Optimization of an irradiation instrument for photodynamic therapy

Paloma Solis[1], Mafalda Laranjo[2,3], Francisco Caramelo[2,3]

[1] University of Coimbra, Faculty of Science and Technologies, Department of Physics, [2] University of Coimbra, Institute of Biophysics and Institute of Clinical and Biomedical Research (iCBR), Faculty of Medicine, [3] University of Coimbra, Centre for Innovative Biomedicine and Biotechnology (CIBB)

Abstract:

Photodynamic therapy (PDT) has increasingly been a subject of study due to being a promising therapeutic modality for cancer treatment, while combining photosensitizers, light-sensitive drugs, and a light source. Therefore, the present study was devoted to create and optimize an irradiation instrument for PDT, by developing an open-source Python-based graphic user interface (GUI) PyPDTWiZ. The application can be used as the software to any instrument that comprises WiZ LED bulbs, offering the user the possibility to customize the irradiated light in terms of wavelength, energy and time. Different measurements performed along this study and a comprehensive view of PyPDTWiZ will be discussed throughout this paper. The application is available for download at: https://github.com/solis17/PyPDTWiZ

(1.) Introduction:

Cancer is a leading cause of death worldwide, according to the Global Cancer Observatory, with records of around 10 million deaths in 2020[1]. Chemotherapy, surgery, and radiotherapy are the most common treatments strategies for the management of these conditions and have reasonable success rates for certain types of cancer[2]. However, there are some challenges[1] such as cancer cell resistance, and serious side effects due to the lack specificity. Photodynamic therapy (PDT) is showing promising results and could represent another therapeutic strategy to overcome these problems[3].

PDT is a minimally invasive treatment that combines specific light-sensitive drugs, called photosensitizers (PS), and a light source to target and kill abnormal cells[3]. The procedure consists in firstly incorporating the photosensitizing agents into the targeted cells and then to irradiate them with light[3]. When activated by a specific wavelength and energy of light[4], the PS molecules produce reactive oxygen species that lead to the instability of the cell and consequently its death[5] . The light source properties depend on the type of photosensitizers used for the treatment[3]. Regardless of the specific light properties, the irradiation normally comes from certain kinds of lasers or from light-emitting diodes (LEDs)[3].

As PDT has becoming more widely recognized as an invaluable treatment option for the aforementioned disease[6] , the number of studies in the field has been growing at a high pace. Thus, we intend to create and optimize a software capable of controlling an in-home pre-developed irradiation instrument and its light characteristics, making it possible to be used in future vitro studies.

(2.) Methodology

The instrument system

The instrument comprises two "WiZ" LED full colour bulbs[7] with a wire and a plug so that it can be connected to the electric supply. It also has a light diffusor making the light to be evenly scattered across the cell culture plate. There are several important aspects with these types of bulbs: colour and intensity control and Wi-Fi connection. It is possible to select from a wide range (16 million[7]) of colour options and there is the choice to customize the brightness of the light. Another important feature is the use of smart Wi-Fi technology, meaning that once connected to an available Wi-Fi network the light bulb can be controlled by a smartphone, via the "WiZ" mobile app.

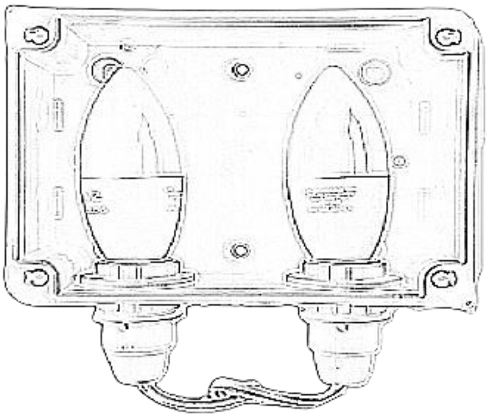

Figure 1: Representation of the irradiation instrument.

Command-line interface

As these lamps are smart devices with some degree of automation and that can be easily programmed, we developed an intuitive user interface to make easier for the user to control the device. Due to Python's versability and range of open-source[8] applications[9], we chose it as our programming tool.

Once selected the scripting language, the next step was to connect and control the light bulbs without the mobile app. This meant choosing the best Python connector for WiZ devices, within the few available options: "pywizlight"[10] and "wizcon"[11]. Despite both of these packages being able to connect to the lamps, only the first one was suitable to build our software as it provided us more functions and methods[12] – like setting or getting a specific light feature. When allying Python's library "asyncio"[13] with "pywizlight", we were able to run coroutines concurrently and control their execution, for instance, making it possible to turn on the lamp at a specific brightness and colour for a given amount of time.

It is important to note that there is still a dependence to the "WiZ" app. The lamps we are using, have to be connected to the app and share the same Wi-Fi network with the computer that runs the program. Only then, the IP address can be provided. Unfortunately, 5GHz Wi-Fi or open networks without password protection are not supported by the app[14], and thus cannot be used with our application.

Graphic user interface

Firstly, we implemented a Python command-line interface (CLI) that emulated the app, and was able of executing all the options, which was used as the basis of our software. Although being functional, this text-based program was not much intuitive or user-friendly and quite limited regarding the routines defined for the lamp. The solution was to convert it into a graphic user interface (GUI) with the Python's library "PyQt5". This tool allowed us to create a visual representation of the software in which the user could easily interact through clickable icons [15]. After designing and achieving a working and user-friendly GUI program, it was crucial to make it optimal for future phototherapy studies. This meant analysing the instrument's characteristics to find the relation between the levels of brightness and colour of the lamps and the wavelength and energy of the irradiated light.

Colour and lights' wavelength

"Pywizlight" uses the RGB system, that combines red, green and blue in various proportions to create different colours[12]. There are three levels: R (red), G (green) and B (blue), each one being represented by the range of integer numbers from 0 to 255[16]. Although being used in many devices displays, this system is not suitable for laboratory environments, due to the lack of information about the light wavelength. To solve this problem, we used the Ocean Optics spectrophotometer to analyse the red, green and blue monochromatic lights, RGB(255,0,0), RGB(0,255,0) and RGB(0,0,255), respectively. After analysing the spectrums, we were able to associate a reduced range of wavelength values to each one of the three RGB colours. We, then, made the necessary changes to our software, so the user could choose between one of these wavelengths.

Brightness and lights' power

The level of brightness of the lamp is given by the range of integer numbers from 0 to 255. In order to get the relation between the brightness of the lamp and the luminance of the light, we used the Apacer spectroradiometer. For each one of the three aforementioned wavelength options, we measured the luminance of the lamps in three different positions, as shown in **Fig.2**, for evenly spaced values of brightness. Because the obtained results were given in Candela per square meter (cd/m^2), we had to make use of the equations, **Eq.(1)-(4)** to get results in Joule per square meter (J/m^2).

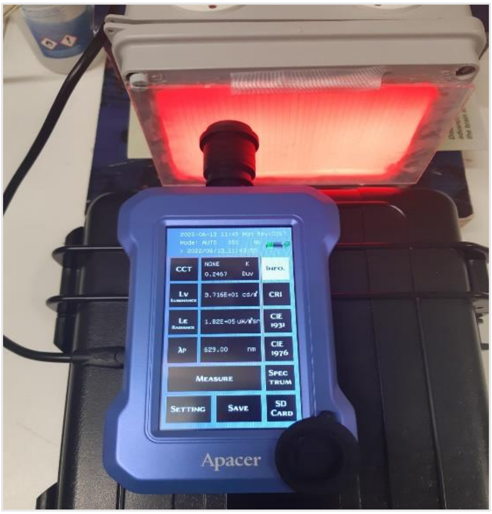

Figure 2: Irradiation instrument and spectroradiometer in one of the three measurement positions.

The luminous flux (Φ) in lumens, for an uniform and isotropic light, equals the luminous intensity (I) in candela times the solid angle in steradians (Ω) [17],

$$\Phi_{(lm)} = I_{(cd)} \times \Omega_{(sr)} \qquad \textbf{(1)}$$

The solid angle in steradians, with θ being the apex angle, is given by[17]:

$$\Omega_{(sr)} = 2\pi(1 - \cos(\theta/2)) \qquad \textbf{(2)}$$

The power per area, P, (W/m^2) is given by the quotient between the luminous flux(Φ), in lumens, and the luminous efficacy (η), in lumens per Watt [19],

$$P_{(W/m^2)} = \frac{I_{(cd/m2)} \times 2\pi(1 - \cos(\theta/2))}{\eta_{(lm/W)}} \qquad \textbf{(3)}$$

Knowing how much power (P) per area (W/m^2) is consumed and the time length (s), we can calculate the total energy used (E) per area (J/m^2),

$$E = P_{(W/m^2)}\,\Delta t_{\ (s)} \qquad \textbf{(4)}$$

Standalone application

Our goal was to develop a software that runs in any device, without requiring Anaconda or other distribution of Python programming language. Therefore, once the final adjusts were made, we created the standalone application of our software, an operating system capable of functioning independently of other hardware or connection[19], using the Python's package "pyinstaller"[20].

(3.) Results:

Graphic user interface

The first GUI we developed was a simple WiZ app emulator. The user was able to turn on the lamp for a limited or unlimited time (by inserting "+") at specific levels (0-255) of brightness and RGB. Besides that, it was also possible to set one of the pre-defined light mode scenes from the app.

Spectrophotometer analysis

Fig.3 (a) shows the curves of intensity observed with the spectrophotometer's software, for three specific light colours: RGB (255,0,0), RGB (0,0,255) and RGB (0,255,0).

Defining the curve peak single value as our wavelength was not a valid option, as it meant working with significant uncertainty. Instead, we considered a range of values with high probability of containing the actual wavelength value. Fitting a gaussian distribution to each one of the obtained curves, **Fig.3 (b)-(d)**, we were able to get the parameters listed in **Tab.1**

Table 1:
Parameters of the gaussian fitting curve and associated range of values, for each one of the three light colours. All values, except the coefficient of determination (R^2), are in nanometres (nm) units.

	(255,0,0)	**(0,255,0)**	**(0,0,255)**
Centre (c)	628,77	519,80	465,61
Width (w)	16,47	28,07	25,19
Variance (**σ**)	8,24	14,03	12,60
FWHM	19,40	29,65	33,05
Range of values	629 ± 19	520 ± 28	466 ± 30
R^2	0,9865	0,9882	0,9860

When considering c being the centre of the distribution, σ the variance, w the width, FWHM the full width at half maximum, the range of values in **Tab.1** are given by,

$$\sigma = \frac{w}{2} \quad \textbf{(5)}$$

$$\text{FWHM} = 2\sqrt{2\,ln2}\;\sigma \quad \textbf{(6)}$$

$$\text{Range of Values} = \text{c} \pm \text{FWHM} \quad \textbf{(7)}$$

Table 2:
Comparison between obtained and tabled wavelength ranges for each light colour.

Colour	Obtained range (nm)	Tabled range (nm)	Overlapping
Red	610-648	625-740	✓
Green	492-548	520-565	✓
Blue	436-496	430-500	✓

Spectroradiometer analysis

Using a data set with the spectroradiometer measures, for each one the three positions at a given wavelength, we were able to graph the luminance of the light in function of the intensity, **Fig.4 (a),** and apply a ninth-degree polynomial curve fitting, **Fig.4 (b)-(d),** to each one of the three wavelengths.

For each curve fitting we obtained a ninth-degree polynomial and in order to simplify it we ignored every term with coefficient extremely close to zero (order of magnitude bellow or equal to minus 5). The third-degree polynomials coefficientes are listed in **Tab.3**.

Table 3:
Parameters (a) and coefficients of determination (R^2) obtained for the third-degree fitting polynomial ($y = a_0 + a_1x + a_2x^2 + a_3x^3$), for each one of the three light colours.

Coefficient	(629 ± 19) nm	(466 ± 30) nm	(520 ± 28) nm
a_0	4,79 ± 0,70	2,03 ± 0,28	9,16 ± 1,38
a_1	0,18 ± 0,23	0,05 ± 0,09	0,31 ± 0,45
a_2	-0,026 ± 0,021	-0,007 ±0,008	-0,045 ± 0,041
a_3	0,0012 ± 0,0008	0,0004 ± 0,0003	0,002 ± 0,002
R^2	0,9949	0,9999	0,9999

Since one of the requirements was to allow the user introducing the wanted energy per area (J/m^2) and receive information about the needed time for the irradiation, we use **Eq.(4)** for that purpose. To reduce the time length of each irradiation, we only worked with the maximum luminance of each light color, obtained at the maximum value of intensity of light (255), **Tab.4**. In addition, the values considered for the apex angle of the spectroradiometer, and the lamp efficacy listed in WiZ's product specifications[21], are also listed in **Tab.4**.

Table 4:
Parameters to calculate time of irradiation.

Constant	Value	Description
$I_{max\ red}$	127.67 cd/m^2	Maximum luminance for red light at maximum brightness (255)
$I_{max\ green}$	262.02 cd/m^2	Maximum luminance for green light at maximum brightness (255)
$I_{max\ blue}$	59.35 cd/m^2	Maximum luminance for blue light at maximum brightness (255)
η	91 lm/W	Lamp efficacy
θ	2º	Apex angle

Given **Tab.4** constants and **Eqs.(1)-(4)**, we were able to calculate, for each wavelength, the time needed to irradiate the cells, Δt, with the energy per area value defined by the user, E (J/m^2).

$$\Delta t = \frac{E \times \eta}{I_{max} \times 2\pi\left(1 - cos\,(\theta/2)\right)} \quad \textbf{(8)}$$

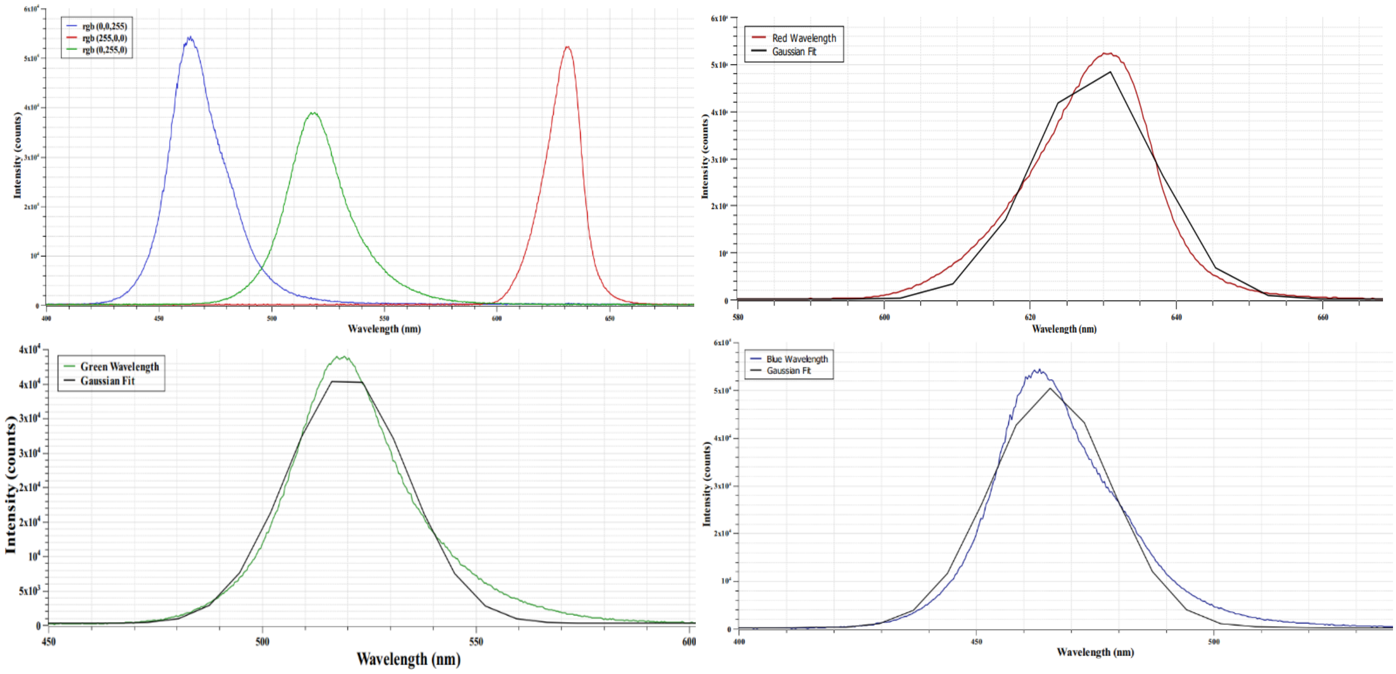

Figure 3: Spectrums with intensity (counts) in function of the wavelength. **(a)** RGB(255,0,0), RGB(0,255,0) and RGB(0,0,255) spectra (red, green and blue lines, respectively); **(b)** RGB(255,0,0) spectrum (red line) and gaussian fitting (black line); **(c)** RGB(0,255,0) spectrum (green line) and gaussian fitting (black line); **(d)** RGB(0,0,255) spectrum (blue line) and gaussian fitting (black line).

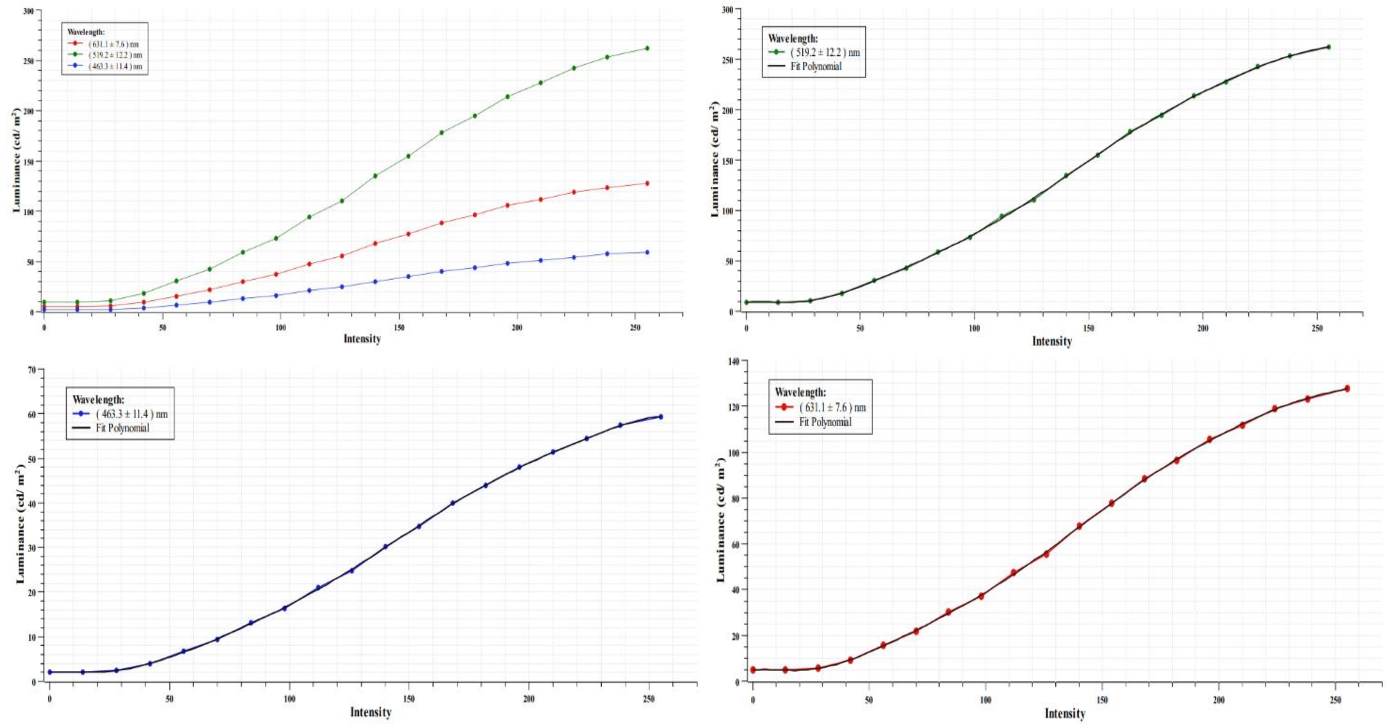

Figure 4: The curves of luminance against different intensity values. **(a)** Red, green and blue light curves (red, green and blue lines, respectively); **(b)** red light curve (red line) and polynomial fitting (black line); **(c)** green light curve (green line) and polynomial fitting (black line); **(d)** blue light curve (blue line) and polynomial fitting (black line);

Final Interface/Standalone application

Fig.5 depicts the final GUI, which was implemented after carrying out all tests and applied all adjusts. As our final step, we created the standalone application of PyPDTWiZ.

To make use of this application, the user has to download the executable file named PyPDTWIZ.exe and save it in the system drive. After that, for a correct use of PyPDTWiZ application, the user should take the following steps:

1. Choose IP addresses
   Option (a): By manually inserting them;
   Option (b): By searching all bulbs in the network broadcast space.
2. Select only one of the wavelength ranges values;
3. Insert energy value (in Joule per square meter);
4. Calculate required time (in seconds);
5. Set all values;
6. Turn on the light;
7. Clear all values.

(4.) Discussion:

Throughout this work, our aim was to develop a software application that was capable of providing the user the opportunity to adapt the lamp characteristics to different photodynamic test requisites. Since the type of information given by the WiZ connector was not compatible for laboratory environments, we had to identify a relation to the common light properties (wavelength and intensity), by performing the previously described procedures.

Regarding the spectrophotometer analysis, **Tab.2** compares the ranges of wavelength obtained during our work to the range of wavelength for red, green and blue subregions of the visible electromagnetic spectrum[22]. For each one of the RGB colours tested, we verify an overlapping. Although for red and green we do not have an absolute overlap, the centre value of each wavelength range is comprised in the respective intervals listed in **Tab.1**.

On the other hand, the spectroradiometer analysis was able to provide information about the light's intensity. We verified that lower levels of brightness were directly associated with lower luminance and consequently

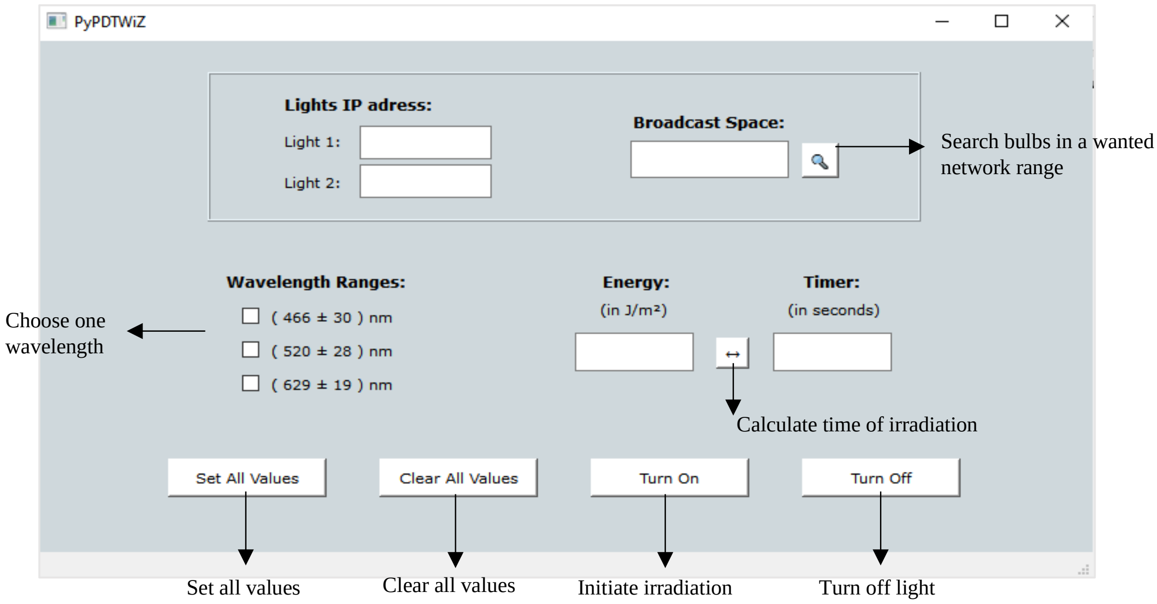

Figure 5: Final graphic user interface (GUI) display.

longer irradiation time intervals, so we decided to only use the lights at their maximum brightness. **Eq.(8)** made it possible to implement a new option on our software: the user can provide input about the energy per area and automatically receive information about the time (s) needed to reach that value. Comparing the luminance obtained for each light colour at maximum brightness, **Tab.4**, for the same energy irradiated we can expect to have a shortest time span for green light and the longest for the blue light.

Finally, we obtained the last version of our GUI program and created the standalone software, PyPDTWiZ. It is important to highlight the role of “WiZ” mobile app: without having the lamps paired to the app, the user won’t be able to use PyPDTWiZ. Only after having the lights connect to the app, the IP addresses become accessible and visible in the network. We believe that this dependency to the mobile app is the biggest limitation of our application and the main aspect that should be improved in future studies. Other recommendations for further researches would be to increase the list of possible wavelengths, shorten the range values and having more options to work with different levels of brightness.

(5.) Conclusion:

In this project we presented the development and optimization of a standalone graphic user interface (GUI) software application, PyPDTWiZ, written in Python’s programming language. Once connected to a lamp paired in “WiZ” mobile app, within the Wi-Fi network range, this program provides the necessary tools to unable this irradiation instrument to perform future photodynamic therapy studies. Without prior coding experience being required, the available options allow the user to customize the irradiated light in the terms of wavelength and energy.

(6.) Computer code availability

Name of the code: PyPDTWiZ
Contact: Paloma Solis (psolis2204@gmail.com), Francisco Caramelo, Mafalda Laranjo
Program language: Python 3.6
Software Required: Windows (64 bit if using.exe file)

Source code: The source code is available for download at the link: https://github.com/solis17/PyPDTWiZ

Acknowledgments:

The authors thank Coimbra Institute for Clinical and Biomedical Research (iCBR), from the Faculty of Medicine, University of Coimbra (FMUC), for supporting this work.

References:

[1] Gyanani, V., Haley, J. C., & Goswami, R. (2021). Challenges of current anticancer treatment approaches with focus on liposomal drug delivery systems. In Pharmaceuticals (Vol. 14, Issue 9). Available at: https://doi.org/10.3390/ph14090835

[2] Algorri, J. F., Ochoa, M., Roldán-Varona, P., Rodríguez-Cobo, L., & López-Higuera, J. M. (2021). Photodynamic Therapy: A Compendium of Latest Reviews. Cancers, 13(17), 4447. Available at: https://doi.org/10.3390/cancers13174447

[3] Cancer.org, 2018, PDT | What is PDT? | Photodynamic Therapy, [Online] Available at: https://www.cancer.org/treatment/treatments-and-side-effects/treatment-types/radiation/photodynamic-therapy.html , Accessed 9 Jun. 2022

[4] Allison, R. R., & Moghissi, K. (2013). Photodynamic therapy (PDT): PDT mechanisms. Clinical Endoscopy, 46(1). Available at: https://doi.org/10.5946/ce.2013.46.1.24

[5] National Cancer Institute, 2022, NCI Dictionary of Cancer Terms, [Online] Available at: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/photosensitizing-agent, Accessed 9 Jun. 2022.

[6] Yoo, J.-O. and Ha, K.-S. (2012). New Insights into the Mechanisms for Photodynamic Therapy-Induced Cancer Cell Death. International Review of Cell and Molecular Biology, [Online] pp.139–174. Available at: https://doi.org/10.1016/B978-0-12-394306-4.00010-1

[7] WiZ, 2018, LED Lamp Candle B12 E12WiZ. [Online] Available at: https://www.wizconnected.com/en-us/p/led-lamp-candle-b12-e12/046677603526, Accessed 12 Jun. 2022.

[8] Python.org, 2022, History and License Python 3.10.5 documentation, [Online], Available at: https://docs.python.org/3/license.html , Accessed 12 Jun. 2022.

[9] Coursera, 2022, What Is Python Used For? A Beginner’s Guide, [Online] Available at: https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python, Accessed 10 Jun. 2022.

[10] sbidy, 2022, sbidy/pywizlight: A python connector for WiZ devices. [Online] GitHub, Available at: https://github.com/sbidy/pywizlight, Accessed 10 Jun. 2022.

[11] rgomezjnr, 2021, rgomezjnr/wizcon: Control Philips WiZ Connected smart light bulbs, [Online] GitHub. Available at: https://github.com/rgomezjnr/wizcon, Accessed 10 Jun. 2022.

[12] Sbidy, 2022, GitHub - sbidy/pywizlight: A python connector for WiZ devices, [Online] GitHub, Available at: https://github.com/sbidy/pywizlight#classes , Accessed 10 Jun. 2022.

[13] Python.org, 2022, asyncio — Asynchronous I/O — Python 3.10.5 documentation, [Online] Available at: https://docs.python.org/3/library/asyncio.html, Accessed 10 Jun. 2022.

[14] WiZ, 2018, Support WiZ, [Online] Available at: https://www.wizconnected.com/en-us/support, Accessed 12 Jun. 2022.

[15] MrNayaNi, 2020, Command Line Interface (CLI) vs. Graphical User Interface (GUI), [Online] Cybrary, Available at: https://www.cybrary.it/blog/0p3n/command-line-interface-cli-vs-graphical-user-interface-gui/, Accessed 12 Jun. 2022.

[16] TechTarget Contributor, 2022, RGB (red, green, and blue), [Online] WhatIs.com, Available at: https://www.techtarget.com/whatis/definition/RGB-red-green-and-blue, Accessed 13 Jun. 2022.

[17] Rapidtables.com, 2022, Candela to lumens (lm) conversion calculator, [Online] Available at: https://www.rapidtables.com/calc/light/candela-to-lumen-calculator.html, Accessed 15 Jun. 2022.

[18] Rapidtables.com, 2022, How to convert lumens to watts (W), [Online] Available at: https://www.rapidtables.com/calc/light/how-lumen-to-watt.html, Accessed 15 Jun. 2022.

[19] Techterms.com, 2020, Standalone, [Online] Available at: https://techterms.com/definition/standalone, Accessed 17 Jun. 2022.

[20] PyPI, 2022, pyinstaller, [Online] Available at: https://pypi.org/project/pyinstaller/, Accessed 17 Jun. 2022.

[21] WiZ, 2018, LED Lamp Candle B12 E12 WiZ, [Online] Available at: https://www.wizconnected.com/en-us/p/led-lamp-candle-b12-e12/046677603526#specifications, Accessed 17 Jun. 2022.

[22] D. Malacara, *Color* Vision and Colorimetry: Theory and Applications, Second Edition, SPIE Press, Bellingham, WA (2011), Available at: https://doi.org/10.1117/3.881172