

ALE Stdin/Stdout Protocol

Marc G. Bellemare, Joel Veness

April 17, 2013

1 Overview

This document describes the agent/environment protocol for agents communicating with ALE via stdin/stdout. The protocol is text-based and allows the possibility of run-length encoding the screen.

2 Protocol

After preliminary handshaking, the protocol enters a loop in which ALE sends information about the current time step and the agent responds with both players' actions (in general agents will only control the first player). The loop is exited when one of a number of termination conditions occurs.

2.1 Handshaking

ALE first sends the width and height of its screen matrix as a hyphen-separated string:

```
www-hhh\n
```

where `www` and `hhh` are both integers.

The agent then responds with a comma-separated string:

```
s,r,k,R\n
```

where `s`, `r`, `R` are 1 or 0 to indicate that ALE should or should not send, at every time step, screen, RAM and episode-related information (see below for details). The third argument, `k`, correspond to a desired frame skip: if greater than 0, then `k` frames are skipped between every observation.

Note. Frame skipping is currently incompletely supported and should be set to 0.

2.2 Main Loop – ALE

After handshaking, ALE will then loop until one of the termination conditions occurs; these conditions are described below in Section 2.4. If terminating, ALE sends

DIE\n

Otherwise, ALE sends

<RAM_string><screen_string><episode_string>\n

Where each of the three strings is either the empty string (if the agent did not request this particular piece of information), or the relevant data terminated by a colon.

2.2.1 RAM_string

The RAM string is 128 2-digit hexadecimal numbers, with the i^{th} pair denoting the i^{th} byte of RAM; in total this string is 256 characters long, not including the terminating ‘:’.

2.2.2 screen_string

In “full” mode, the screen string is $\mathbf{www} \times \mathbf{hhh}$ 2-digit hexadecimal numbers, each representing a pixel. Pixels are sent row by row, with \mathbf{www} characters for each row. In total this string is $2 \times \mathbf{www} \times \mathbf{hhh}$ characters long.

In run-length encoding mode, the screen string consists of a variable number of (colour,length) pairs; denoting a run-length encoding of the screen, also row by row. Both colour and length are described using 2-digit hexadecimal numbers. Each pair indicates that the next length + 1 pixels take on the given colour; run length is thus limited to 256. Runs may wrap around onto the next row. The encoding terminates when the last pixel (i.e. the bottom-right pixel) is encoded. The length of this string is 4 characters per (colour,length) pair, and varies depending on the screen.

In either case, the screen string is terminated by a colon.

2.2.3 episode_string

The episode string contains two comma-separated integers indicating episode termination (1 for termination, 0 otherwise) and the most recent reward. It is also colon-terminated.

2.2.4 Example

Assuming that the agent requested screen, RAM and episode-related information, a string sent by ALE might look like:

000100...A401B2:3C3C3C3C00003C3C3C...4F4F0000:0,1:\n

^ 2x128 characters ^ 2x160x210 characters ^ongoing episode, reward of 1

2.3 Main Loop – Agent

After receiving a string from ALE, the agent should now send the actions of player A and player B. Both are sent as a pair of comma-separated integers on a single line, e.g.:

```
2,18\n
```

where the first integer is player A’s action – FIRE – and the second integer, player B’s action – NOOP. Emulator control (reset, save/load state) is also handled by sending a special action value as player A’s action. See Appendix for list of actions.

2.4 Termination

ALE will terminate (and potentially send a DIE message to the agent) whe one of the following conditions occur:

- `stdin` is closed, indicating that the agent is no longer sending data, or
- The maximum number of frames (user-specified, with no maximum by default) has been reached.

ALE will send an end-of-episode signal when one the following is true:

- The maximum number of frames for this episode (user-specified, with no maximum by default) has been reached, or
- The game has ended, usually when player A loses their last life.

Once the end of episode is reached, no further emulation takes place. This avoids odd situations where the player can reset the game through button presses, or where the game normally resets itself after a number of frames. This makes for a cleaner environment interface.