CYBATHLON

Brain-Computer Interface Race

Manual for

CybathlonBrainrunnersTraining1.225

In collaboration with Zurich University of Arts, Specialization in Game Design

KEY MODIFICATIONS COMPARED TO PREVIOUS VERSION OF BRAINRUNNERS (0.9351)

- **Game mechanics on action pads: Correct commands can be aborted by incorrect commands; incorrect commands can be corrected by correct commands.**
- **Balancing: Acceleration-deceleration originating from correct/incorrect commands changed**
- **Configuration options improved**
- **OnChangePad feature (for training version only): A pad label is sent from BrainRunners computer to Player-IP when the avatar enters a new pad. This feature will not be available in the official race version at the Cybathlon.**
- **Game speed control (for training version only): For training purposes, the velocity of the gameplay can be set manually.**
- **Level control (for training version only): For training purposes, levels can be defined manually.**

IMPORTANT NOTES

1) **Details of the *Cybathlon* BCI racing game *BrainRunners* will be subject to modification.**
2) **Please order *BrainRunners* via competition@cybathlon.com.**
3) **Please find the rules on the BCI Race on www.cybathlon.com.**

FEEDBACK AND QUESTIONS

If you have any feedback or question on the current version of *BrainRunners* or BCI Race rules, please contact competition@cybathlon.com.

1. General Introduction to BrainRunners

BrainRunners is a multiplayer “running” game developed for the BCI Race of the *Cybathlon*. Up to four pilots in each race are equipped with brain-computer interfaces (BCIs) that enable them to control an avatar in the game. Each avatar moves forward by itself, and thus, reaches the finish line of the race even without input signals from the pilot. The pilot has to send appropriate commands using the BCI within the correct time frame while virtually walking on dedicated areas indicated by colored “pads”. Giving the correct command at the correct time or not sending a command if no command is desired accelerates the avatar, thus providing the pilot with an advantage over her or his competitors by moving the avatar closer to the finish line faster. Pilots can trigger the avatar to accelerate on a “SPEED” action pad (cyan), jump over prickles on a “JUMP” action pad (magenta), or roll below rays on a “ROLL (yellow)” action pad (Figure 1). Thus, a maximum of three different commands can be sent from the BCI within the games, but not every command has to be implemented. One command is sufficient to play the game, but the more different commands the pilot can fire and the BCI system can detect, the faster the avatar arrives at the finish line. Incorrectly chosen commands (e.g. JUMP instead of SPEED) or incorrectly timed commands yield a disadvantage, i.e. the avatar decelerates and loses time. On the “NOINPUT” pad, pilots accelerate if they manage not sending any input and decelerate if they send a command accidently.

Four pilots can simultaneously compete in the same race. Every pilot’s avatar gets a track. Each pilot sits in front of a separate screen to play the game and sees his or her avatar as well as the three competing avatars. The pilot whose avatar first crosses the finish line wins the race.

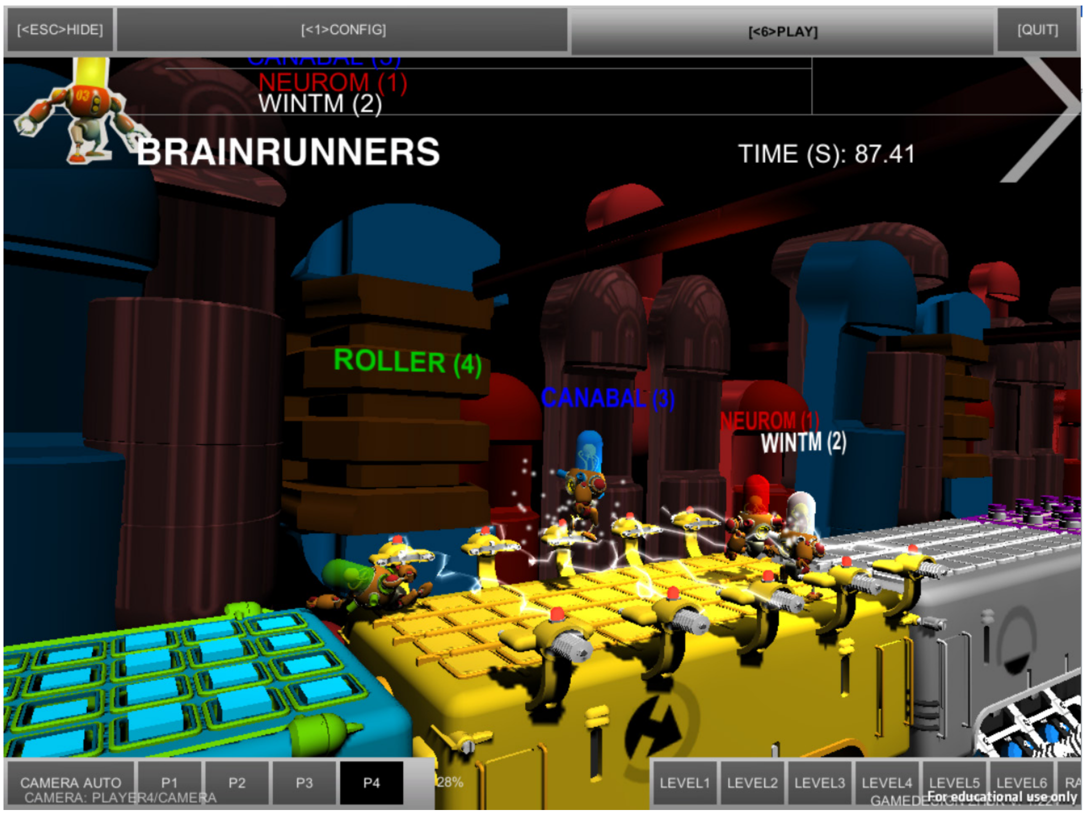


Figure 1 Screen shot of BrainRunners. Cyan PAD: SPEED. Yellow PAD: ROLL. Gray PAD: NOINPUT. Magenta PAD: JUMP.

2. Actions

There are three different types of action pads on the track, i.e. SPEED, JUMP and ROLL pads, and one pad where no action is desired, i.e. the NOINPUT pad. The pilot has to trigger (not trigger) the right action on the PADs to get an advantage over the other competitors.

Action	***Description***
SPEED/JUMP/ROLL	As soon as the avatar is on the SPEED/JUMP/ROLL pad, the SPEED/JUMP/ROLL command should be sent to accelerate the avatar. The SPEED/JUMP/ROLL boost lasts until the end of the pad is reached or until another command is received. Thus, a correct action can be aborted by sending an incorrect command, or an incorrect action can be corrected by sending the correct command. The goal is to send the correct command as early as possible after having reached the pad. The earlier it is received, the larger the time advantage the avatar can get.
NOINPUT	On the NOINPUT pad, no command should be sent to benefit from a higher velocity. If an incorrect command is received, the avatar moves with a lower velocity for a certain duration (see Section 4) or until the next pad is reached.
INCORRECT	Any incorrect command received, i.e. any command on a NOINPUT pad, or incorrect command on an action pad, e.g. a ROLL command on a SPEED pad, makes that pilot's own avatar slow down, i.e. the avatar loses time.
Transition to the subsequent pad	Every time the avatar transitions to a subsequent pad, all the commands/actions are cleared. The base velocity for the reached pad is set until a new command is received.

- **At the *Cybathlon 2016*, the order of the pads will not be known to the teams before the race.**
- **Each pad will appear four times, i.e. four times SPEED, four times JUMP, and times ROLL, and four times NOINPUT.**

3. Configurations and Protocol

3.1 Configurations

BUTTONS	***Description***
# Starting	
[CONFIG]	Configure players, names, interfaces
[PLAY]	Start game
[ESC]	Toggle configuration menu
# Pads	
[SPEED]	Use speed command
[JUMP]	Use jump command
[ROLL]	Use roll command
[NOINPUT]	Use no command
# Player	
[PLAYER 1/2/3/4]	Add/remove player from game
[BCI-INPUT/BOT]	Switch between BCI-INPUT mode and BOT (simulation) mode for players 2-4. Simulation can be overruled by key input.
[skill]	Select skill level of players 2-4 in BOT mode
# Levels	
[LEVEL 1+2]	Fields: [SPEED,JUMP,ROLL,EMPTY]
[LEVEL 3+4]	Fields: [SPEED,JUMP,EMPTY]
[LEVEL 5]	Fields: [JUMP,EMPTY]
[LEVEL 6]	Fields: [SPEED,EMPTY]
[LEVEL RANDOM]	Fields: random
Text field on start up	The level that appears when the game is started can be specified in a text file. See "# Define own levels" under Protocol.
# Camera	
[CAMERA AUTO]	Automatic view adaption
[P1]	Player 1 view
[P2]	Player 2 view
[P3]	Player 3 view
[P4]	Player 4 view
# Background	
[PILOT VERSION (NO BACKGROUND PARTICLES)]	Switch to black background (for training purposes only, option not available for the competition)
[DEACTIVATE KEYS FOR SIMULATION]	Deactivate keyboard inputs (e.g. in BOT mode)
[SWITCH TO CLASSIC VIEW]	Play game with original game graphics
# Game speed control	
[←],[→], [space]	Change the velocity of the avatar, i.e. of the gameplay. By default, the velocity is set to the official velocity (100%). By pressing the buttons left and right, you can slow down or speed up the velocity for training purposes. Pressing the space key sets the velocity back to default (100%). The current velocity settings are indicated at the bottom of the screen when changed. However, the game may not work correctly for very high speeds.

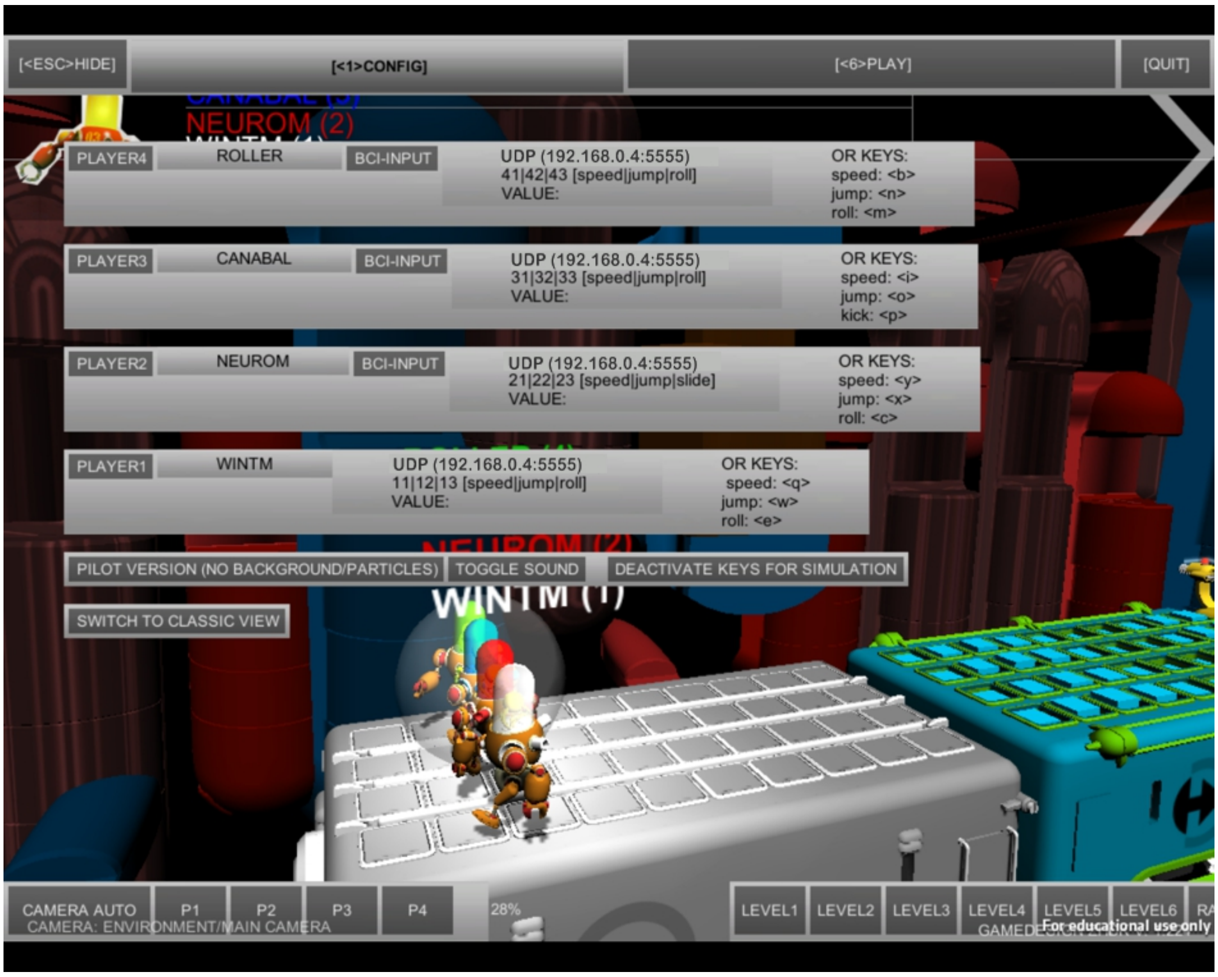
3.2 Protocol

# System requirements
Windows (Tested: 32 or 64 Bit –Windows 7+) or Mac (Tested: 10.6+)

# Interface BCI:
UDP (standard), see Figure 2

Player	***SPEED input value***	***JUMP input value***	***ROLL input value***
1	11	12	13
2	21	22	23
3	31	32	33
4	41	42	43

Protocol: Send command as 1 Byte BigEndian



```
PLAYER4  ROLLER  BCI-INPUT  UDP (192.168.0.4:5555)              OR KEYS:
                            41|42|43 [speed|jump|roll]          speed: <b>
                            VALUE:                              jump: <n>
                                                                roll: <m>

PLAYER3  CANABAL  BCI-INPUT  UDP (192.168.0.4:5555)             OR KEYS:
                             31|32|33 [speed|jump|roll]         speed: <i>
                             VALUE:                             jump: <o>
                                                                kick: <p>

PLAYER2  NEUROM  BCI-INPUT  UDP (192.168.0.4:5555)              OR KEYS:
                            21|22|23 [speed|jump|slide]         speed: <y>
                            VALUE:                              jump: <x>
                                                                roll: <c>

PLAYER1  WINTM  UDP (192.168.0.4:5555)              OR KEYS:
                11|12|13 [speed|jump|roll]           speed: <q>
                VALUE:                              jump: <w>
                                                    roll: <e>

CAMERA: ENVIRONMENT/MAIN CAMERA
```

Figure 2 Player configuration and command definition

# Tests via shell (Unix, Linux, Mac)

```
// PLAYER 1:
SPEED:        printf '\x0B' | nc -u 192.168.0.4 5555
JUMP:         printf '\x0C' | nc -u 192.168.0.4 5555
ROLL:         printf '\x0D' | nc -u 192.168.0.4 5555

// PLAYER 2:
SPEED:        printf '\x15' | nc -u 192.168.0.4 5555
JUMP:         printf '\x16' | nc -u 192.168.0.4 5555
ROLL:         printf '\x17' | nc -u 192.168.0.4 5555
```

# Tests via MATLAB® (tested with R2014a)

```
hudps = dsp.UDPSender('RemoteIPAddress','192.168.0.4','RemoteIPPort',5555);

% PLAYER 1:
step(hudps, uint8(11));        % SPEED Player 1
step(hudps, uint8(12));        % JUMP Player 1
step(hudps, uint8(13));        % ROLL Player 1

% PLAYER 2:
step(hudps, uint8(21));        % SPEED Player 2
step(hudps, uint8(22));        % JUMP Player 2
step(hudps, uint8(23));        % ROLL Player 2

release(hudps);
```

# Define own levels

You can define your own levels with *level.csv* (see execution folder). If existent the optional level.csv file will be loaded automatically when BrainRunners.exe is started. You can define your own levels, by renaming *_level.csv* to *level.csv*, changing the pad labels and restarting the game.
The level.csv contains the pad labels, separated by a ";"-character without spaces. Ensure with a text editor not to add a space or break line after the last number (some versions of excel do that by default).

Example contents level.csv
1;2;3;1;2;3;0;1;2;0;3;1;3;2;2;1

Where the pad labels correspond to:

NOINPUT Pad (Grey)	***SPEED Pad (Cyan)***	***JUMP Pad (Magenta)***	***ROLL Pad (Yellow)***
0	1	2	3

# IP Filtering and OnChangePad Labels

The optional file *ip.csv* (see execution folder) is used to specify the IP address of the player/BCI computer.

If the file ip.csv exists, the BrainRunners.exe will filter input commands for avatars coming from the specified IP, e.g. 11, 12, 13 will only be accepted from the IP of player 1. Additionally, whenever an avatar enters a new field a pad label will be sent to the player's IP address. This feature was asked for easier data labelling during training, but will be deactivated in the official competition. If you do not want to use the ip.csv, you can rename it to e.g. _ip.csv or delete it.

The OnChangePad label will be sent as an UDP package (of 1 Byte BigEndian) to the player IP and port 6666. It contains a number 0, 1, 2, or 3 for the pad label the avatar enters. The same labeling is used as for the level definition (see table above).

The ip.csv must contain the IPs of up to four players' computers separated by a ";"-character. Ensure with a text editor not to add a space or break line after the last number (some versions of MS Excel do that by default).

Example contents ip.csv
192.168.0.10;192.168.0.20;192.168.0.30;192.168.0.40

Where the first IP 192.168.0.10 would be the IP of player 1, 192.168.0.20 of player 2 and so on.

# Results and Result History

You can find the results of any completed or aborted race in a text file (see execution folder after completion of a race) named "resultTIMESTAMP.csv. (The same results file is also saved redundantly into a folder called "ResultHistory".)

# gameevents.csv

Ignore the file "gameevents.csv". It contains some unsupported debugging information from the game developer.