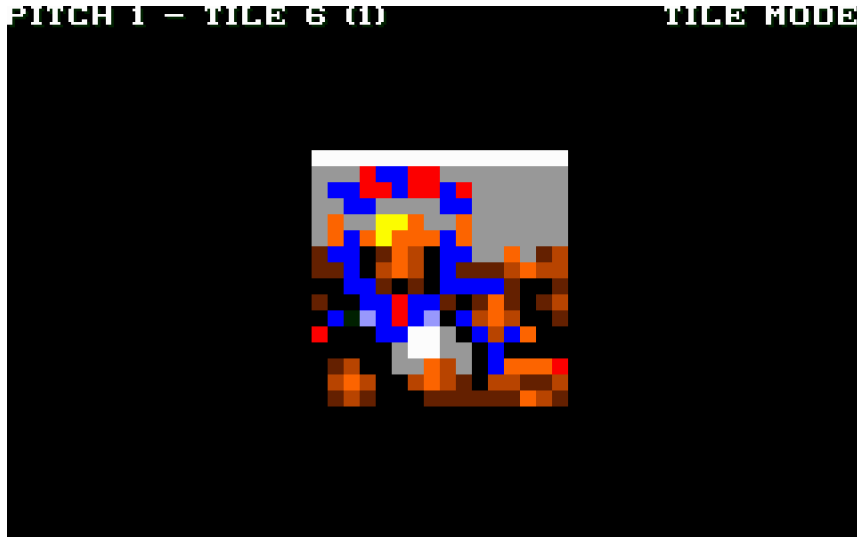


Figure 8: Tile mode

2.4.1 Replacing tiles

Process of replacing tiles is similar to replacing sprites. Here the only allowed bitmap dimension is 16 x 16. Bitmap data are simply copied into corresponding place in the pitch file. Also index of the tile can be changed in pitch mode.

When editing a tile in graphics editor, note that colors 10 and 11 will be converted. Color no. 10 will be shorts color if team is using one color shirt, or shirt basic color if not, and color 11 will always be primary shirt color. Only 42 tiles have this color converting property, and those are mostly crowd tiles. Those colors are red and blue, and you can see in **PITCH** mode that fans are mostly covered with it.

Pitch types are achieved by palette modification. Palette entries: 0, 7, 9, 78, 79, 80, 81, 106 and 107 are changed depending on pitch type accomplishing visual changes according to conditions.

Tiles 1 to 24 have special properties - they are animated crowd. Tiles 1..12 are upper crowd, and tiles 13..24 are lower crowd. Odd indices are pictures when not moving, and even indices are pictures when animated (jumping, cheering, etc.). Tile 0 is usually empty tile (except in training pitch).

You can replace individual tiles without any special considerations, but inserting whole pitch from bitmaps requires some attention. Following does not hold for training pitch, which does not have crowd, and therefore animated tiles. When some pitch with crowd is inserted, tile zero will be set to empty tile (filled with zeros), and tiles from 1 to 24 will be set: 1 = 2, 3 = 4, etc., every odd and following even number will hold the same tile. By default, this will turn animation off. To re-enable it, use change tile number feature and set tiles that will be animated into odd tiles 1..24. Even tiles must be inserted manually.

Important thing to know when inserting bitmaps is that maximum allowed number of unique tiles on pitch is 296. **SWOS** uses only 75,776 bytes of memory for tiles, and if that limit is exceeded each tile with index greater than 295 will be overwritten.

2.5 Replay mode

This mode is in fact external replayer for **SWOS** replay and highlight files. In main menu you choose between highlight and replay files. If there are files of selected kind in directory, you will be presented with a menu with names of files. Bring cursor to file you wish to watch and press enter. Hopefully, program will start replaying file.

Shortcut keys available in menus, apart from arrows, are:

BACKSPACE	–	return to previous menu (if any)
HOME	–	go to first menu entry
END	–	go to last menu entry

While replaying, use following commands:

ESCAPE	–	stop replaying
F	–	toggle frame rate display
+/-	–	speed up/slow down replay
P	–	pause replay
N	–	(while paused) step one frame

Please note that the replayer is not completely finished at the moment. There are still a few things missing such as sound, sliding advertisements, animated fans and their colors.



Figure 9: Replay mode

3 Known bugs/limitations

This program is provided as is, there is no support and there will definitely be no more official releases (but never say never – like, for example right now).

Some of the latest changes were quickly hacked in and the program might crash occasionally. If it does crash really badly and bothers you a lot, you can try messaging me. I probably won't fix it, but if you include detailed info with screenshots, videos and steps to reproduce, miracles might happen. There is also file named `errlog.txt` in the program directory which could be helpful.

With all that said, I hope the program can still prove to be useful, and I know poor looks and badly behaving interfaces were never enough to stop **true SWOS maniacs** :)

Wishing you all the best,
Z.