Dart-O-Mat 3000

Jan 23, 2021

Contents

So you found your way here because you want to play steel dart but you don't want to bother counting, right? Then you are very welcome to try out **Dart-O-Mat 3000**!

CHAPTER 1

About the project

As I started playing Darts I always used to play on electronic Dart boards. I started to wonder how it will be to play steel dart, so I bought a steel dart board and some steel darts and played. Soon I began to realise that scoring on paper or with an mobile app might work but is not as **sexy** as I hoped for.

That was the moment this project was born. There aren't much projects like this online ready to use. Either they are closed source or they do not work like I wanted them to do.

So with this motivation I started coding this project.

1.1 The idea

I was thinking of a scoreboard which has to handle score mechanisms on different games. It has to work with multiple players and has to look some way of **sexy**.

1.2 The structure

The scoreboard is supposed to be shown on a display in front of the player. At the best the display is standing right on top of the dart board compartment or maybe a little to the left or right of the dart board. You will use any smartphone or tablet as a game controller to control the game setup.

1.3 Recognition anyone?

As there were already a few projects online using webcams and openCV to recognise Darts beeing thrown to a dart board and to get the number and modifier hit I was curious if this will be working with my scoreboard project maybe. So I designed the scoreboard to have a RESTful API which the recognition software can use to insert throws automatically. This way the scoreboard will work fully automatically giving the players the feeling they are playing with a electronic dart board, BUT with a regular steel dart board.

1.4 No recognition, no problem!

If you do not want to use a “complicated” recognition setup I didn’t want to let players down. So I designed the scoreboard to handle this case as well. In this case buttons will be displayed within the gameController module. With this buttons you can use the scoreboard like any other mobile app you can find for dart scoring.

1.5 A few words on the coding language

As I am using python a lot at work it came naturally to me to use python here as well. Also I think it will address a broad community if someone wants to contribute. For the webservice I am using python flask with sqlalchemy to store game data. For the live updates with the scoreboard python socketio is used. Together it will build a nice and smooth experience while playing.

1.6 Ready to start?

So you feel like you want to test it out yourself? Then go ahead and check out how information on *recognition software* and how to *install* everything.

CHAPTER 2

Installation

2.1 Git

It is recommended to clone the project with git. So you will need to install git for your specific operating system. Afterwards you can clone the project like so:

```
git clone https://github.com/patrickhener/dart-o-mat-3000.git
```

2.2 Python

As the first requirements you will need *Python3* and *Python3-Pip*. You might wanna refer to the official documentation on how to install and for your specific operating system.

2.3 Virtual Environment

It is recommended to use a virtual environment to keep your pip clean. If you do not want to use one just continue with the next section.

To install and activate the virtual environment do the following:

```
pip install --user virtualenv
python -m venv ./venv
. venv/bin/activate
```

2.4 Install python package requirements

To install the requirements you can use *pip* again like shown in the code block below. Keep in mind that you can do this either within a virtual environment or not.

```
pip install --user -r requirements.txt
```

You should no have everything in place to *configure* and *run* the scoreboard.

CHAPTER 3

Recognition

3.1 Recommendation on recognition software

As this project is developed depending on the of Niels Lüdemann it is recommended to use it.

3.2 RESTful API Requests

You can use any other recoginition as well, as long as it is propagating it's throws and next player via a GET Request to the scoreboard engine in this format:

1. URL to insert a throw /game/throw/*number-of-throw*/*modifier-of-throw*

- Example: Single 20 will be /game/throw/20/1
- Example: Miss will be /game/throw/0/1
- Example: Double 18 will be /game/throw/18/2

2. URL to switch to next player /game/nextPlayer

For a full overview of possible API calls refer to *API*

CHAPTER 4

Configuration

I decided to use a central config file on all the configurations a user is supposed to do. The file is called *config.py* and can be found in the root directory of the project.

4.1 Configuration Parameters

The following configuration parameters can be found in the configuration file. The table gives an overview for what they are used:

Parameter	Function
IPADDR	QRCode IP address for index page
IFACE	Server Interface IP for the webserver
PORT	Server Interface Port for the webserver
SSL	Server Interface is https (SSL) or plain
RECOGNITION	Use recognition software or not. GameController will look different depending on this flag
SOUND	Control wether to play sound or not
BA-BEL_DEFAULT_LOCALE	Controls the language to be displayed. For now either *de* or *en*.
BA-BEL_DEFAULT_TIMEZONE	Timezone to use. Isn't used yet within the code, though.
DEBUG	Python Flask Debugging mode
TESTING	Python Flask Testing mode

The other parameters are not supposed to be changed.

Edit the configuration to your liking and proceed to *running* the scoreboard.

CHAPTER 5

Run and Deploy

There are two ways to run the scoreboard. Running it for development or running it in a productive environment. The follwing sections will describe both as well as how to deploy it to run automatically at the systems start up.

5.1 For Development

For development you can simply run it like the following code block shows. Remember to activate the virtual environment first, if you happen to use one: . *venv/bin/activate*

```
./run.py
```

You will now have a rich output of what is happening whilst you play. The development server will restart everytime you change code and save it. It will directly crash and give you stack traces if you have an error. This is very convenient for developing.

Navigate your browser to to see if it is working.

5.2 Deploying for production

The following instructions are mainly taken off . Feel free to follow along those instructions instead or follow the steps below.

5.2.1 Gunicorn

You can use gunicorn do deploy the app in a production environment. It is highly recommended to follow this steps.

Create virtual environment

If you have not already done this now is the time to. Navigate to the root folder and do the following:

```
python -m venv ./.venv
```

Activate the virtual environment afterwards and install requirements.

```
source .venv/bin/activate
pip install -r requirements
```

Test gunicorn

Test if everything is working out.

```
gunicorn --bind 0.0.0.0:5000 --worker-class eventlet -w 1 run:app --reload
```

The output should show something like:

```
[2019-04-25 13:25:26 +0200] [1605] [INFO] Starting gunicorn 19.9.0
[2019-04-25 13:25:26 +0200] [1605] [INFO] Listening at: http://0.0.0.0:5000 (1605)
[2019-04-25 13:25:26 +0200] [1605] [INFO] Using worker: eventlet
[2019-04-25 13:25:26 +0200] [1608] [INFO] Booting worker with pid: 1608
```

You can exit out pressing *CTRL+C* and deactivate the virtual environment by issueing the command *deactivate*.

5.2.2 Systemd service for autostart

We will use a systemd service to automatically start the gunicorn server for us.

As root create a service file called **dom3000.service** under **/etc/systemd/system/**. It's content should look like this. Remember to alter the paths to fit your path structure:

Systemd service file

```
[Unit]
Description=Gunicorn instance to serve Dart-O-Mat 3000
After=network.target

[Service]
User=patrick
WorkingDirectory=/home/patrick/dart-o-mat-3000
Environment="PATH=/home/patrick/dart-o-mat-3000/.venv/bin"
ExecStart=/home/patrick/dart-o-mat-3000/.venv/bin/gunicorn --workers 1 --worker-class␣
↪eventlet --bind 0.0.0.0:5000 run:app --reload

[Install]
WantedBy=multi-user.target
```

Choosing 0.0.0.0:5000 as a bind address will bind the server to all interface IPs on port 5000. Afterwards you can test if it starts. And if it does you can enable it to start automatically at boot time.

Enable service

```
sudo systemctl start dom3000.service
sudo systemctl status dom3000.service
sudo systemctl enable dom3000.service
```

Finally to test if the service is running fine just go and point your browser to directly on the Dart-O-Mat 3000 machine or to it's corresponding interface IP like for example

5.2.3 Nginx configuration

We will use nginx as a proxy to handle connections to our gunicorn service. If you have not yet installed nginx is the time to.

Nginx config file

Next we will create a config file called **dom3000** under **/etc/nginx/sites-available/**

It will have this content

```
server {
    listen 80;
    server_name localhost;

    location /socket.io/ {
        proxy_http_version 1.1;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        proxy_pass http://localhost:5000/socket.io/;
    }


    location / {
        include proxy_params;
        proxy_pass http://localhost:5000/;
    }
}
```

Enable site

Now we enable the site by linking to this file

```
sudo ln -s /etc/nginx/sites-available/dom3000 /etc/nginx/sites-enabled
sudo systemctl restart nginx
```

Test if it is working

You now should be able to get your **Dart-O-Mat 3000** at either directly on the machine or at it's corresponding IP address like for example

5.2.4 Setup config.py for production

You should be sure to choose the correct configuration flags in config.py in the root directory of **Dart-O-Mat 3000**. Those are

```
IPADDR = '192.168.1.10'  # QRCode IP
DEBUG = False
TESTING = False
```

As we are not testing or developing in this scenario be sure to choose False for both of this flags. Also you need to choose the right IP address so the QR code will be drawn accordingly.

Be sure to restart the **dom3000.service** afterwards issueing the command *sudo systemctl restart dom3000.service*

5.2.5 Autostart browser and index page

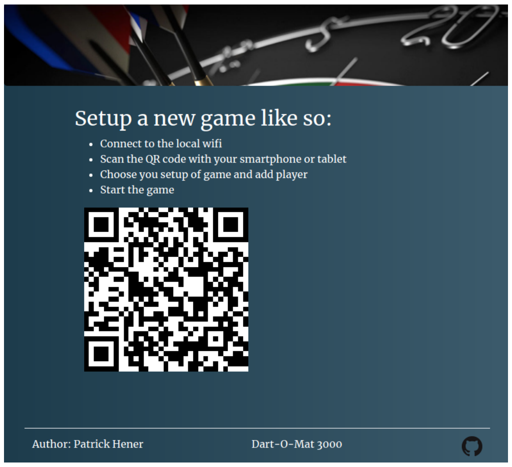
CHAPTER 6

Pages

This chapter will describe every single page involved in the scoreboard and it's function. For notation I will choose to just give you the subpath without the complete *http://hostip:port*.

6.1 Index

The index page can be found at path */game/*.


Setup a new game like so:

- Connect to the local wifi
- Scan the QR code with your smartphone or tablet
- Choose you setup of game and add player
- Start the game

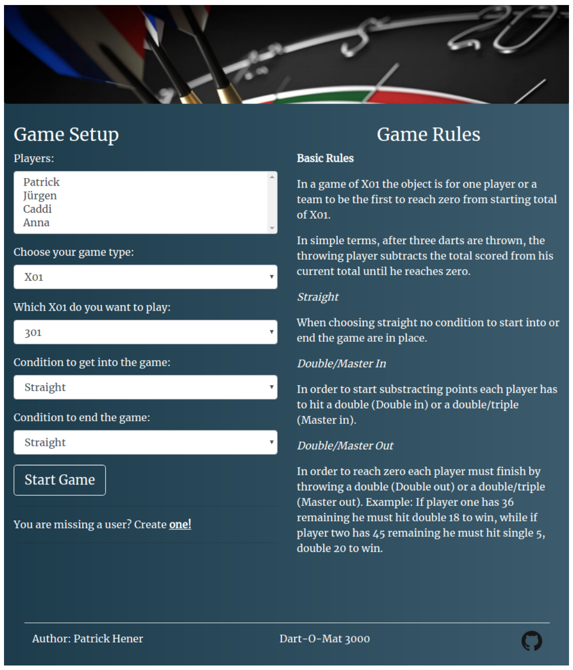
Author: Patrick Hener Dart-O-Mat 3000

The user is supposed to scan the generated QR code to redirect his smartphone to the *Admin* page.

6.2 Admin

The admin page can be found at path */game/admin*.

The admin page will give you all the possibilities to setup a game.



Game Setup

Players:

Patrick
Jürgen
Caddi
Anna

Choose your game type: X01

Which X01 do you want to play: 301

Condition to get into the game: Straight

Condition to end the game: Straight

Start Game

You are missing a user? Create **one!**

Game Rules

Basic Rules

In a game of X01 the object is for one player or a team to be the first to reach zero from starting total of X01.

In simple terms, after three darts are thrown, the throwing player subtracts the total scored from his current total until he reaches zero.

Straight

When choosing straight no condition to start into or end the game are in place.

Double/Master In

In order to start substracting points each player has to hit a double (Double in) or a double/triple (Master in).

Double/Master Out

In order to reach zero each player must finish by throwing a double (Double out) or a double/triple (Master out). Example: If player one has 36 remaining he must hit double 18 to win, while if player two has 45 remaining he must hit single 5, double 20 to win.

Author: Patrick Hener   Dart-O-Mat 3000

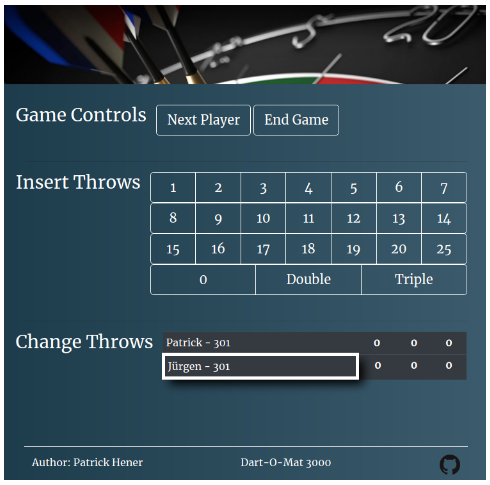
You first select all player you want to create a game with. If you have not already created player the link on the bottom of the page (*linktext: one!*) will redirect you to page *Create User*. Then you can choose which game to play and the game specific settings. After pressing *Start Game* the admin page will be redirected to the *game controller* and the *index page* will be redirected to the corresponding *scoreboard*.

Go *here* if you want to get a good overview of the games available and the rules.

6.3 GameController

The game controller page can be found at path */game/gameController*.

Within this page the user is able to influence several things within a running game. In general the controller page will look slightly different depending on which game you play. Also it will not show certain areas when playing with an active darts recognition software in comparison to playing without one.



Game Controls   Next Player   End Game

Insert Throws

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	25
0		Double		Triple		

Change Throws

Patrick - 301	0	0	0
Jürgen - 301	0	0	0

Author: Patrick Hener   Dart-O-Mat 3000

As an example you see two players playing 301. The page consists of different sections.

6.3.1 Game Controls

With the game controls you can either switch to the next player by pressing the button *Next Player* or you can end the game by pressing the button *End Game*. Also when a game is over the next player button will turn into a Rematch button which will start over the game with the exact same setup, when pressing the button.

6.3.2 Insert Throws

This section will only be shown when no recognition software is used (i.e. if the RECOGNITION flag in the *configuration* is set to False). Here you can insert throws just like in any mobile app for counting darts. If you wanna enter a double or a triple value be sure to hit double or triple first.

Caution: When hitting double or triple there is no way of *unhitting* it for now.

Tip: When three throws were inserted the controller is changing to the next player automatically.

The throw will then be inserted to the current players throws and will be handled by the game.

6.3.3 Update Throws

This section will give you the possibilities to update throws which were inserted wrong either by hand or by the recognition software.

Danger: Only throws within the last round of the player can be changed. So be sure to change wrong throws right away!

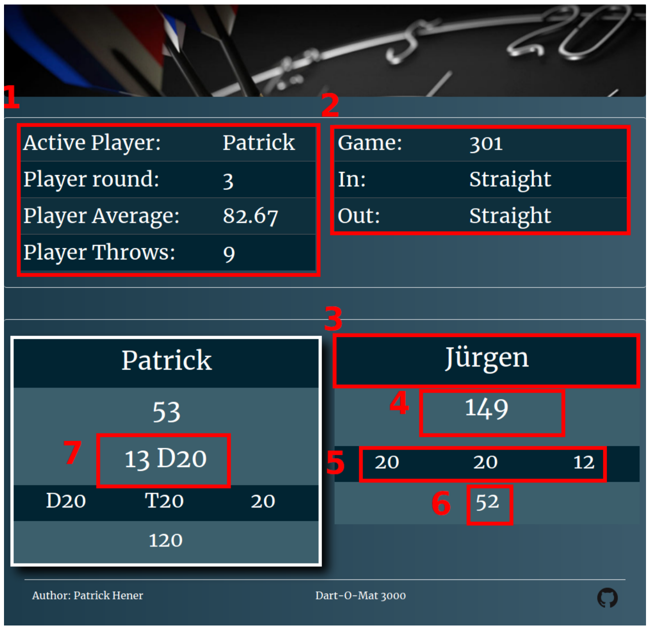
To change a throw just hit the throw which is wrong and a menu will appear. It looks exactly like the *Insert Throws* interface and works alike.

6.4 Scoreboards

Scoreboards will always highlight the active player with a white frame.

6.4.1 X01

This Scoreboard will show the progress of a X01 game. The X01 Scoreboard page can be found at path */game/scoreboardX01*.



Active Player:	Patrick
Player round:	3
Player Average:	82.67
Player Throws:	9

Game:	301
In:	Straight
Out:	Straight

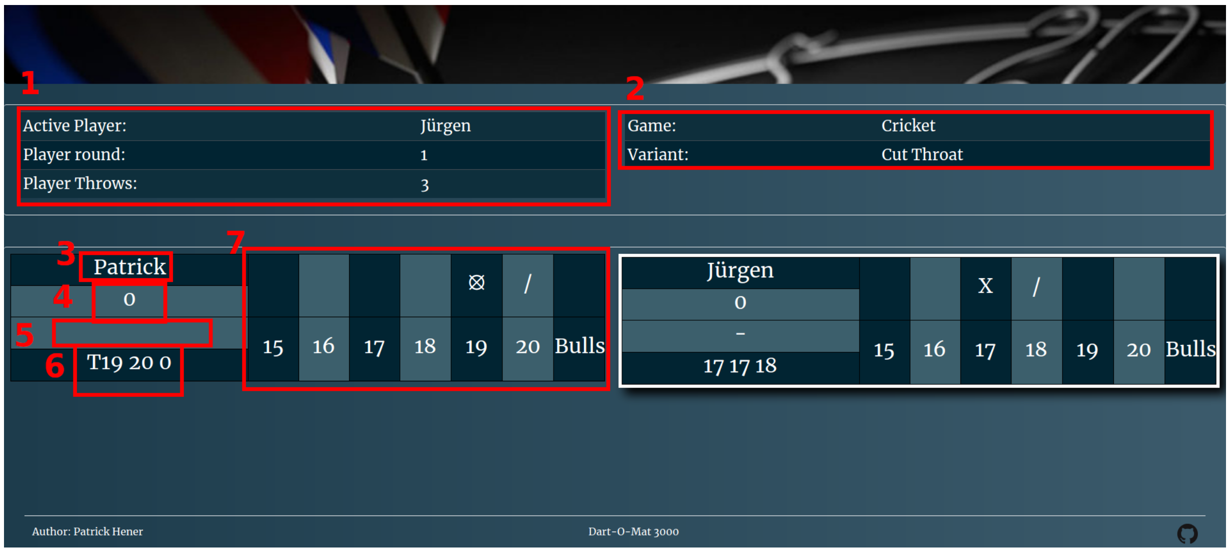
Patrick
53
13 D20
D20 T20 20
120

Jürgen
149
20 20 12
52

1. This area will show:
   - The current active player name
   - How many rounds the player has already played
   - The throw average of the rounds
   - The throwcount of the player
2. This area will show:
   - The game which is player
   - Game specific variants chosen
3. Individual player name
4. Player Score
5. Last three player throws
6. Sum of last three palyer throws
7. Message area used for checkout possibilities or other messages (Winner, Remove Darts, . . .)

6.4.2 Cricket

This scoreboard will show the progress of a Cricket game. The Cricket Scoreboard page can be found at path */game/scoreboardCricket*.



Active Player:	Jürgen
Player round:	1
Player Throws:	3

Game:	Cricket
Variant:	Cut Throat

Patrick							
0					⊗	/	
T19 20 0	15	16	17	18	19	20	Bulls

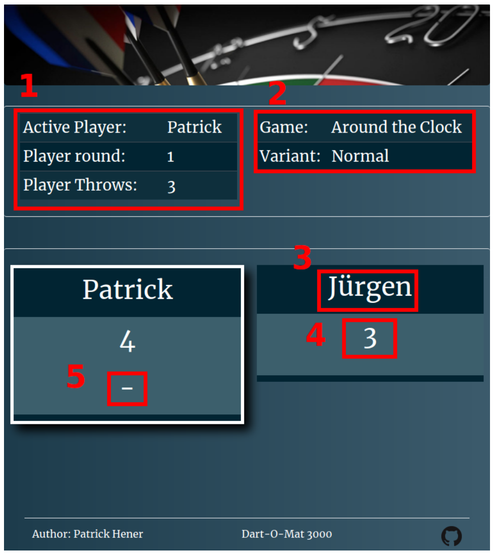
Jürgen							
0			X	/			
-							
17 17 18	15	16	17	18	19	20	Bulls

Author: Patrick Hener Dart-O-Mat 3000

1. This area will show:
   - The current active player name
   - How many rounds the player has already played
   - The throwcount of the player
2. This area will show:
   - The game which is player
   - Game specific variants chosen
3. Individual player name
4. Player Score
5. Message area used for messages (Winner, Remove Darts, . . .)
6. Last three player throws
7. Cricket table (Numbers will vanish when closed)
   - / means got hit once
   - X means got hit twice
   -

6.4.3 Around the clock

This scoreboard will show the progress of a Around the clock game. The Around the clock Scoreboard page can be found at path */game/scoreboardATC*.



Active Player:	Patrick
Player round:	1
Player Throws:	3

Game:	Around the Clock
Variant:	Normal

Patrick
4
-

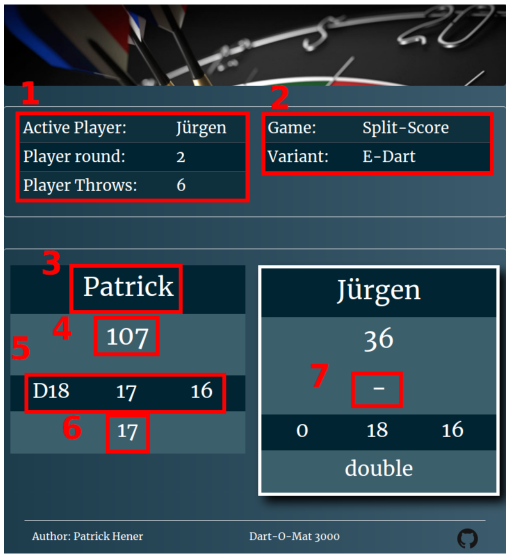
Jürgen
3

Author: Patrick Hener
Dart-O-Mat 3000

1. This area will show:
   - The current active player name
   - How many rounds the player has already played
   - The throwcount of the player
2. This area will show:
   - The game which is player
   - Game specific variants chosen
3. Individual player name
4. The next number supposed to be hit
5. Message area used for messages (Winner, Remove Darts, . . .)

6.4.4 Split-Score

This scoreboard will show the progress of a Split-Score game. The Split-Score Scoreboard page can be found at path */game/scoreboardSplit*.



Active Player:	Jürgen
Player round:	2
Player Throws:	6

Game:	Split-Score
Variant:	E-Dart

Patrick
107
D18 17 16
17

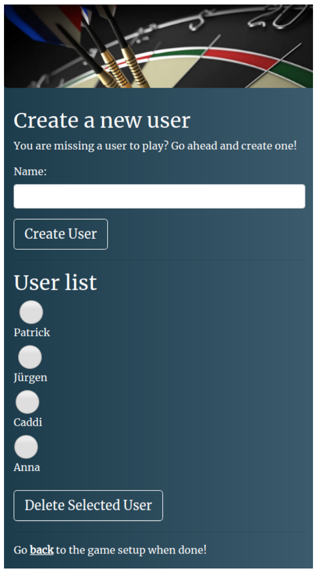
Jürgen
36
–
0 18 16
double

1. This area will show:
   - The current active player name
   - How many rounds the player has already played
   - The throwcount of the player
2. This area will show:
   - The game which is player
   - Game specific variants chosen
3. Individual player name
4. Player Score
5. Last three player throws
6. The next number / area supposed to be hit
7. Message area used for messages (Winner, Remove Darts, . . .)

6.5 Create User

The index page can be found at path */game/manageuser*.

On this page you can manage available users.


Create a new user

You are missing a user to play? Go ahead and create one!

Name:

User list

Patrick

Jürgen

Caddi

Anna

Go back to the game setup when done!

Either you enter a name and hit *Create User* to create one or you choose a user in the list and hit *Delete Selected User* to delete the user.

You can go back to the *admin* page by hitting the link text *back*.

CHAPTER 7

Games

This page will be used to describe the different Game rules and variants.

7.1 X01

In a game of X01 the object is for one player or a team to be the first to reach zero from starting total of X01.

In simple terms, after three darts are thrown, the throwing player subtracts the total scored from his current total until he reaches zero.

7.1.1 Straight

When choosing straight no condition to start into or end the game are in place.

7.1.2 Double/Master In

In order to start substracting points each player has to hit a double (Double in) or a double/triple (Master in).

7.1.3 Double/Master Out

In order to reach zero each player must finish by throwing a double (Double out) or a double/triple (Master out). Example: If player one has 36 remaining he must hit double 18 to win, while if player two has 45 remaining he must hit single 5, double 20 to win.

7.2 Cricket

The object of Cricket is to close all numbers from 15 to 20, and the bull's eye before the opponent does. To close a number, a player must score three of that number (any combination of singles doubles and triples).

7.2.1 Scoring & Winning

Normal

Once a player closes a number every subsequent score on that number increases their score, unless the other player has closed that number as well. The player who has closed all numbers and whose score is not lower than the opponent's, wins the game.

Cut Throat

Once a player closes a number every subsequent score on that number increases the score of other players unless the other player has closed that number as well. The player who has closed all numbers and whose score is not higher than the opponent's, wins the game.

No Score

No score will be tracked throughout the game. The player who has closed all numbers wins the game.

7.3 Around the clock

The object of Around The Clock is to score all numbers from 1 to 20 in the exact order.

7.3.1 Normal

Any combination of single, double or trible will count +1 on the number which has to be hit next.

7.3.2 Fast

Single will count +1, double will count +2 and triple will even count +3 on the number which has to be hit next.

7.4 Split-Score

The goal of this game is to hit predefined numbers / target areas. Three darts are thrown on each number or target area. The hits are added to the score. If you do not hit the target with a dart, your score will be devided into half. The starting points are defined by the games variant.

Hitting order:

- 15
- 16
- Any Double
- 17
- 18
- Any Triple
- 19
- 20
- Bulls-Eye

7.4.1 E-Dart

Each player will start with 40 points.

7.4.2 Steeldart

Each player gets to throw 3 darts first. The sum of the darts will be their starting points

CHAPTER 8

TODOs

This chapter will describe open TODOs and will therefore illustrate some kind of a roadmap.

8.1 Features

- Different Sounds to choose from

8.2 Bugs

- Find out which font to install to properly display crossed circle in cricket. Otherwise this will be a square

8.3 New Games

Currently there are no new Games to be developed.

CHAPTER 9

Changelog

9.1 2019-05-15 - v0.9.9.2: Pre-release Version

Working condition: Works great so far. Still no testing with a real recognition software though.

Features added:

- Tweaked the sound a little bit

Game added:

- Split-Score
- Variants: E-Dart, Steeldart

Documentation:

- Documented the new game as well
- Pages and API are now documented

Issues fixed:

- Issue fixed

9.2 2019-04-29 - v0.9.9.1: Pre-release Version

Working condition: Pre-Release state. There should not be bugs (although there will be for sure). This is still a pre-release.

Features added:

- Redesigned a few things again to work better
- Fixed a whole lot of bugs
- Further translation into DE

Game added:

- Around the Clock (numeric order)
- Variants: Normal, Fast

Documentation has been switched over to readthedocs.org

9.3 2019-04-17 - v0.9.9: Beta version

Working condition: Near release state. There will be minor bugs (hopefully just minor ones). Therefore it is Beta.

Features added:

- Redesigned a few things to look better
- Game Controller was redesigned as well (changed order of containers)
- Rematch Button implemented
- Confirmation implemented after hitting "End Game" Button
- If Sound is on there will now also be a sound when changing player
- Checkout Possibilities shown in X01 games (Dict with checkouts, handed to game via message variable)
- Player Scores and active Player shown in gameController
- Average now is calculated right
- Last three throws now are emptied as a new throw of a new round is done

Game added:

- Cricket was added with variants Normal, Cut Throat, No Score

9.4 2019-04-10 - v0.9: Alpha version

Working condition: It might not work that stable. There will still be bugs. Therefore it is only Alpha.

Ready features:

- Game mechanism X01 is working with double/master in/out and sound
- Game Controller X01 is working to either correct throws or to insert throws like any mobile app does

CHAPTER 10

API

This chapter will describe every RESTful API call you can make. You will have to change *localhost:5000* to whatever is you configuration.

10.1 manageuser

You can add and remove player with a POST request. The *Create User* page uses this.

10.1.1 Parameter _action

- *add*: to add a user
- *del*: to delete a user

10.1.2 Parameter username

This parameter is used to add a user. User will be named like the value of this parameter.

10.1.3 Parameter delusername

This parameter is used to delete a user. User with the name like the value of this parameter will be deleted.

10.1.4 Examples

Add a user named Johnny:

```
curl -X POST 'http://localhost:5000/game/manageuser' -d "_action=add&username=Johnny"
```

Delete a user named Johnny:

```
curl -X POST 'http://localhost:5000/game/manageuser' -d "_action=del&
↪delusername=Johnny"
```

10.2 throw

You can add a throw with a GET request. The URL will look like *http://localhost:5000/game/throw/hit/mod*. The *game controller* page uses this.

10.2.1 Parameter Hit

This parameter determines which number was hit. It is supposed to be between 0-20 or 25, where 0 will be a miss 1-20 will be the corresponding number on the board and 25 will be bulls eye.

10.2.2 Parameter Mod

This parameter determines which modifier was hit. It is supposed to be between 1-3, where 1 is a hit within the big number fields or single bulls eye, 2 is a double or double bulls eye and 3 is a triple

10.2.3 Examples

Hitting a single 5:

```
curl -X GET 'http://localhost:5000/game/throw/5/1'
```

Hitting triple 20:

```
curl -X GET 'http://localhost:5000/game/throw/20/3'
```

Hitting double bulls eye

```
curl -X GET 'http://localhost:5000/game/throw/25/2'
```

10.3 throw update

You can update a throw with a GET request. The URL will look like *http://localhost:5000/game/throw/update/throwid/newhit/newmod* The *game controller* page uses this.

10.3.1 Parameter throwid

This parameter determines which throw has to be updated. Throws are counted up. So the first throw of the first player will start with id number 1 and the second throw will be number 2 and so on.

10.3.2 Parameter newhit

This parameter determines to which number the throw should be corrected. It is supposed to be between 0-20 or 25, where 0 will be a miss 1-20 will be the corresponding number on the board and 25 will be bulls eye.

10.3.3 Parameter newmod

This parameter determines to which modifier the throw should be corrected. It is supposed to be between 1-3, where 1 is a hit within the big number fields or single bulls eye, 2 is a double or double bulls eye and 3 is a triple

10.3.4 Example

Correct throw with id *1* to be triple 20 instead.

```
curl -X GET 'http://localhost:5000/game/throw/update/1/20/3'
```

10.4 nextPlayer

You can change to the next player with a GET request. The URL will look like *http://localhost:5000/game/nextPlayer* The *game controller* page uses this. Also this is used by a javascript function when no recognition is used to automatically switch user after three throws were inserted.

10.4.1 Example

```
curl -X GET 'http://localhost:5000/game/nextPlayer'
-
```

10.5 endGame

You can end the game with a GET request. The URL will look like *http://localhost:5000/game/endGame* The *game controller* page uses this.

10.5.1 Example

```
curl -X GET 'http://localhost:5000/game/endGame'
Done
```

10.6 rematch

You can trigger a rematch of a game with a GET request. The URL will look like *http://localhost:5000/game/rematch* The *game controller* page uses this.

Caution: The game is supposed to be active but won (e.g. player score is 0 and podium placement is shown in scoreboard). Will not work if 'endGame' was issued.

10.6.1 Example

```
curl -X GET 'http://localhost:5000/game/rematch'
-
```

CHAPTER 11

Donate

If you want to donat you can do so by leaving a cup of coffee or an icecold German beer.