| Circuit-to-game messages | Acknowledgement | Description |
|---|---|---|
| **card** | None | Sent when the player scans their badge on the reader. Is repeatedly sent while the badge remains on the reader. The game can ignore such messages in some situations. |
| **init:ok** | None | Acknowledgement message sent for an “init” message. Informs the game that the circuit and badge were successfully initialized. |
| **write:ok** | None | Acknowledgement message sent for a “write” message. Informs the game that the circuit has successfully written data into the badge. |

| Game-to-circuit messages | Acknowledgement | Description |
|---|---|---|
| **init** | init:ok | Sent to the circuit when level 3 starts. Allows to initialize the data in the badge with known values. The game waits for the “init:ok” response. |
| **init:cancel** | None | If the player does not have either a badge or the card reader, level 3 cannot start. The player is forced to return to the game’s main menu. This message is then sent to the circuit to cancel the initialization phase. |
| **terminal:<terminal_number>** | None | The game sends a terminal number (a positive integer) to display on the LCD. |
| **terminal:none** | None | Sent when the player exits from the state of being in front of a terminal. Allows to clear the LCD (nothing to display). |
| **id:<employee_id>** | None | The game sends the employee ID (a sequence of 6 positive integers) currently stored in the player’s badge. The ID gets printed by the LCD for the player to check it. |
| **card:ok** | None | Sent only if the last message received by the game is a “card” message (the player is scanning their badge), the player’s avatar is in front of a terminal, and the employee ID in the badge matches the one required to activate the terminal. |
| **card:ko** | None | Same as “card:ok” except the IDs do not match. |
| **hacked** | None | Sent when the player hacks a terminal. |
| **write** | write:ok | Sent when the player has hacked a terminal and wants to change the employee ID stored in their badge. The game waits for a “write:ok” response. |
| **write:cancel** | None | If the player decides to cancel the badge reconfiguration, this gets sent to the circuit to halt all write operations on the badge. |