Underwaterrugby (UWR) 3D simulation and visualization application for training and interactive positioning discussion purpose for player and referee: April29 2018.

It visualize positioning of players in a swimmingpool in 3D (interaction area of player simplified with blue / white spheres; red ball). Furthermore,

- **It creates images (ending png) for showing positioning of players in the swimmingpool field, inclusive dimension/positions of the swimmingpool, red rugby ball and the baskets at the swimming pool bottom.**
- **It creates positioning files (ending csv) which can be reloaded for dynamic simulation of UWR games. During this running simulation, 3D rotation of the field can be done with the mouse.**
- **It creates video (ending mp4) for showing movement of players in the swimmingpool field.**

The “matplotlib” functionality makes 3D rotating and picture saving possible.

Author pascaldagornet@yahoo.de

Based on underwaterrugby rules available on www.vdst.de and referee training in baden-wuerttemberg by kneer@gmx.net

No warranty: all sport recommendations/rules of www.vdst.de remain valid.

Application tested on

1. Linux 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64 GNU/Linux on a Notebook Lenovo T560 screen resolution 1920x1080 GNOME desktop with the latest packages python3 python3-matplotlib gtk+3.0 python3-cairocffi python3-numpy python3-pygobject python3-gi python3-gi-cairo (not exhaustiv).
2. Linux Debian Stretch LXDE9.3 32bit screen 1280x1024 Pentium4 with the packages needed in (1)
3. Linux Debian Stretch Kernel 4.9 32bits 1280x800 Toshiba Satellite Pro

Application NOT tested for now on

1. Windows7 or Windows10 PCs
2. Rasperry PI

Remark: PCs of high spec allow a quicker picture move. If a low spec PC is used, only slow motion will be seen.

Installation advice

The application consist of 2 files

- game_uwr.py
- game_uwr.glade

The file game_uwr.glade was created with the application “GLADE” for GUI creation in GTK+
It can be placed in any directory; please identify the path: example, /home/uwr_game/.

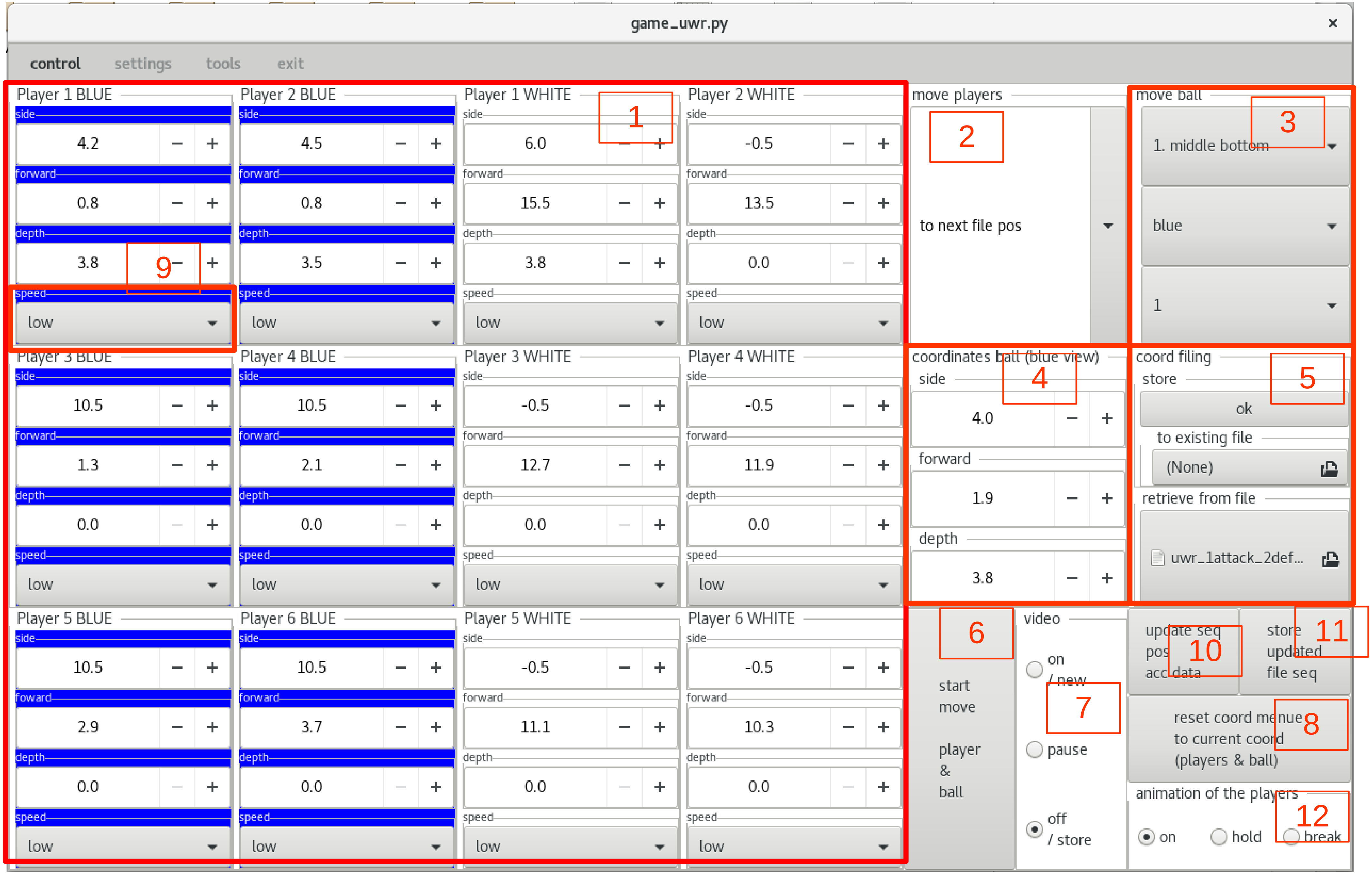
The file game_uwr can be placed in any directory; please identify the /pathtothescript/. It will be started from a terminal/console in this path with the command "python3 /pathtothescript/uwr_game.py".

Before the application starts, do following,

1. create the directory /home/family/Bilder in your PC or adapt the python script game_uwr.py with the new location for storage of images and video. The script area of uwr_game.py to be adapted (edit it and change it with a text editor), are localized between strings of type ”*****************”.
2. modify in the uwr_game.py the path of the game_uwr.glade file. The path can be seen below the “`class fooclass:`” and is between strings of type ”*****************”.
3. install all necessary packages in your PC: python3 python3-matplotlib gtk+3.0 python3-cairocffi python3-numpy python3-pygobject python3-gi-cairo (not exhaustiv)

Start the application with "python3 /pathtothescript/uwr_game.py" from a terminal console.

Startmenue “control”



	side	forward	depth	speed
Player 1 BLUE	4.2	0.8	3.8	low
Player 2 BLUE	4.5	0.8	3.5	low
Player 1 WHITE	6.0	15.5	3.8	low
Player 2 WHITE	-0.5	13.5	0.0	low
Player 3 BLUE	10.5	1.3	0.0	low
Player 4 BLUE	10.5	2.1	0.0	low
Player 3 WHITE	-0.5	12.7	0.0	low
Player 4 WHITE	-0.5	11.9	0.0	low
Player 5 BLUE	10.5	2.9	0.0	low
Player 6 BLUE	10.5	3.7	0.0	low
Player 5 WHITE	-0.5	11.1	0.0	low
Player 6 WHITE	-0.5	10.3	0.0	low

move players: to next file pos

move ball: 1. middle bottom; blue; 1

coordinates ball (blue view)
side: 4.0
forward: 1.9
depth: 3.8

coord filing
store: ok
to existing file: (None)
retrieve from file: uwr_1attack_2def...

start move player & ball

video
○ on / new
○ pause
● off / store

update seq pos acc data

store updated file seq

reset coord menue to current coord (players & ball)

animation of the players
● on ○ hold ○ break

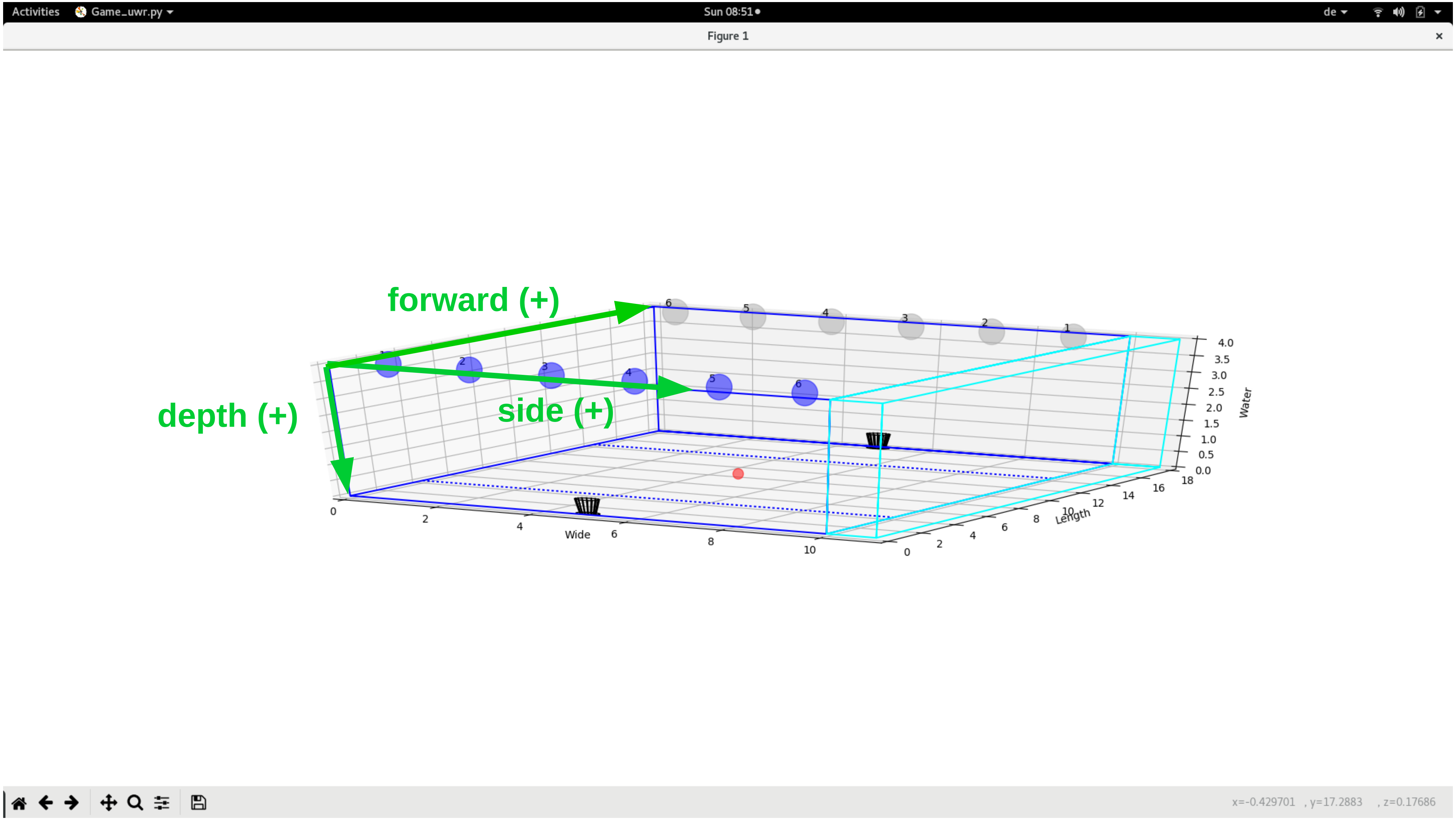
Explanation of startmenue “control” (1/8):

(1) Entries for new player 3D coordinates.

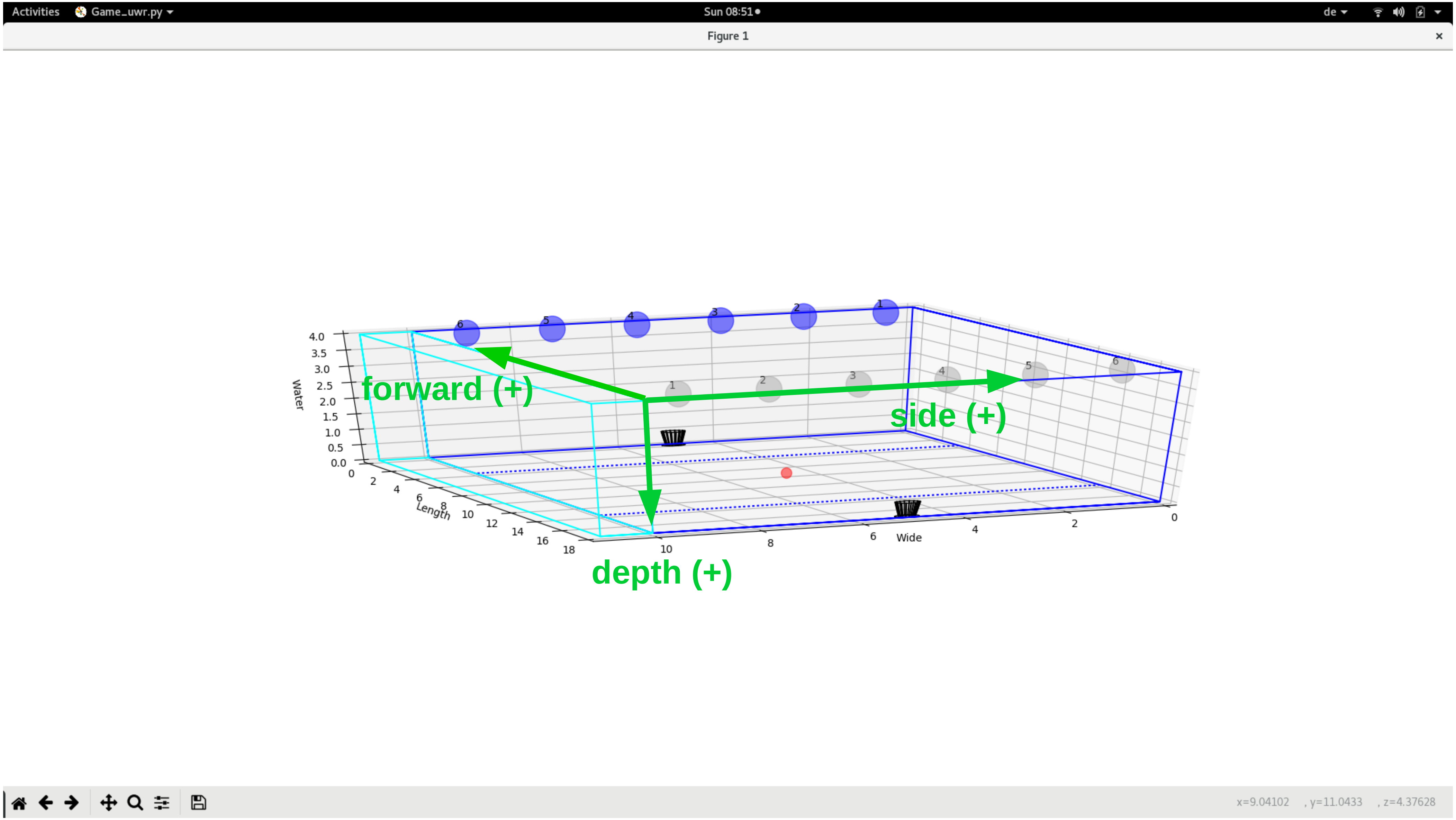
The value is positive for the blue team: front positioning, positioning to the right and positioning below (maximum 4m). In order to move the players to the given position, click on (6) when at (2) “to menu coord” is selected.

For the white team: negative value can be given for the white players in case, for example, they are outside the field left in the exchange area observing a penalty.

Startmenue “control”, coordinate for blue



Startmenue “control”,
coordinate for white



Explanation of startmenue “control” (2/8):

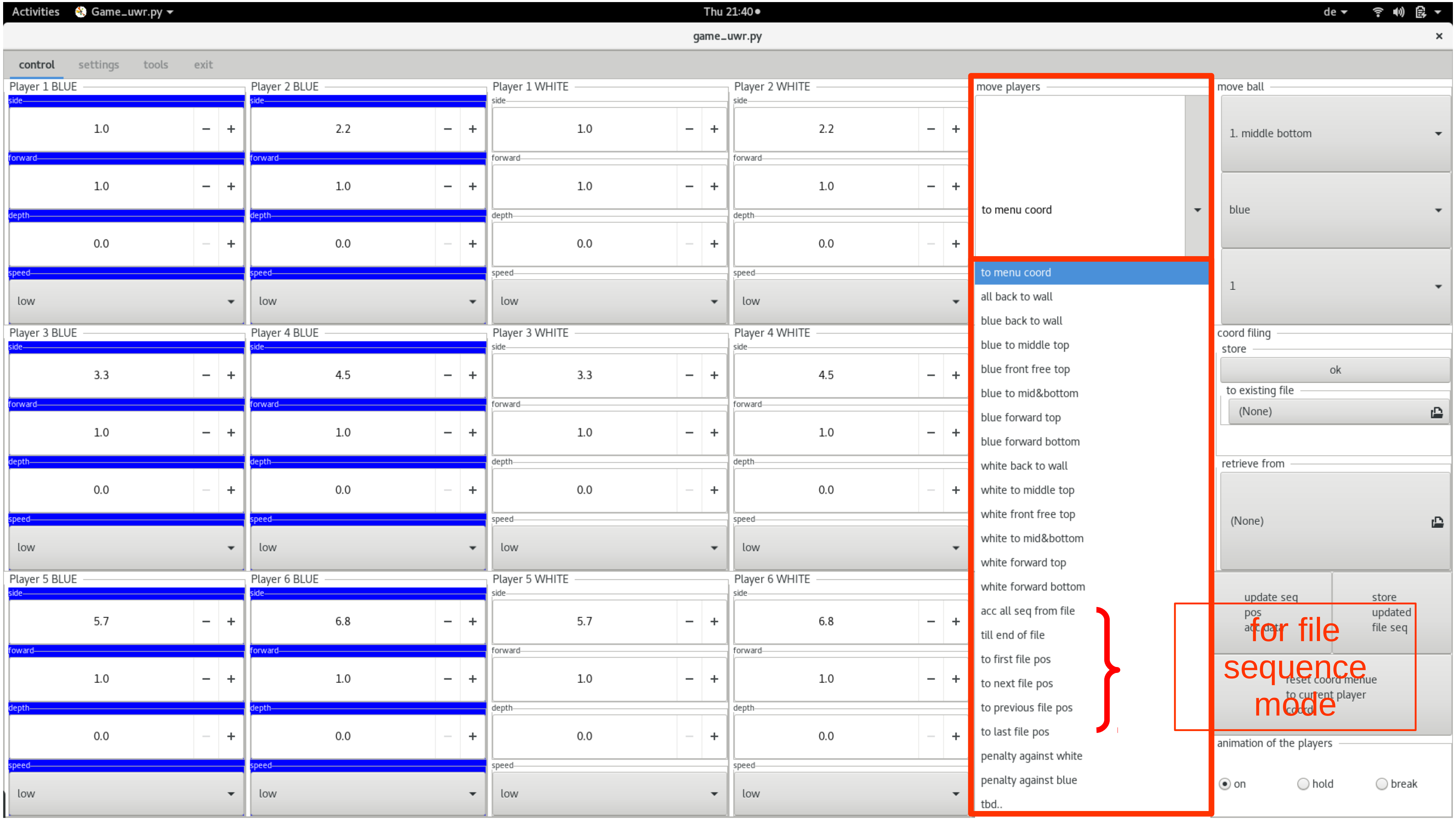
(2) Select several defined move of the players via an expanding menue (see screen copy at next page).

Data from (1) will be taken into acount only when “to menu coord” is selected. The move will become effective only by clicking on (6).

Example: choose “penalty against white” in (2), click (6), then the players will move according this “penalty against white” positioning: see picture “Start position of a penalty” later in this manual.

The positioning of the ball is controlled independantly from the players positioning and is defined in (3), except a moving according a file sequence is choosen (in that case, the ball coordinates of the csv file are automatically overtaken).

Entry “move players” (2)



for file sequence mode

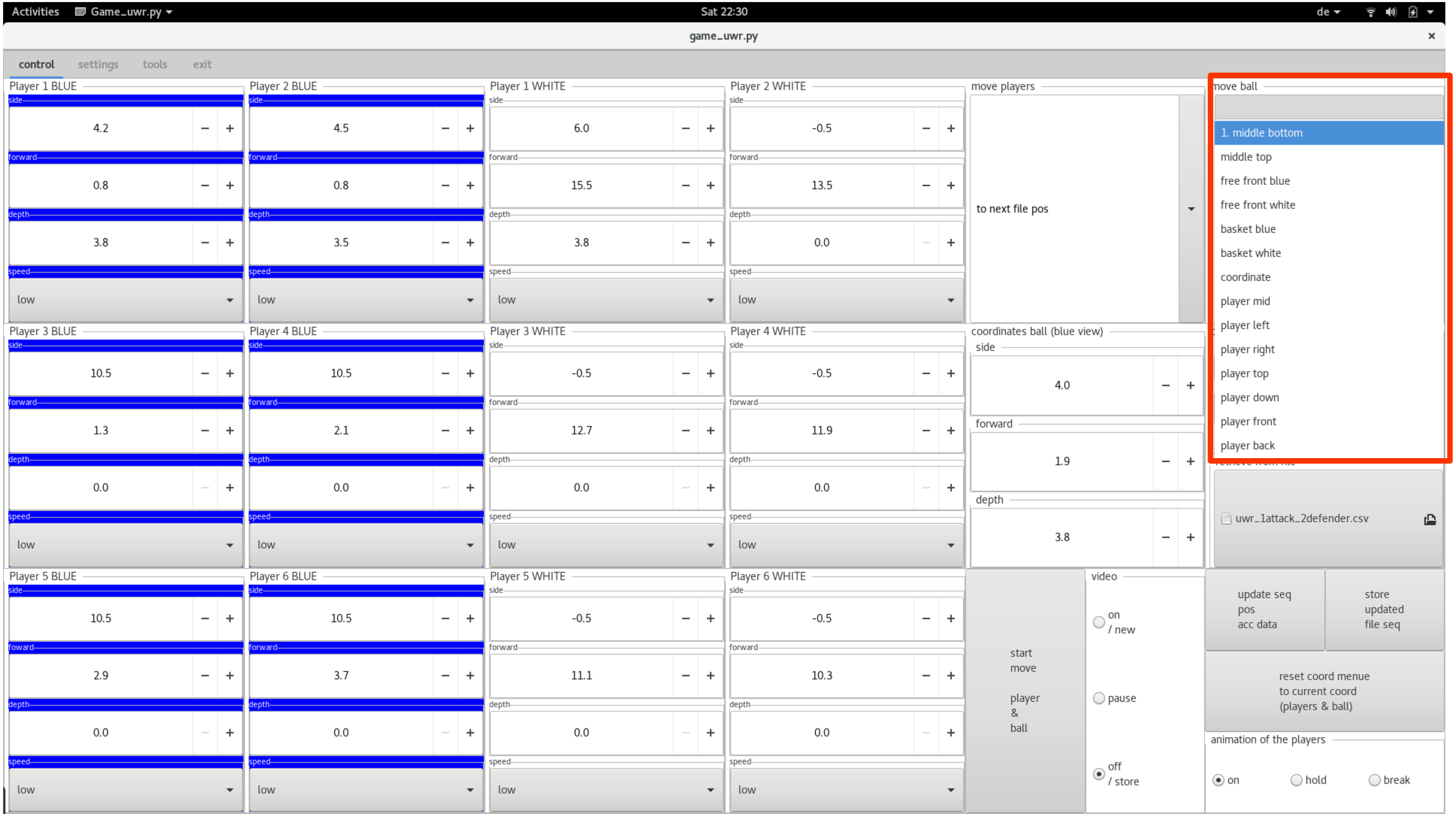
Explanation of startmenue “control” (3/8):

(3) select where the ball has to be positioned to. 13 different positions can be choosen.

If an action of type “player” is selectioned, then the fields underneath this entry (3), “blue/white” and “1...6” in expanding menue, will be readen in or to move the ball to the desired position.

After clicking (6), the move of the ball will start.

Entry “move ball” (3)



Activities Game_uwr.py ▾ Sat 22:30 de ▾

game_uwr.py ×

control settings tools exit

Player 1 BLUE
side 4.2 − +
forward 0.8 − +
depth 3.8 − +
speed low ▾

Player 2 BLUE
side 4.5 − +
forward 0.8 − +
depth 3.5 − +
speed low ▾

Player 1 WHITE
side 6.0 − +
forward 15.5 − +
depth 3.8 − +
speed low ▾

Player 2 WHITE
side -0.5 − +
forward 13.5 − +
depth 0.0 − +
speed low ▾

move players
to next file pos ▾

move ball
1. middle bottom
middle top
free front blue
free front white
basket blue
basket white
coordinate
player mid
player left
player right
player top
player down
player front
player back

Player 3 BLUE
side 10.5 − +
forward 1.3 − +
depth 0.0 − +
speed low ▾

Player 4 BLUE
side 10.5 − +
forward 2.1 − +
depth 0.0 − +
speed low ▾

Player 3 WHITE
side -0.5 − +
forward 12.7 − +
depth 0.0 − +
speed low ▾

Player 4 WHITE
side -0.5 − +
forward 11.9 − +
depth 0.0 − +
speed low ▾

coordinates ball (blue view)
side 4.0 − +
forward 1.9 − +
depth 3.8 − +

uwr_1attack_2defender.csv

Player 5 BLUE
side 10.5 − +
foward 2.9 − +
depth 0.0 − +
speed low ▾

Player 6 BLUE
side 10.5 − +
forward 3.7 − +
depth 0.0 − +
speed low ▾

Player 5 WHITE
side -0.5 − +
forward 11.1 − +
depth 0.0 − +
speed low ▾

Player 6 WHITE
side -0.5 − +
forward 10.3 − +
depth 0.0 − +
speed low ▾

start
move

player
&
ball

video
○ on / new
○ pause
● off / store

update seq pos acc data

store updated file seq

reset coord menue to current coord (players & ball)

animation of the players
● on ○ hold ○ break

Explanation of startmenue “control” (4/8):

(4) Entries for new ball 3D coordinates.

The value is positive similar to the coordinates of the blue team: front positioning, positioning to the right and positioning below. In order to move the ball to the given position, click on (6) when at (3) “coordinate” is selected.

Explanation of startmenue “control” (5/8):

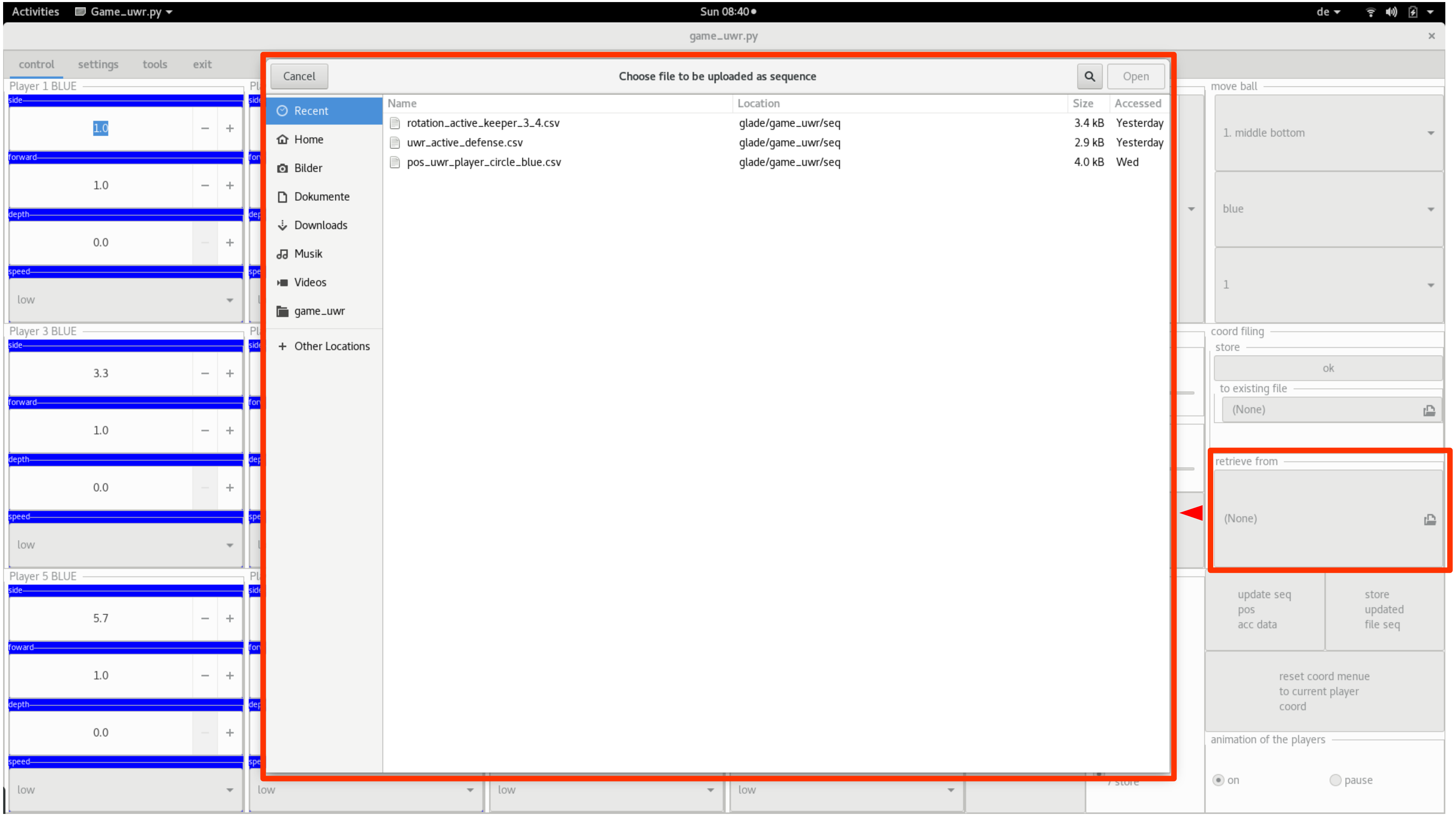
(5) menu for managing coordinates files, containing the 3D position of the ball and players.

“store ok” “to existing file” entry: store the current player and ball position into a file.
If the file was not choosen in “to existing file”, it will create a new file at each click. The name of the file is indicated in the page “settings”.
If the file was choosen in “to existing file”, it will add a position at the end of the file at each click (several clicks after each others possible; this will add a new sequence of 13 position lines - 2x 6 players and 1x ball - at each time).

It is possible to retrieve positions from a csv file and upload them into the application: see the screen copy at next page (entry “retrieve from”).

A continuous move according all positions of a csv file is possible by choosing in (2) “acc all seq from file” and then click on (6): if the video button of (7) is activated, it will generate a video. Scrolling of the position in the file is possible: for example “to first file pos” “to next file pos” “to last file pos” etc. see scrolling menu in (2). When this option “acc all seq from file” is choosen, the entries in (1) and (3) are ignored.
The option “till end of file”, if choosen, the entries in (1) and (3) are ignored and the all file sequence will run from the current file position on the screen till last file position.

Entry “retrieve from” (5)



Choose file to be uploaded as sequence

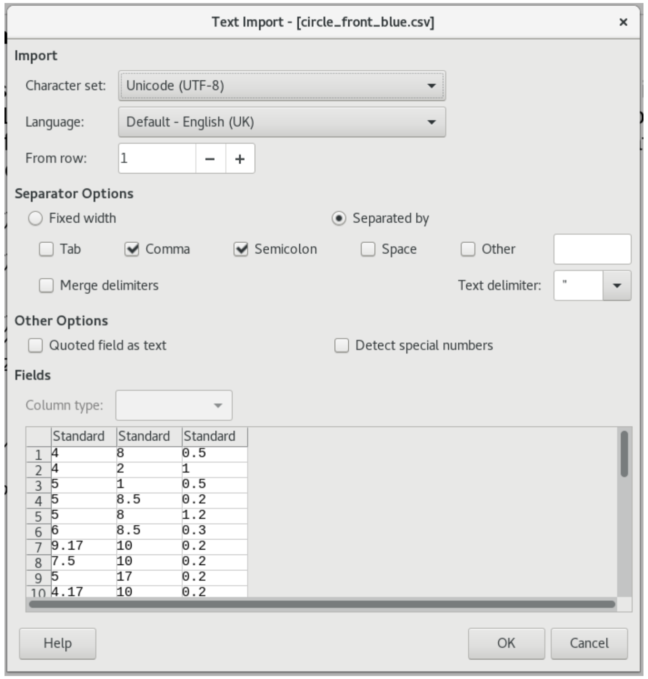
Name	Location	Size	Accessed
rotation_active_keeper_3_4.csv	glade/game_uwr/seq	3.4 kB	Yesterday
uwr_active_defense.csv	glade/game_uwr/seq	2.9 kB	Yesterday
pos_uwr_player_circle_blue.csv	glade/game_uwr/seq	4.0 kB	Wed

Explanation of startmenue “control”, entry (2) “.. file..” (6/8):

Format of the position file; it has the endig .csv and can be uploaded into Libroffice calc; this is a repetitive sequence of 13 lines of 3 numbers per line separated by a “,”. The line 1 to 6 are the 3D coordinates X Y Z of the player blue 1 to 6. The lines 7 to 12 are the 3D coordinates of the players white 1 to 6. The line 13 is the 3D coordinate of the ball. The players and the ball can be moved from each position to next/previous with the entry (2); file extract:

0.83,9,4 *(line 1 = player1 blue X Y Z)*
2.5,9,4 *(line 2 = player2 blue X Y Z)*
4.17,9,4 *(line 3 = player3 blue X Y Z)*
5.83,9,4 *(line 4 = player4 blue X Y Z)*
7.5,9,4 *(line 5 = player5 blue X Y Z)*
9.17,9,4 *(line 6 = player6 blue X Y Z)*
9.17,17,4 *(line 7 = player1 white X Y Z)*
7.5,17,4 *(line 8 = player2 white X Y Z)*
5.83,17,4 *(line 9 = player3 white X Y Z)*
4.17,17,4 *(line 10 = player4 white X Y Z)*
2.5,17,4 *(line 11 = player5 white X Y Z)*
0.83,17,4 *(line 12 = player6 white X Y Z)*
4.17,9.6,4 *(line 13 = ball XYZ)*
… *(next 13 lines = next positioning)*

1 position



Text Import - [circle_front_blue.csv]

Import

Character set: Unicode (UTF-8)

Language: Default - English (UK)

From row: 1

Separator Options

○ Fixed width ● Separated by

□ Tab ☑ Comma ☑ Semicolon □ Space □ Other

□ Merge delimiters Text delimiter: "

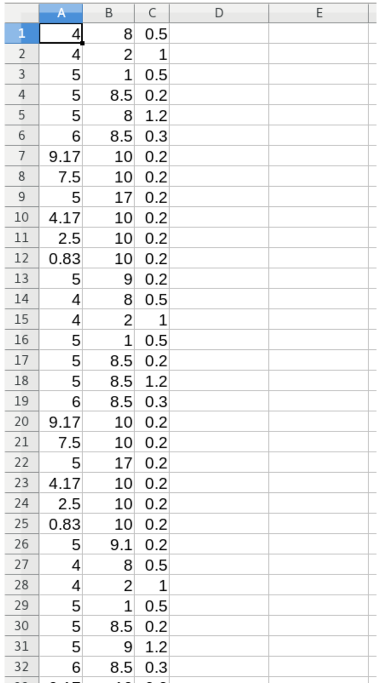
Other Options

□ Quoted field as text □ Detect special numbers

Fields

Column type:

	Standard	Standard	Standard
1	4	8	0.5
2	4	2	1
3	5	1	0.5
4	5	8.5	0.2
5	5	8	1.2
6	6	8.5	0.3
7	9.17	10	0.2
8	7.5	10	0.2
9	5	17	0.2
10	4.17	10	0.2



	A	B	C	D	E
1	4	8	0.5		
2	4	2	1		
3	5	1	0.5		
4	5	8.5	0.2		
5	5	8	1.2		
6	6	8.5	0.3		
7	9.17	10	0.2		
8	7.5	10	0.2		
9	5	17	0.2		
10	4.17	10	0.2		
11	2.5	10	0.2		
12	0.83	10	0.2		
13	5	9	0.2		
14	4	8	0.5		
15	4	2	1		
16	5	1	0.5		
17	5	8.5	0.2		
18	5	8.5	1.2		
19	6	8.5	0.3		
20	9.17	10	0.2		
21	7.5	10	0.2		
22	5	17	0.2		
23	4.17	10	0.2		
24	2.5	10	0.2		
25	0.83	10	0.2		
26	5	9.1	0.2		
27	4	8	0.5		
28	4	2	1		
29	5	1	0.5		
30	5	8.5	0.2		
31	5	9	1.2		
32	6	8.5	0.3		

Left: double click with the mouse on the .csv file make libroffice opening. Click “ok”.
Right: File sequence appear

Explanation of startmenue “control” (7/8):

(6) button for generating a move of the players and the ball according the choosen parameter in (2), for the players, and (3), for the ball.

(7) buttons for generating a video; the video generation will be activ when the animation is running/visible in the plot window.

(8) it allow to upload the coordinates of the players which can be seen in the game window, into the 3D coordinate menue (1). (1) dont change automatically according the moves defined by (2): it can be updated with this button.

(9) Time based movements (depending of speed); not implemented yet.

(10) When a sequence from a file was uploaded via (5) “retrieve from”, all positions are in the memory. When a position of that file was placed on the screen (via (2) and choosing for example “first file pos”) and if some changes were made on the coordinates of the players/ball, it is possible to overwrite the sequence in memory with the position on the screen. It helps to modify uploaded sequence files afterward (see 11).

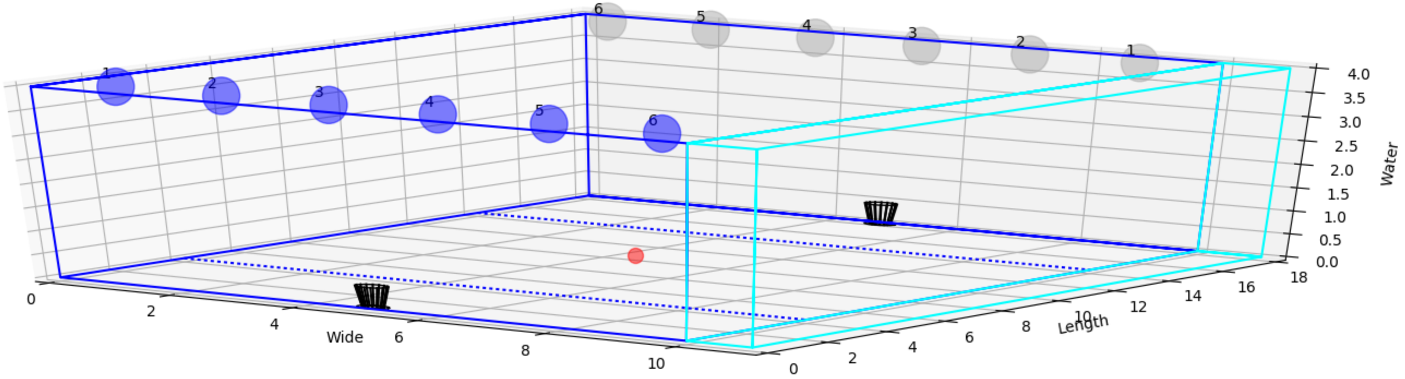
(11) The modified sequence in (10) can be stored again in a file which name is indicated in the menu “settings”, entry (3).

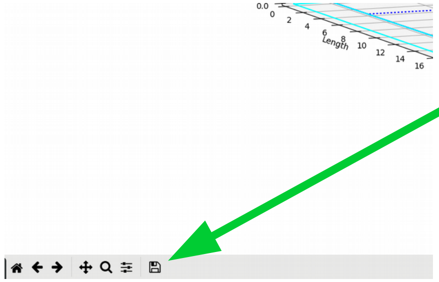
Explanation of startmenue “control” (8/8):

(12) When a file of movement sequence (xxx.csv) is uploaded in (5) “retrieve from” and the function (2) is running with “acc file seq”, after clicking on (6),

- it is possible to “hold” the animation by clicking in this area. This can be usefull in order to pause a video creation or have time to speak to others when an animation is running.
- If in (2) “acc file seq” is running, “break” will help to cancel the automatic running of the full sequence in the file (the animation will end at the next file position).
- Clicking on “on” will activate again if “hold” was activ. A “on” after a “break” will not do anything.

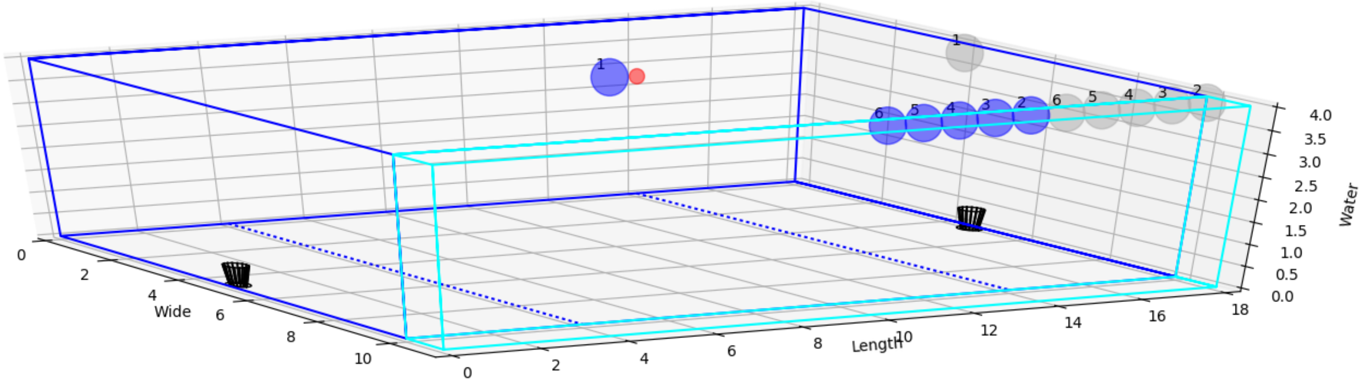
Start of a game; the application was just started.
Picture taken with screen copy or clicking the
store button of the matplotlib menue;
see below left



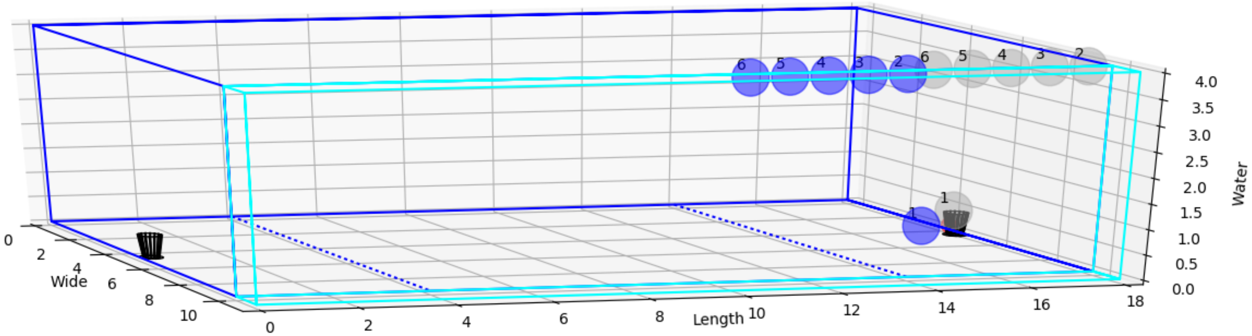


Matplotlib store button in the toolbar

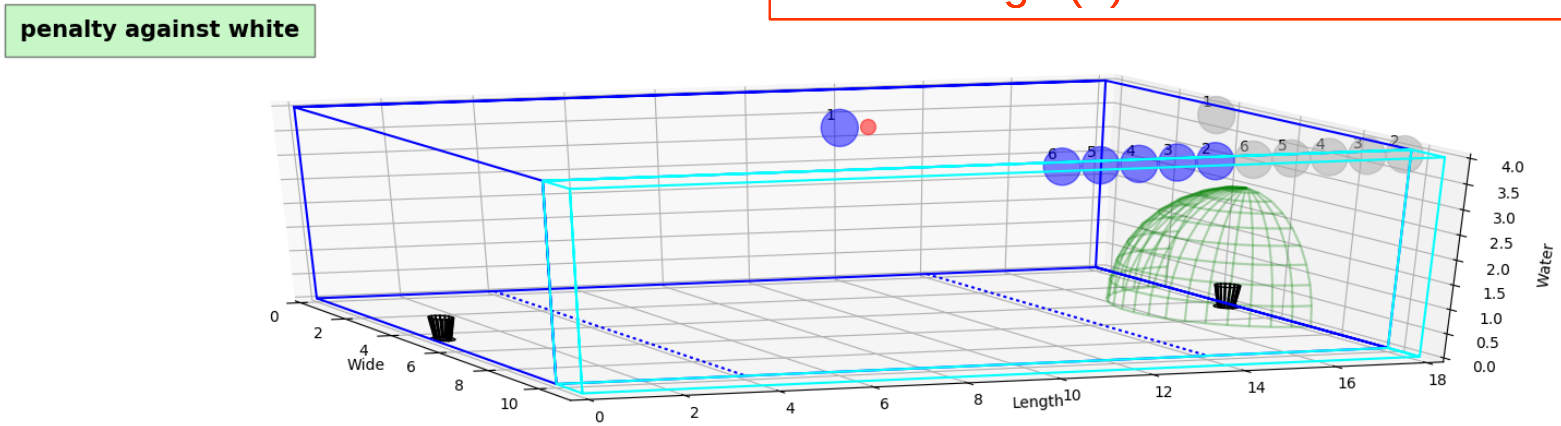
Start position of a penalty; for creation of that picture, put the players at the position “penalty against white” in (2), moving the ball (3) to “player front” “1” “blue”, click on (6), store the picture using the store button in the matplotlib window



Penalty in action; put the players at the position “penalty against white” in (2), move the ball to “player front” “1” “blue” in (3), click on (6) then (8), feed new coordinate entries of (1) for both players nb 1, choose “to menue coord” in (2), click on (6), store the picture by clicking the store button in the matplotlib toolbar

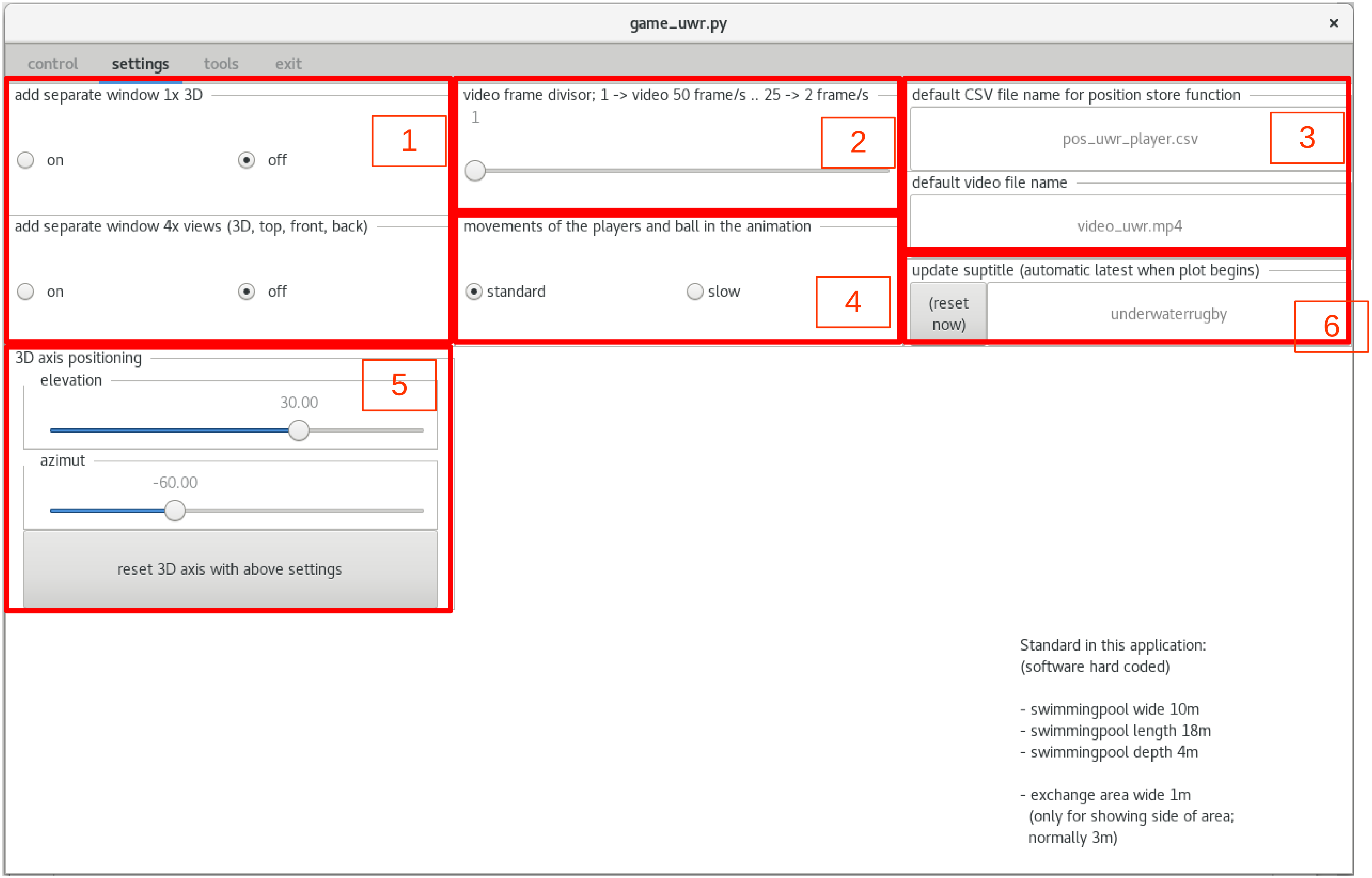


Area the goalkeeper should not leave during a penalty (except he has control of the ball) is in green; see menue “tools” for activation / deactivation.
The suptitel content (green box) can be updated with settings (3)



penalty against white

Menue “settings



add separate window 1x 3D: ○ on ● off

add separate window 4x views (3D, top, front, back): ○ on ● off

video frame divisor; 1 -> video 50 frame/s .. 25 -> 2 frame/s: 1

movements of the players and ball in the animation: ● standard ○ slow

default CSV file name for position store function: pos_uwr_player.csv

default video file name: video_uwr.mp4

update suptitle (automatic latest when plot begins): underwaterrugby

3D axis positioning

elevation: 30.00

azimut: -60.00

Standard in this application:
(software hard coded)

- swimmingpool wide 10m
- swimmingpool length 18m
- swimmingpool depth 4m

- exchange area wide 1m
  (only for showing side of area;
  normally 3m)

Explanation of menue “settings”:

(1) additional windows can be opened in case a second desktop is available for presentation to other persons during a rugby discussion/seminar.
→ A 1x 3D window can be selected: this is the copy of the main window which is already open (measurement annotations and spheres for representation of free and penalty distance will not be showned there).
→ A 4x views windows will be opened/closed.

(2) scalling from 1 to 25: this allow to reduce the number of stored images, which will create the video. The effect will be the video will be much quicker during the movement of players (1 high number of frames making a slow motion video; 25 low number of frame making the video fast or jumping from one position to the other). A number of “2” is creating a video with a normal speed on a high spec PC.

(3) entries for giving names to files which can be created/stored: positioning csv file and video file mp4.

(4) when the animation of the player is running at the PC screen, a slow motion can be choosen. This has no effect on the video if the recording is running; for this, look at (2).

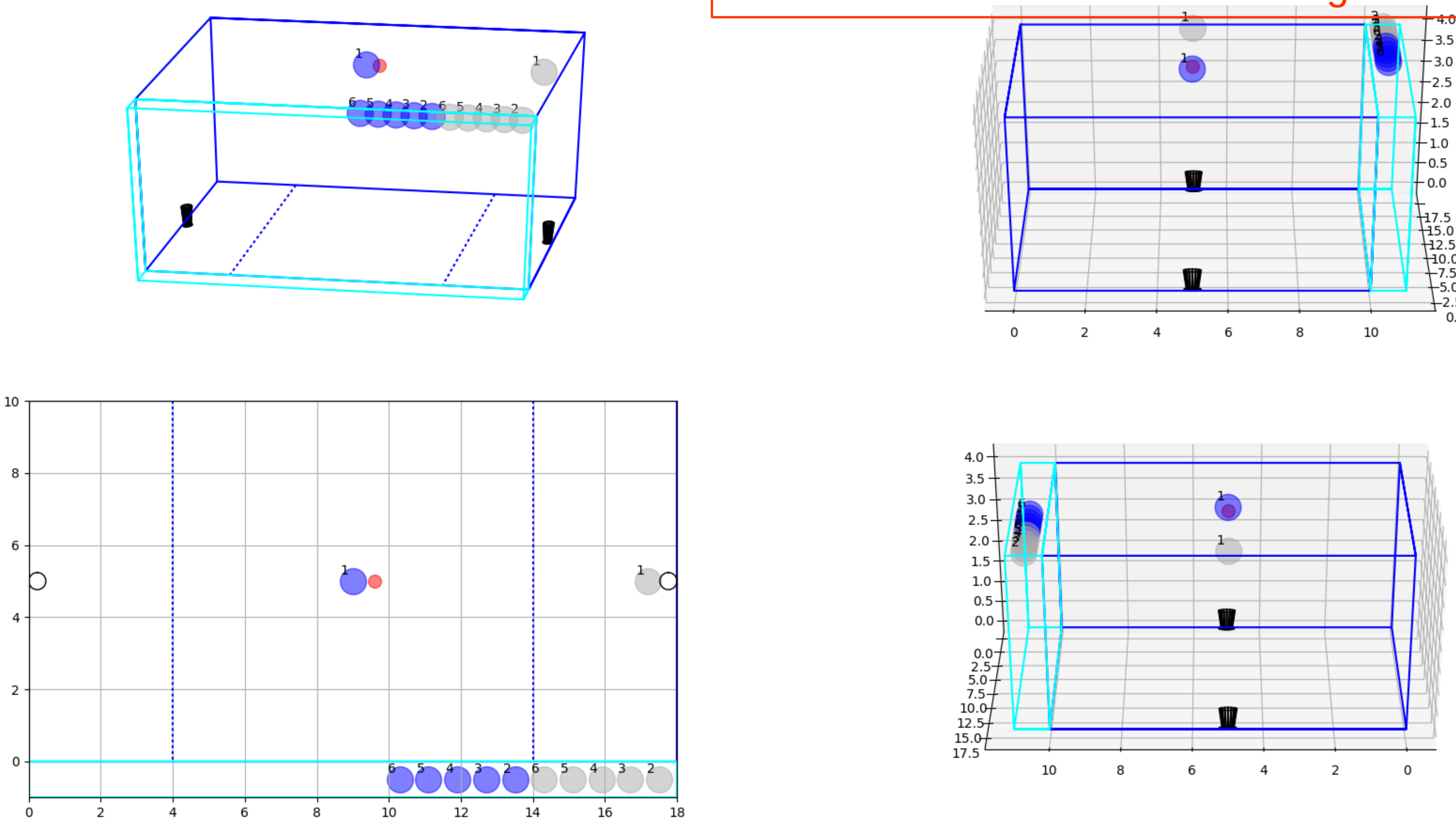
Explanation of menue “settings”:

(5) allow a move of the 3D axes according the given values.

Anyway, the 3D axes can be moved with the mouse by holding the left button and moving the cursor over the window = this is independent of these values in the 3D view.

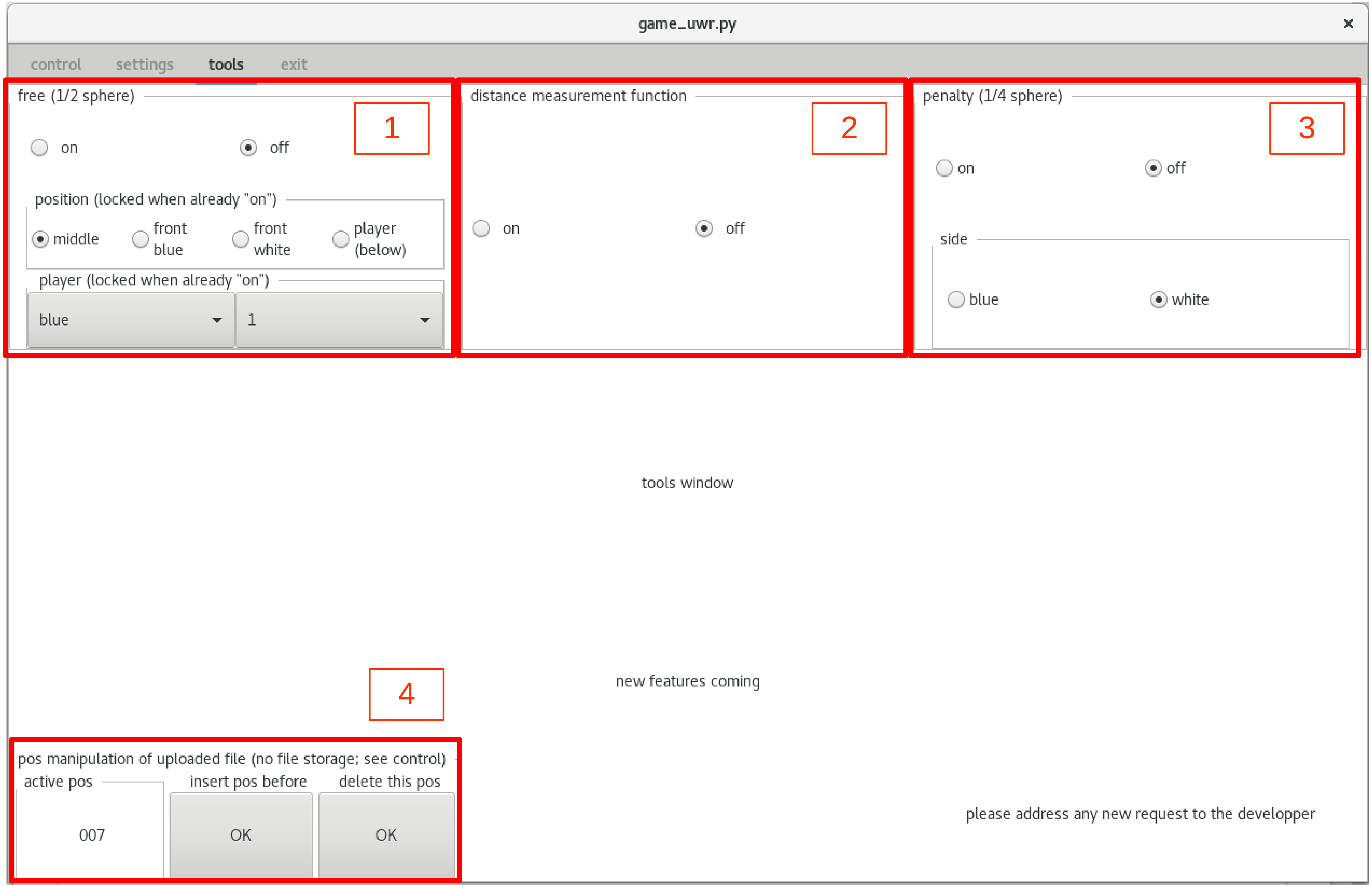
It helps to create video and pictures at the same repetitive identified positions.

(6) the plots have everytime a suptitle (default “underwaterrugby”) and it can be changed. This make possible the commenting of dynamic moving: if the video is “on”, this text/comment will be on the video.



Additional window with 4 views by clicking on (1) in the menue “settings”

Menue “tools”



game_uwr.py ×

control settings **tools** exit

free (1/2 sphere)

○ on ● off

position (locked when already "on")

● middle ○ front blue ○ front white ○ player (below)

player (locked when already "on")

blue ▾ 1 ▾

distance measurement function

○ on ● off

penalty (1/4 sphere)

○ on ● off

side

○ blue ● white

tools window

new features coming

pos manipulation of uploaded file (no file storage; see control)

active pos: 007

insert pos before: OK

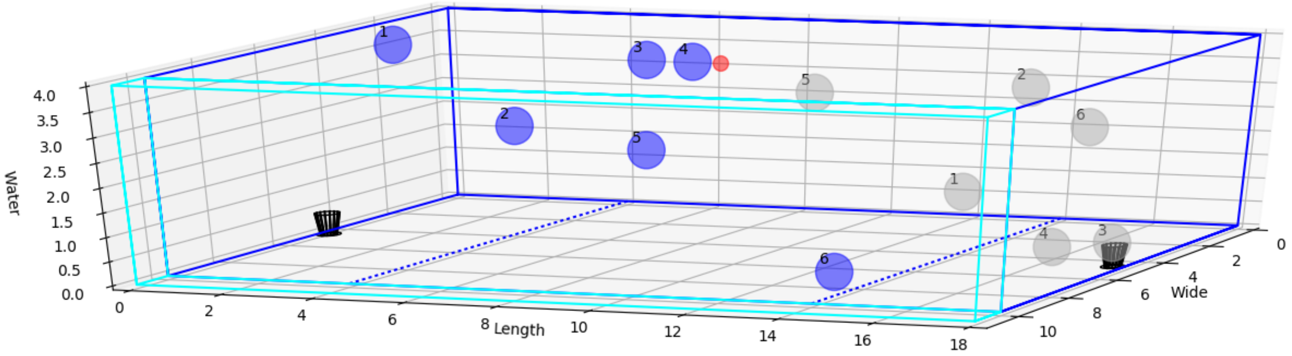
delete this pos: OK

please address any new request to the developper

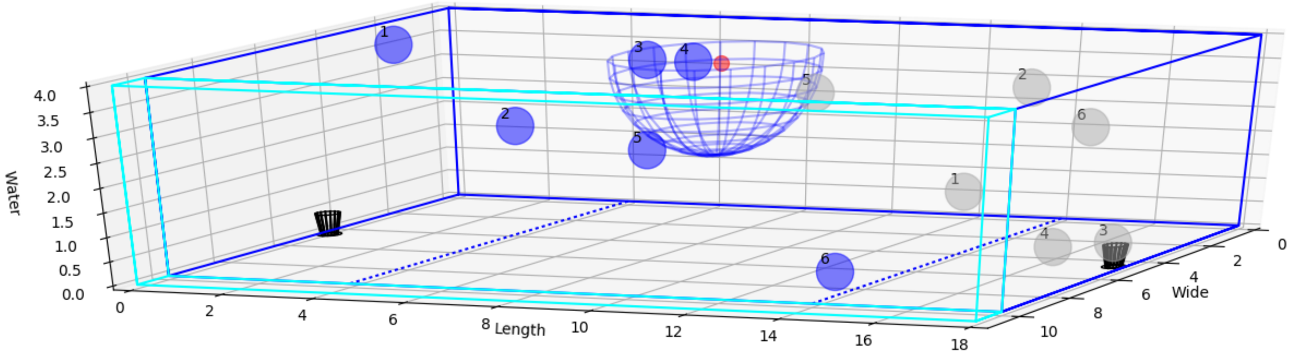
Explanation of menue “tools” (1/3)

(1) Activate a half of a sphere at the given position: this represent the 2m distance where any player of the other team should not act during the release of a free.

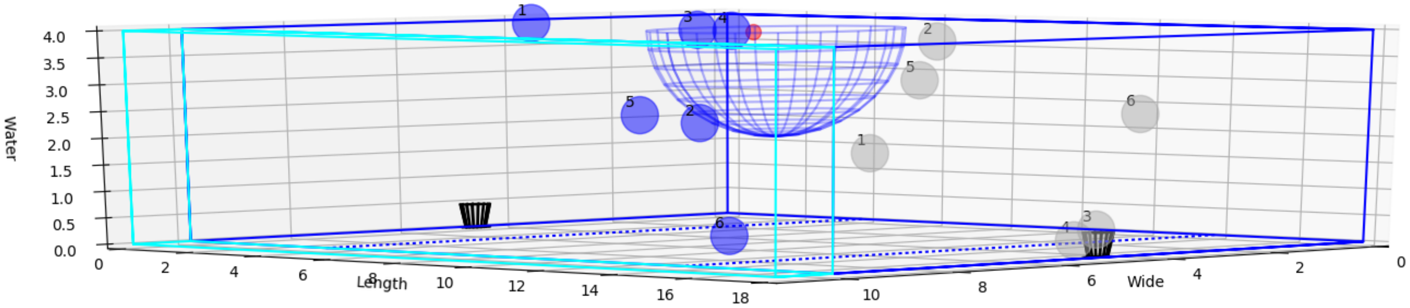
Free against white at startposition; for picture creation, put the players at the position “blue to middle top” in (2), move the ball to “player front” “4” “blue” in (3), click on (6) and then (8), feed new coordinate entries in (1) for all players, click on (6), click on the store button in the matplotlib toolbar



Free against white; 2m distance indicated: click on (1) in the menu “tools” with choosing the player blue 4

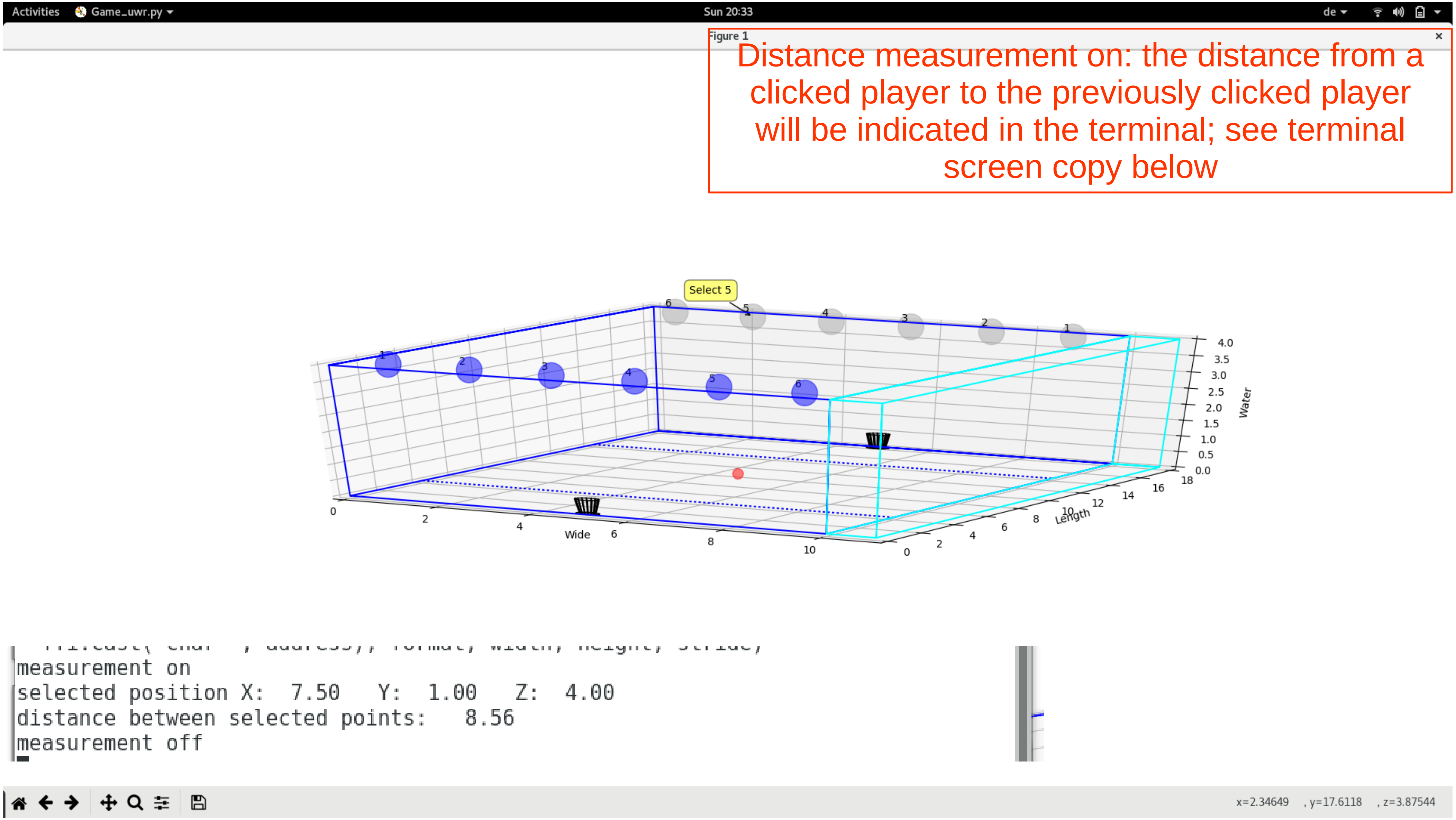


Free against white; nobody in the passive 2m area: it can be seen by turning the matplotlib 3D window with the mouse button left activated



Explanation of menue “tools” (2/3)

(2) a functionality for measuring the distance between 2 players can be activated there. Measurement when “on”: click first a player; click a second player; the distance appear in the command terminal where uwr_game.py was started, at the second click. Further clicking on player will give the distance to the previous player.



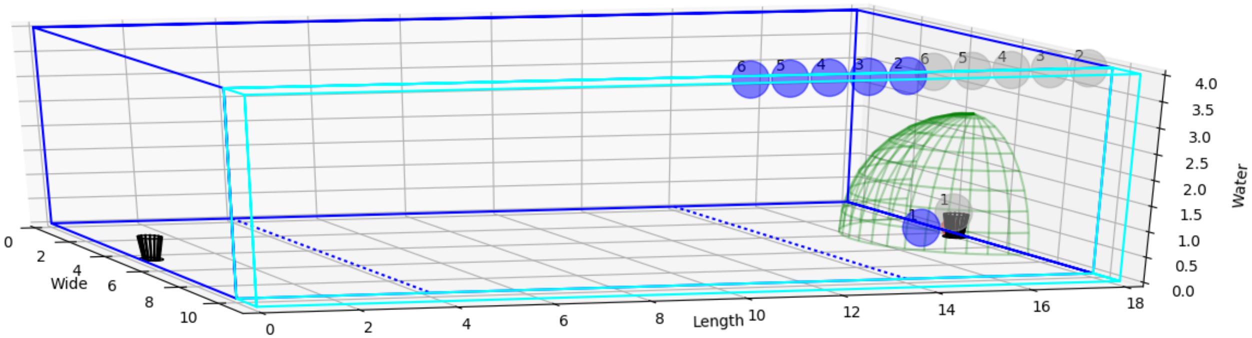
Distance measurement on: the distance from a clicked player to the previously clicked player will be indicated in the terminal; see terminal screen copy below

```
    ffi.cast( char  , address), format, width, height, stride)
measurement on
selected position X:  7.50   Y:  1.00   Z:  4.00
distance between selected points:   8.56
measurement off
```

Explanation of menue “tools” (3/3)

(3) a quarter of a sphere can be drawn at the given position: it represent the distance where the goalkeeper should stay as long he is not having full control of the ball during a penalty. He can goes up, outside of that sphere, to take a breath, but not do anything against the player and the ball outside that sphere.

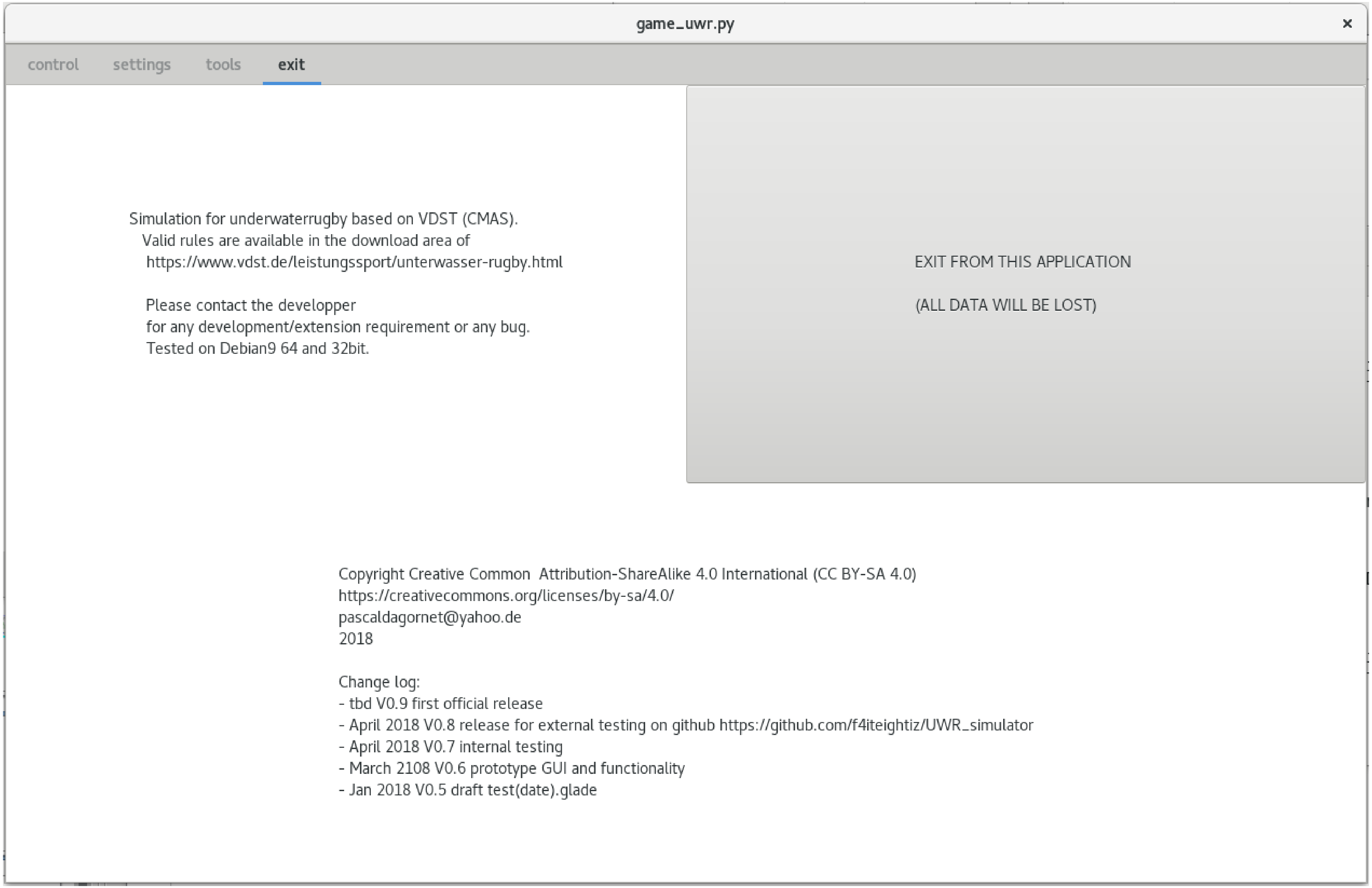
Area the goalkeeper should not leave during a penalty (except he has control of the ball)



Explanation of menue “tools” (3/3)

(4) when a file of sequences is uploaded, positions (uploaded in memory) can be deleted or added.
In menue “control”, click “11” in order to store that modified sequences into a file.

Menue “exit”



Simulation for underwaterrugby based on VDST (CMAS).
Valid rules are available in the download area of
https://www.vdst.de/leistungssport/unterwasser-rugby.html

Please contact the developper
for any development/extension requirement or any bug.
Tested on Debian9 64 and 32bit.

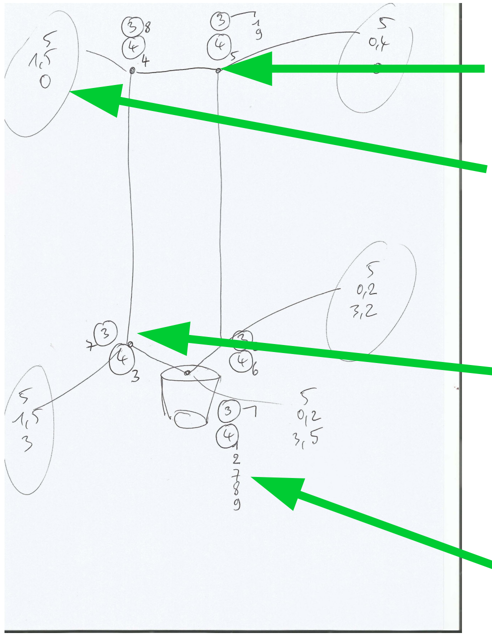
Copyright Creative Common Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)
https://creativecommons.org/licenses/by-sa/4.0/
pascaldagornet@yahoo.de
2018

Change log:
- tbd V0.9 first official release
- April 2018 V0.8 release for external testing on github https://github.com/f4iteightiz/UWR_simulator
- April 2018 V0.7 internal testing
- March 2108 V0.6 prototype GUI and functionality
- Jan 2018 V0.5 draft test(date).glade

HOWTO create a video (1/2):

1) create a csv file of several positions
2) upload that csv file and create the video according that position file

To 1)



<u>create manually a draft of the creation:</u>
a) position 3D: 5 main positions identified

b) the 3D coordinates of the 5 main positions

b) which player will be moved:
3 and 4

c) at which sequence which player will be seen:
Here, player 4 at the sequences 1,2,7,8,9

Move the players 3 and 4 according the coordinate of pos 1.
Click in “control” the “Coord filing” “store “ok” when the first position of the players 3 and 4 are on the screen. The file pos_uwr_player.csv will be created (only 1 sequence in it).
Click in “control” the “coord filing” “to existing file”, choose pos_uwr_player.csv : it allow you to append the future positions 2..9 into that file.

HOWTO create a video (2/2):

Move the players 3 and 4 according the coordinate of pos 2,3..9.
Click in “control” the “Coord filing” “store “ok” each time

To 2) In “coord filing”, choose the file pos_uwr_player.csv in “retrieve from”.
Choose the “move players” “all acc seq file”, click the “video” “on”.
Click the “start move”.
When the animation dont move anymore (= all sequences 1..9 were seen on the screen), click “video” “off”.
The video is created (mp4 file) in the directory dedicated to this (see the installation advice).

In order to create comments,

- in “coord filing”, choose the file pos_uwr_player.csv in “retrieve from”.
- Choose the “move players” “to first file pos”, click the “video” “on” and click on “go to”.
- In the “settings”, change the text of the “suptitle”, choose the “move players” “to next file pos”, and click on “go to”.
- In the “settings”, change the text of the “suptitle”, and click on “go to”.
- In the “settings”, change the text of the “suptitle”, and click on “go to” etc.
- At the end, click on “video” “off”: a video file was created with all the comments indicated in a small green box top left.

Next (not exhaustiv):

- Test in other linux versions, Microsoft Windows and make necessary improvements of the python script and/or README in order to make the application running out of the box in diverse environment
- create new file positioning sequences and video creation of it
- create additional pictures of typical game positioning
- share of generated pictures and video on youtube.com
- Improve/developp functionality: realtime behaviour (time based)
- Move GUI from Gtk3 to Qt5 for use on android tablets

DONT HESITATE TO CONTACT THE AUTHOR FOR RECEIVING VIDEOS AND PICTURES AND FILE SEQUENCES; FEW OF SEQUENCE FILES ARE ALREADY UPLOADED ON GITHUB https://github.com/f4iteightiz/UWR_simulator