

Software Requirements Specification
Game of “Walris”
Robert Keller
17 January 2005

Walris is a video game for one or two cooperative players. While it has some similarities to Tetris, it also has some essential differences.

Players modify the position and orientation of shapes that are launched toward each other from the left and right of a play area on the screen. The shapes are rectilinear and composed of uniform square tiles. The shapes may collide with each other and with other tiles that are already on the screen. When shapes collide, they stop in place and disintegrate into the tiles that compose them. A shape that completely crosses the screen without colliding is taken out of play. When a vertical wall of tiles completely fills the play area from top to bottom, the wall is removed and the tiles on one side or the other fill in. Specifically, if the wall is to the left of center, then the tiles on the left move, while if the wall is to the right of center, then the tiles on the right move. The game ends when a launched shape immediately collides with a tile. Shapes are launched by randomly selecting a pattern from a repertoire of patterns. The patterns are specified in a file `shapes.txt`, as described below.

A player or players controls the position and orientation of the shapes using keys on the keyboard. Each shape is controlled by its own set of keys, so that two players can play, or one player can play both shapes simultaneously. The binding of keys to functions is defined by a configuration file `keys.txt`. It is anticipated that the bindings will be chosen so that two players can play without interference. Also, the layout of the functions of the keys would normally be the same for both shapes.

The following functions are defined for each shape:

- Move the shape up.
- Move the shape down.
- Rotate the shape clockwise.
- Flip the shape to its horizontal mirror image (as if the mirror were placed vertically).
- Accelerate the shape by a defined increment.

In addition, there shall be the following keys available:

- Increase the launch speed for both shapes.
- Pause or resume the game.
- Open help window that describes the key bindings and scoring information.
- Restart the game (from a paused state, or from demo mode).
- Start demo mode (from a paused state), as described below.

The game starts in the paused state.

Scoring is as follows:

- For each tile that becomes stationary due to a collision, the speed is recorded at the time of collision.
- Speeds range from 1 to 10 tile spaces per second, and the speed is displayed on the screen.
- When a wall is formed, the score is incremented by the recorded speed of all tiles in the wall.
- When a shape moves off the screen without colliding, the score is decremented by the speed of the shape times the number of tiles in the shape.
- The current score is displayed on the screen.

The game maintains a list of the top 10 scores, along with their players, in a file `scores.txt`. When, at the end of a game, a new score in the top 10 is achieved, a player is asked to enter identification.

Sampling of the input and shape motion is done in terms of discrete “ticks”.

A configuration file `config.txt` is used to set various parameters of the game, including:

- The duration of a tick in milliseconds (use 10).
- The width of the play area in tiles (use 25).
- The height of the play area in tiles (use 12).
- The height and width of a tile in pixels (use 20).
- The default launch speed (use 1).

These are simply to be read in the order given, without any specific identification.

The game can be driven by an input event script contained in a file, in lieu of specific key strokes. This is known as *demo mode*. Specifically, an input event script specifies a list of pairs, with each pair being a number of ticks (> 0) since the previous even, followed by a function number. The demo script resides in a file `demo.txt`. The function numbers are as below:

0. move-left-up
1. move-left-down
2. rotate-left
3. flip-left
4. accelerate-left
5. move-right-up
6. move-right-down
7. rotate-right
8. flip-right
9. accelerate-right
10. increase-launch-speed
11. pause-resume
12. open-help
13. restart
14. demo

The function numbers are also used to structure the key binding file, `keys.txt`, which contains a list of key characters, in the order of functions to which they are bound, as listed above.

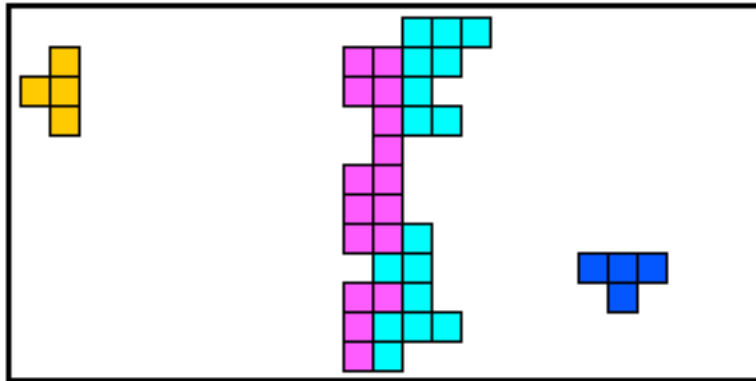
The structure of the `shapes.txt` file is a list of pairs of relative shape coordinates, one per line, with the two coordinates for each tile in succession. For example, the following shapes might be used, which defines an “ell” shape, a “tee” shape, a “block” shape, and a linear shape, respectively

```
0 0    1 0    2 0    2 1
0 0    1 0    2 0    1 1
0 0    1 0    0 1    1 1
0 0    0 1    0 2    0 3
```

An applet for an earlier version of Walris can be found at:

<http://www.cs.hmc.edu/~keller/javaExamples/walris.html>

The main difference between the earlier version and the one specified here is that the earlier version is only one player, and there is a single set of keys controlling both shapes. The “focus” is changed between the shapes by a key. Whichever shape has the focus is colored differently.



Walris play area, showing tiles from disintegrated shapes that almost form a wall (in center), along with left and right shapes moving in opposite directions.