



UNIVERSITAT DE  
BARCELONA

Final Degree Project

**Biomedical Engineering Degree**

**Monte Carlo simulation  
software for realistic SPECT  
studies**

Barcelona, June 7, 2023

Author: Cristina Martín Barceló

Directors: Dra. Aida Niñerola Baizán and  
Dr. Raúl Tudela Fernández

Tutor: Dra. Aida Niñerola Baizán

## Acknowledgements

En primer lloc, volia agrair la meva tutora, l'Aida Niñerola, per oferir-me la possibilitat de fer aquest treball, que he patit però he gaudit moltíssim, ja que m'ha fet aprendre molt sobre una branca de l'Enginyeria Biomèdica que no sabia si se'm donaria bé. Gràcies per guiar-me i donar-me suport cada vegada que ho he necessitat. També agrair al Raúl Tudela per donar-me suport amb els problemes de programació que apareixien pel camí. Agrair en general al GIB-UB i la organització dels seminaris setmanals, ja que m'ha fet conèixer projectes molt interessants del món de la investigació d'imatges biomèdiques i estar en contacte amb gent d'aquest entorn.

D'altra banda, volia agrair els meus amics per estar al meu costat, quasi tots estàvem en el mateix moment i el suport conjunt ha estat important per seguir endavant en moments difícils. En especial, agrair l'Alba, per fer juntes aquest camí i donar-nos tot el suport que podíem, i a la Maria, per totes les hores passades juntes intentant treure endavant els nostres projectes.

Finalment, volia agrair a la meva família, per creure amb mi i donar-me suport des de la distància. I també als meus companys de pis, la Marta i el Miquel, gràcies per aguantar-me teclejant tot el dia i ser la meva família aquí a Barcelona, i sobretot, per cuidar-nos sempre mútuament.

## Abstract

Parkinson's disease is the second most prevalent neurodegenerative disorder worldwide, and its incidence is rising every year. Physical and psychological symptoms are present in PD, and as the condition progresses, these symptoms lead to high rates of disability and care needs.

In some cases, a DaTscan SPECT test is performed to differentiate the essential tremor of the disease from other central nervous system diseases. This test is used to detect the loss of dopaminergic nerve endings in the striatum region of the brain. Lately, the potential benefits for PD patients of performing this scan for diagnosing has been studied. Moreover, even if brain  $^{123}\text{I}$ -loflupane SPECT scans are typically interpreted by the professionals' visual inspection, an increase interest in  $^{123}\text{I}$ -loflupane quantification methods to quantify radioactive distribution in the nigrostriatal region has been observed, leading this to the release of several quantification and semiquantification methods collected in commercial software. However, its generalized use has been hindered because of the lack of standardization between those methodologies.

This project aims to develop a software which integrates Monte Carlo simulation techniques and reconstruction algorithms to generate realistic SPECT images. The simulated SPECT images can be used to validate the quantification methods using the original digital information as ground truth.

To achieve this, a python program based on SimSET and STIR packages has been developed and integrated into a graphical interface to choose files and display results from both processes. The obtained results were generally good, as the objectives of the project were met, but future lines of work have been described in order to optimise and improve the final result.

---

**Keywords:** loflupane; quantification; SPECT; DaTscan; Monte Carlo; simulation; reconstruction; standardization.

## List of figures

<b>Figure 1.</b> SPECT acquisition and tomographic reconstructions. Obtention of 1D projection profiles and final reconstructed image.....	4
<b>Figure 2.</b> Obtention of the sinogram from 1D projection profiles.....	5
<b>Figure 3.</b> DATscan imaging technique. Left) DaTscan from a normal patient. Right) DaT scan from a patient with loss of transporters of dopamine.....	6
<b>Figure 4.</b> Process of the simulation of a DaTscan SPECT study. ....	7
<b>Figure 5.</b> Summary of inputs and outputs of the MC simulation SimSET package (Own source).....	8
<b>Figure 6.</b> Simple shell scripting code to run a simulation process (GIB-UB).....	9
<b>Figure 7.</b> Simple shell scripting code to apply a Butterworth and FBP filter to reconstruct input image. (GIB-UB).....	9
<b>Figure 8.</b> Graphic illustrating searches in PubMed of SPECT and loflupane quantification from 1980 to 2020. [23].....	10
<b>Figure 9.</b> BasGan algorithm applied in (a) a control subject and in (b) a PD patient. The VOIs positioned by BasGan software are identified by black dotted lines [24]. ....	11
<b>Figure 10.</b> DaTQUANT save screen. White VOI placement in the images segmenting striatal region. Normal database on top right, marked with red square [25]. ....	11
<b>Figure 11.</b> Hoffman 3-D Brain Phantom cylinder and 3-D Brain Insert [26].....	12
<b>Figure 12.</b> Striatal Phantom [27] .....	12
<b>Figure 13.</b> Data and scripts organisation in the working directory (Own source). ....	18
<b>Figure 14.</b> Workflow of the simulation and reconstruction programs using SimSET and STIR packages (Own source). ....	18
<b>Figure 15.</b> UML diagram showing the different classes and files (main and auxiliary) to accomplish SPECTSim performance (Own source). ....	19
<b>Figure 16.</b> Block diagram showing the flux of the simulation program. The inputs, the program flow and the outputs (Own source).....	22
<b>Figure 17.</b> Block diagram showing the flux of reconstruction program. The input, the process, and the outputs. Also the reconstruction algorithms considered for the program (Own source).....	23
<b>Figure 18.</b> Widgets and buttons diagram of SPECTSim graphical interface (Own source). ....	26
<b>Figure 19.</b> Scheme showing the inputs and outputs needed for both simulation and reconstruction flows (Own source). ....	27
<b>Figure 20.</b> Activity distribution map showing striatal nuclei (left) and attenuation map (right) of striatal phantom in axial view. (Source: ITK-SNAP) .....	28
<b>Figure 21.</b> Simulation results visualised with (left) MRlcro and (right) nilearn library from Python (Own source).....	28
<b>Figure 22.</b> Coronal (y) view of sinogram for 5.000.000 photons simulated. (Source: MRlcro). ....	29
<b>Figure 23.</b> Coronal (y) view of sinogram for 15.000.000 photons simulated. (Source: MRlcro). ....	29
<b>Figure 24.</b> Coronal (y) view of sinogram for 100.000.000 photons simulated. (Source: MRlcro). ....	30
<b>Figure 25.</b> Reconstruction with FBP algorithm. CSA order plotted in jet colour map. The colour map has been altered to distinguish better the brain regions with the background. Zoom of 1.5 applied. ((Source: ITK-SNAP).....	30

<b>Figure 26.</b> Reconstruction with OSEM algorithm. CSA orde, plotted in jet color map r. (Source: ITK-SNAP)	30
<b>Figure 27.</b> Main window of GUI (Own source).	31
<b>Figure 28.</b> File explorer window opened when the select buttons are clicked (Own source).	31
<b>Figure 29.</b> Simulation pop-up window (Own source).	32
<b>Figure 30.</b> List of options to choose the reconstruction filter or algorithm preferred (Own source).	32
<b>Figure 31.</b> Entries to introduce the reconstruction parameters for (left) FBP filters and for (right) iterative OSEM algorithm (Own source).	33
<b>Figure 32.</b> Reconstruction pop-up window (Own source).	34
<b>Figure 33.</b> Work Breakdown Structure (WBS) of the project (Own source)	36
<b>Figure 34.</b> GANTT diagram (Own source).	40
<b>Figure 35.</b> SWOT diagram to study technical viability (Own source).	41
<b>Figure 36.</b> Pie chart showing the cost distribution of the project (Own source).	44
<b>Figure A1.</b> Block diagram illustrating the fit between the different modules of SimSET package.	52
<b>Figure A2.</b> PHG inputs and outputs.	53
<b>Figure A3.</b> Control flow for tracking photons in PHG module.	53
<b>Figure A4.</b> Fan- beam and Cone-beam collimation in UNC SPECT simulation.	55
<b>Figure A5.</b> Front face of collimator in a UNC SPECT simulation.	55
<b>Figure A6.</b> Example of a STIR interfile for a SPECT reconstruction.	58
<b>Figure A7.</b> Example of STIR file to apply a FBP2D reconstruction.	59
<b>Figure A8.</b> Example of STIR file to apply a FBP3D reconstruction.	59
<b>Figure A9.</b> Example of STIR file to apply a OSEM2D reconstruction.	60
<b>Figure A10.</b> Example of STIR file to apply a OSEM3D reconstruction.	60

## List of tables

**Table 1.** Statistical summary of simulation for different numbers of simulated photons

**Table 2.** WBS dictionary describing each task.

**Table 3.** PERT, Optimistic, normal and pessimistic time of each activity.

**Table 4.** Materials cost of the project.

**Table 5.** Services cost of the project.

**Table 6.** Personnel cost of the project.

**Table A1.** Complete cost of the project.

**Table A2.** Considered parameters to configure PHG module.

**Table A3.** Considered parameters to configure collimator module.

**Table A4.** Considered parameters to configure detector module.

**Table A5.** Considered parameters to configure binning module.

## List of Abbreviations

**SPECT:** Single photon emission computed tomography

**PET:** Positron emission tomography

**PD:** Parkinson's Disease

**DaTscan:** Dopamine Transporter scan

**MC:** Monte Carlo

**SimSET:** Simulation System for Emission Tomography

**SimPET:** Simulation for Positron Emission Tomography

**GUI:** Graphical User Interface

**2D:** Two-dimensional

**3D:** Three-dimensional

**PHG:** Photon History Generator

**STIR:** Software for Tomographic Image Reconstruction

**FBP:** Filtered Back Projection

**OSEM:** Ordered Subset Expectation Maximization

**GIB-UB:** *Grup d'imatges biomèdiques – Universitat de Barcelona*

**CT:** Computed Tomography

**VOI:** Voxel of Interest

## Table of contents

<b>ACKNOWLEDGEMENTS.....</b>	<b>I</b>
<b>ABSTRACT .....</b>	<b>II</b>
<b>LIST OF FIGURES .....</b>	<b>III</b>
<b>LIST OF TABLES.....</b>	<b>V</b>
<b>LIST OF ABBREVIATIONS.....</b>	<b>VI</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 Objectives of the project .....	2
1.2 Methodology and structure of the project.....	2
1.3 Scope.....	3
1.4 Location of the project.....	3
<b>2. BACKGROUND.....</b>	<b>3</b>
2.1 State of the art .....	3
2.1.1 <i>SPECT imaging technique</i> .....	4
2.1.2 <i>Monte Carlo simulation</i> .....	6
2.2 State of the situation .....	8
<b>3. MARKET ANALYSIS .....</b>	<b>10</b>
3.1 Sectors to which the tool is directed .....	10
3.2 Historical evolution of the market.....	10
3.3. Future market prospects .....	12
<b>4. CONCEPT ENGINEERING .....</b>	<b>14</b>
4.1 Solutions study.....	14
4.1.1 <i>Monte Carlo simulation toolkits</i> .....	14
4.1.2 <i>Reconstruction toolkits</i> .....	15
4.1.3 <i>Programming language to develop the main program</i> .....	15



4.1.4 Programming language to develop the graphical interface .....	15
4.2 Proposed solution .....	16
4.3 Alternative solution.....	16
<b>5. DETAILED ENGINEERING .....</b>	<b>18</b>
5.1 Workflow of simulation and reconstruction process .....	18
5.2 Python code for the back end (main program).....	19
5.2.1 Definition of the program .....	21
5.2.2 Main program .....	21
5.2.3 Auxiliary function files.....	24
5.3 Python code for the front end (GUI development) .....	25
5.4 Results.....	27
5.5 Discussion .....	34
<b>6. EXECUTION SCHEDULE .....</b>	<b>36</b>
6.1 Tasks and time definition .....	36
6.1.1 Work Breakdown Structure (WBS).....	36
6.1.2 WBS dictionary.....	37
6.2 Precedence analysis and critical path method .....	38
6.3 Schedule; GANTT diagram.....	40
<b>7. TECHNICAL VIABILITY .....</b>	<b>41</b>
<b>8. ECONOMIC VIABILITY .....</b>	<b>43</b>
<b>9. LEGISLATION AND REGULATION .....</b>	<b>45</b>
<b>10. CONCLUSIONS AND FUTURE WORK .....</b>	<b>46</b>
<b>BIBLIOGRAPHY .....</b>	<b>47</b>
<b>ANNEXES .....</b>	<b>51</b>



## 1. Introduction

Parkinson's disease (PD) is a chronic and progressive neurodegenerative brain disorder which affects the nervous system, mainly the motor system. The symptomatology ranges from physical symptoms like involuntary tremors, muscular rigidity, and inflexibility, slowed motion or loss of automatic movements to psychological symptoms (depression and anxiety, problems with body balance, insomnia, memory problems, etc). As the disorder progresses, symptoms result in high rates of disability and care requirements [1]. PD is the second most common neurodegenerative disorder after Alzheimer's disease, its prevalence is higher each year, and disability and death due to this disease are increasing more than for other neurological disorders [2].

Pathologically, PD is characterised by a loss of the dopaminergic neurons in the substantia nigra pars compacta (SNpc) brain region, which causes a loss of dopamine in the striatum, a brain region involved in the control of movement, leading to the known symptomatology (tremor at rest, rigidity, etc.). Emission tomography imaging, such as Positron Emission Tomography (PET) and (Single Photon Emission Computed Tomography (SPECT) has contributed to a better understanding of PD. Specifically, SPECT with the radiotracer [<sup>123</sup>I] IFP-CIT, commercially known as DaTscan, is a useful tool for dopaminergic neurons evaluation [3].

SPECT imaging is a type of diagnostic imaging within the field of nuclear medicine which is commonly used in neurology to study brain function and neurological disorders like PD, epilepsy, and dementia. Even if the interpretation of these images is typically through visual inspection, there is a growing interest in employing quantitative and semiquantitative approaches to improve the diagnostic accuracy in brain disorders [4]. Hence, the use of commercial software for semiquantitative analysis has been progressively introduced in clinical practice. However, there is a lack of standardisation among quantification methods, which hinders their widespread use [5].

The importance of quantification in emission tomography – PET and SPECT imaging – is that it allows to measure the amount of injected radiopharmaceutical present in a specific area of the body. In other words, it allows to analyse data quantitatively on the distribution of the radiotracers throughout the body. In any case, the different quantification methods must be validated, as well as the acquisition protocols to acquire the image and reconstruction methods to obtain the three-dimensional view. This can be done either with phantoms or simulations. With phantoms, an accurate validation may be performed, but it has the drawback that since it is a physical body, it must be prepared by filling radiotracers to measure the radioactivity emitted and the image must be taken with a PET or SPECT machine. [6] The downside of this practice is that every time a new study is performed, the entire process of preparing and imaging the phantom must be repeated, which is not optimal to analyse many different scenarios. On the other hand, working with Monte Carlo (MC) or analytical simulations solves the problem of the preparation and physical study, as the simulated data allows the generation of many different images from different scenarios. Loads of data may be obtained by only changing parameters like the quantity



of dose, the geometry of the phantom or machine, among many others, without having to work with physical phantoms. So, MC simulation is now an essential technique in nuclear medicine imaging, going from the aim to optimize the design of imaging systems, accuracy of reconstruction methods and to validate quantitative and semiquantitative methods [7]. Nevertheless, generating nowadays simulated SPECT images is not a trivial task, since the use of all related software requires a solid background in programming as well as access to high computing power, making MC simulation often inaccessible to the specialized community which implements these software tools, thus limiting its outreach to clinical facilities, where quantitative methodologies must be validated.

For that reason, the main goals are to make simulation more accessible from an accessible platform. Developing a program which generates simulated data and reconstructs it through a user-friendly graphical interface will facilitate the integration of simulations as a validation tool in the field of research, business, and healthcare. Following this line, the project focuses on the development of SimSET-based software to simulate realistic SPECT images to contribute to the improvement of diagnostic imaging. It will be based on the SimPET project [5], developed to obtain realistic PET images.

### 1.1 Objectives of the project

The main objective of this project is to develop a MC simulation software to generate realistic images from SPECT neuroimaging studies. Regarding this purpose, it should be divided in three small aims:

- Generate the simulation class to generate a sinogram of the projections simulated
- Generate the reconstruction class to perform the reconstruction of the simulated projections

As a second objective, following the line of this first one, is to make a user-friendly, easy to use and visual interface. It is key that the software is designed to be user-friendly and intuitive for the professionals it is aimed at.

The third aim is that when all the project purposes are fulfilled, to work on brief manuals for the user, as it will be a software destined to research studies and not all professionals are used to work with image simulations, and they are not expected to have knowledge on all the parameters files needed to run the simulation.

### 1.2 Methodology and structure of the project

The methodology used to develop the project can be divided into three main parts that are in the same line as the objectives of the project.

- Development of the simulation program.



- Development of the reconstruction program.
- Integrate simulation and reconstruction programmes into a graphical interface

The timeline of the project has been from January to June of 2023, and every week a meeting to catch up on the project was set.

The structure of the project report is divided into ten sections: introduction, background, market analysis, conception engineering, detailed engineering, execution schedule, technical viability, economic viability, legislation and regulation aspects and conclusion and future work. With this, this memory pretends to give a clear idea of how the project was developed and all technical concepts it relies on.

### 1.3 Scope

The scope of the project includes the listed tasks:

- Bibliographic research related to MC simulation technique and reconstruction algorithms and filters for SPECT.
- Bibliographic research related to nuclear medicine general concepts and quantification.
- Familiarization with the scripts that contain the program to run SimSET.
- Familiarization with SimPET code.
- Develop the simulation program based on SimSET.
- Develop the reconstruction program based on STIR.
- Develop a graphical interface (GUI) integrating simulation and reconstruction programs.

### 1.4 Location of the project

The project has been performed in collaboration with the biomedical imaging group (GIB-UB) of the Biophysics and Bioengineering Unit from University of Barcelona (UB). This group focuses on the development of models and methods for the acquisition, treatment, and study of biomedical images, including nuclear medicine simulation methods, the main line of research of this project. So, it is important to mention the relation between the group and the service of nuclear medicine of *Hospital Clínic* of Barcelona, as it is the place where I developed my student internship with Aida Niñerola Baizán, the tutor of the project, learning about general concepts regarding this project among other things. As the project consists of an open-source project, the main work has been able to be developed on remote, but GIB-UB lab has also been a workplace.

## 2. Background

### 2.1 State of the art

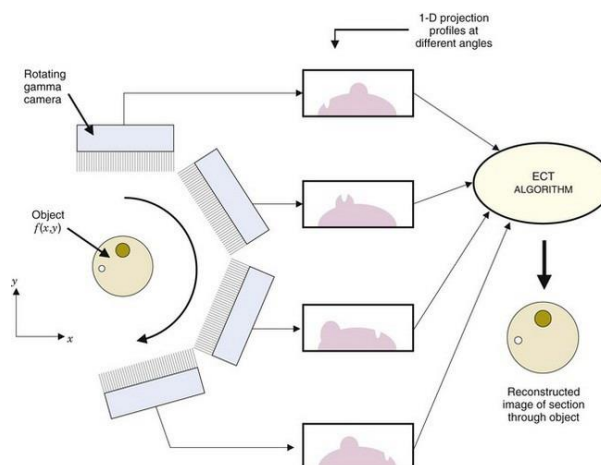
The imaging technique data aimed to simulate is SPECT, which is under the field of nuclear medicine. As the data obtained are projections from each view captured, these need to be reconstructed to obtain the final 3D image. For this, several reconstruction methods, which will be

explained on the next subparts, may be used. All this can be simulated using MC technique, which is particularly useful to validate quantification and semi quantification methods or new reconstruction algorithms as ground truth is known. In the next subsections, mentioned topics will be explained in detail.

### 2.1.1 SPECT imaging technique

SPECT imaging is a type of diagnostic imaging within the field of nuclear medicine where a small amount of a radioactive substance attached to a specific compound (known as a radiotracer or radiopharmaceutical) is administered which travels to the organ or tissue to be studied. As the radiopharmaceutical decays, it emits gamma rays that are detected by a special camera called a gamma camera. The gamma camera is placed over the part of the body being studied to create an image of the distribution of the radiopharmaceutical within the body. [8,9].

Thanks to the gamma camera we obtain 2D projections within different orientations, which then will be reconstructed to form a 3D image (tomographic) to provide the radionuclide spatial distribution information. Several detectors, ranging from one to four, rotate around the patient to obtain the projections within different angles [10,11]. In *Figure 1* the way images are acquired is shown.



**Figure 1.** SPECT acquisition and tomographic reconstructions. Obtention of 1D projection profiles and final reconstructed image.

When the gantry is rotating around the patient, the detected projection profiles at each angle are stored, generating the sinogram. The sinogram is a data matrix in which each row represents an intensity display across an angular view. Every row corresponds to an angular view, obtaining a matrix that maps 1D projection profiles into the 2D sinogram space, forming a sinusoidal path. In *Figure 2*, a scheme to visualize the process of the obtention of the sinogram is observed [11].

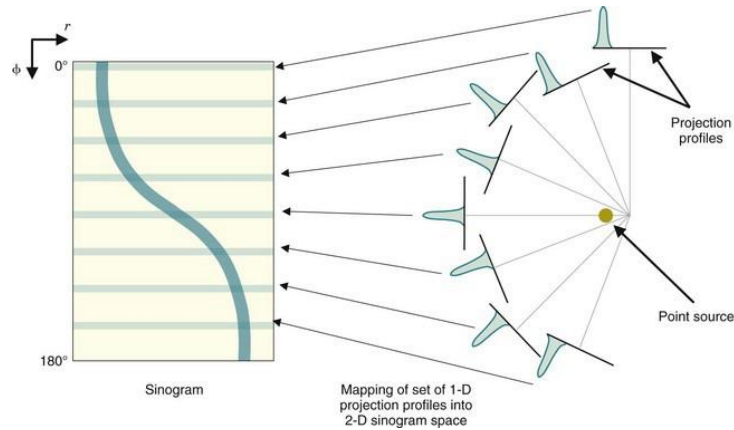


Figure 2. Obtention of the sinogram from 1D projection profiles

Once the sinogram is obtained, to visualize the 3D results, a reconstruction of the image must be done. There are two main reconstruction methods groups to reconstruct SPECT imaging data. It may be done either iteratively (OSEM or MLEM methods) and by FBP technique.

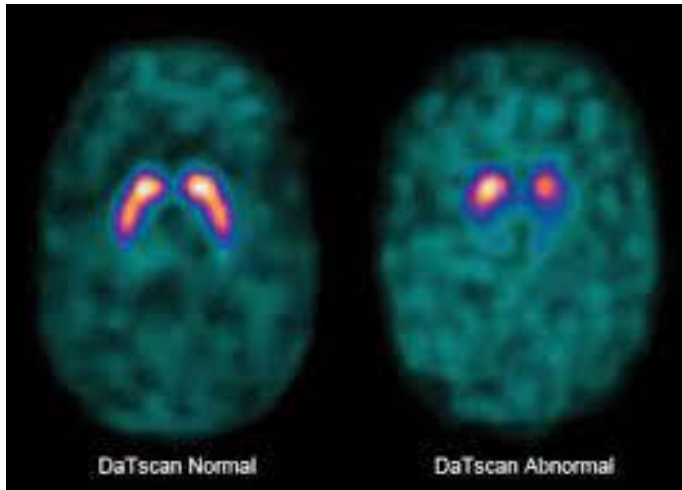
**Filtered Back Projection (FBP)** is a reconstruction analytical method that for its speed, simplicity and computational efficiency is still widely used for clinical SPECT reconstruction. Its basis relies on redistribute all the counts of the projection through the straight line they were generated. This process is repeated along all the pixels and angles. It is usually used with a Butterworth filter, a low pass filter that reduces the amplitude of frequential components above the cut-off frequency [13].

**Ordered-subsets expectation maximization (OSEM)** is one of the most used algorithms in SPECT images reconstruction. It is an iterative reconstruction technique based on MLEM algorithm (Maximum likelihood expectation maximization) [15]. The expression which describes the OSEM algorithm is the one seen in Equation 2, retrieved from Zhu Y.M (2018) [14].

$$\hat{f}_j^{(n,b)} = \frac{\hat{f}_j^{(n,b-1)}}{\sum_{i' \in S_b} H_{i'j}} \sum_{i \in S_b} H_{ij} \frac{p_i}{\sum_k H_{ik} \hat{f}_k^{(n,b-1)} + a_i} \quad (\text{Equation 1})$$

Where  $f$  is the image under reconstruction,  $j$  and  $k$  are the voxel indices,  $n$  is the iteration number,  $b$  is the number of subsets,  $i$  and  $i'$  are the LOR indices,  $S_b$  is the subset  $b$ ,  $H$  is the system matrix,  $p$  the LOR measurement, and  $a$ , the scatter and random correction models.

DaTscan is a radiopharmaceutical used to confirm the diagnostic of PD and dementia in some specific cases. It contains the active substance ioflupane labelled with iodine-123 ( $^{123}\text{I}$ ) which binds to presynaptic dopamine transporters with high affinity. Thanks to SPECT imaging technique we can obtain a 3D image of the striatal dopaminergic system.



**Figure 3.** DATscan imaging technique. Left) DaTscan from a normal patient. Right) DaT scan from a patient with loss of transporters of dopamine.

These images are therefore useful in the detection of the state of nigrostriatal dopaminergic neurons [4]. So, it is used to detect the loss of nerve cells in the striatum, which is a brain region responsible for the movement pathways modulation which are affected in patients with PD. In *Figure 3*, a normal and abnormal DaTscan image are seen, the illuminated zone corresponds to the radioactive distribution in the striatum, it may be seen that in the patient with a disease pattern, this

region is highly reduced, as less dopamine transports are attached to the ioflupane. This technique is highly useful to differentiate between PD and essential tremor and between Lewy body dementia and Alzheimer' disease, as for PD and Lewy body dementia the nigrostriatal uptake is affected.[12].

### 2.1.2 Monte Carlo simulation

MC simulation is a mathematical computational technique of multiple probability used to estimate the probability of outcomes for random variables.

Therefore, MC simulation is a computational technique that is used to model the physical processes taking place in SPECT imaging. It can be used to generate realistic SPECT data where there is no need to obtain physical images and is much more economically viable than phantom SPECT imaging. SPECT data is generated by simulating the processes of the emission and attenuation of gamma rays in the target tissue, also of the collimation and detection of the photons. MC method is very practical because of the stochastic nature of the emission, transport, and detection of radiation [15].

The MC projector that we will use is SimSET, which has been developed at the University of Washington, and is available worldwide. The basic module is the PHG (Photon History Generator), which is the history generator. It is the heart of the simulator and requires the following inputs:

- **Tracer activity distribution**, where we have set activity values, from where the photons will emerge.
- **Attenuation map**, depending on the kind of environment we study, the photons will have more or less attenuation and therefore the photons will arrive with more or less quantity

depending on whether they come from a deeper or shallower area or come from the periphery, this also has to be considered to simulate reality.

- **Data tables** where the equivalence between the structures defined in the maps and the numerical values for the simulation – which will depend on the energy of the photon with which we work – are found. For example, the corresponding attenuation coefficients depending on the energy and the structure.
- **Simulation options** that are specified with four files that contain all the information that we want from the simulation.
  - The binning which contains the information about the projections we want, pixel size, etc.
  - Detector characteristics such as, for example, its energy response.
  - Characteristics of the collimator: type, size of the aperture, etc.

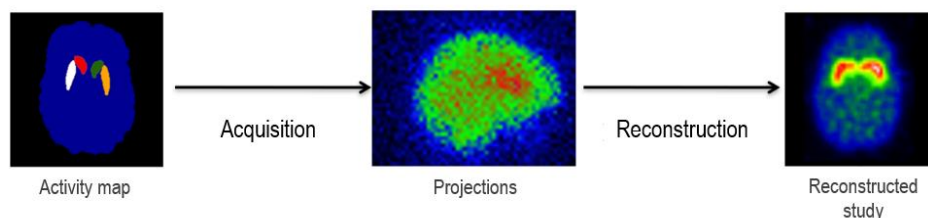
All of this is fixed by the camera we are simulating, and during practice we won't touch it because we'll be simulating the same camera constantly, but we will touch the simulation options because it is related to the scan time, the energy of the tracer we're using, and the number of histories we're simulating, all of which will affect how many photons we'll be detecting at the ending.

The result or output from the simulation is the following:

o A **statistical summary** describing the characteristics of the simulation: observed photons, and simulation variability.

o **Projections** that have been made and are visible in projection format. The distribution that may be seen in the sinogram and projections enables visual evaluation of the quality of our simulation.

In *Figure 4*, a process of a simulate of  $^{123}\text{I}$ -Ioflupane SPECT study with SimSET software is visualised. Then, in *Figure 5*, a summary of the inputs for the SimSET PHG module and the resulting outputs is seen.



**Figure 4.** Process of the simulation of a DaTscan SPECT study.



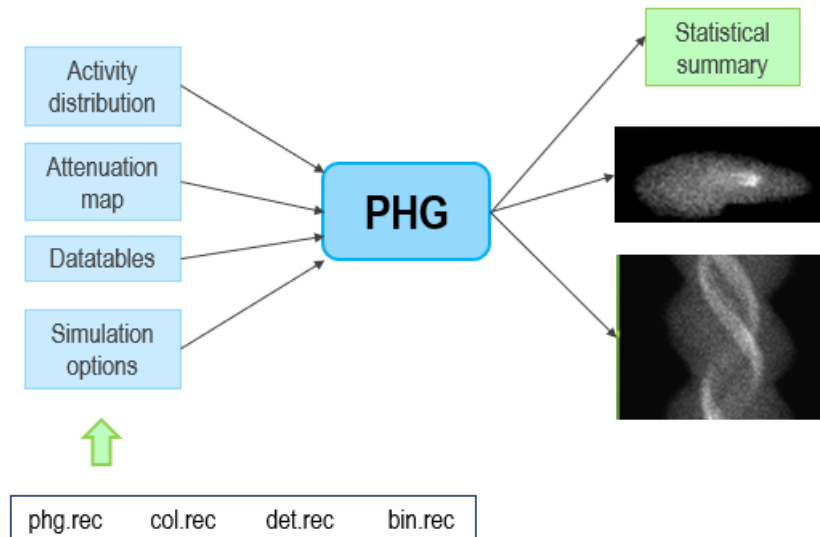


Figure 5. Summary of inputs and outputs of the MC simulation SimSET package (Own source).

## 2.2 State of the situation

Until now, several MC simulation software packages have been developed: SimSET, Geant4 or MNCP (See section 4.1.1). They are PC-based software packages that allow the user to perform simulation flows to obtain SPECT and PET realistic data. However, the drawback is the lack of accessibility for the users that do not have a knowledge background in scripting or particle physics. So, the use of simulations is not that widespread as it could be.

At present, for PET imaging exists a software to generate synthetic brain PET images and datasets: **SimPET**. The project was aimed to validate  $^{18}\text{F}$ -FDG PET scans. SimPET is an open collaboration between University de Santiago de Compostela (USC), *Instituto de Investifación Sanitaria de Santiago de Compostela* (IDIS), Qubitech and *Fundación General de la Universidad de Málaga* (FGUMA) which offers the ability to generate realistic activity and attenuation maps from patients PET/Computed tomography (CT) and Magnetic Resonance Imaging (MRI) images which can be simulated to obtain sinograms and reconstructed images that can be downloaded [16]. It offers the following cited functionalities [17]:

1. *Extract simple activity and attenuation maps from PET and MRI images*
2. *Apply the BrainViset procedure to obtain realistic Activity and Attenuation maps.*
3. *Run Analytic simulations using STIR simulation procedure and MC simulation using SimSET.*

It is an open-source project, so the code may be found on GitHub [17]. It is great alternative to apply simulation flows to generate PET data without having to script the commands to execute MC simulation and reconstruction software packages, as they do not need to be theoretically known. Currently, there is no available software to perform the simulation and reconstruction flow for SPECT like SimPET for PET imaging.

In *Figure 6*, a simple script to call Photon History Generator (PHG) module from SimSET

```
DIR_SIMULACION=/home/raul/practica_simulacio/proyecciones/col_par
DIR_SIMSET=/home/raul/practica_simulacio/simset-2.6.2.5/bin/elscint
DIR_PROGRAMS=/home/raul/practica_simulacio/programes
DIR_OUTPUT_PROJ=/home/raul/practica_simulacio/resultats_par

cd $DIR_SIMULACION
$DIR_SIMSET/phg $DIR_SIMULACION/phg.rec > $DIR_SIMULACION/nohup.simset
```

**Figure 6.** Simple shell scripting code to run a simulation process (GIB-UB).

simulation program and file phg.rec including all the parameters needed for the simulation. The script was provided by the

Grup d'Imatges Biomèdiques de la Universitat de Barcelona (GIB-UB).

Also, scripts to generate the reconstruction images are needed. In *Figure 7*, a simple script to apply a Butterworth filter to projectionssimulation image the projections with a cut frequency of 0.8 and an exponent of 1 is shown. Then, FBP reconstruction is applied with a program called *retro\_fb\_fm\_receptors* developed by GIB-UB in C programming language.

```
WORKDIR=`pwd`
PROJ=$WORKDIR/resultats_fb/rec.weight.img
OUTDIR=$WORKDIR/resultats_fb
DIRPROG=./programes

CUT_FREQ=0.8
EXPONENT=1

REC=$OUTDIR/reconstruccio.img # reconstruction image name

echo butt $CUT_FREQ $EXPONENT > filtre.dat
FILTRE=filtre.dat

$DIRPROG/retro_fb_fm_receptors $PROJ $REC $FILTRE
```

**Figure 7.** Simple shell scripting code to apply a Butterworth and FBP filter to reconstruct input image. (GIB-UB)

As it may be seen, the integration of SPECT simulation studies to validate acquisition, reconstruction or quantification methods is not a trivial task, as previous knowledge on scripting is needed to be able to execute a simulation and reconstruction process.

As for  $^{18}\text{F}$ -FDG PET scans SimPET is born, this project appears from the need of the validation of quantification methods for  $^{123}\text{I}$ -loflupane (DaTscan) SPECT scans. Despite the quantification of DaTscan SPECT images are very useful to establish, there is not an established reference or standard for the dopaminergic imaging [18]. This happens because all the numeric parameters depend on the acquisition and reconstruction parameters, as, for example, it is shown that the reconstruction method used (and their specific parameters like EM-equivalent iterations in OSEM reconstruction) influences the DaTscan quantification [19,20]. Also, the scanner characterisation has an impact on the diagnostic capability for  $^{123}\text{I}$ loflupane imaging [21]. Therefore, the absence of a standardized evaluation procedures for DaTscan SPECT hinders comparisons within individuals and between different subjects. Also, this consequences in challenges when assessing disease progression or evaluating the effectiveness of different drug treatments in PD patients [22].

### 3. Market analysis

#### 3.1 Sectors to which the tool is directed

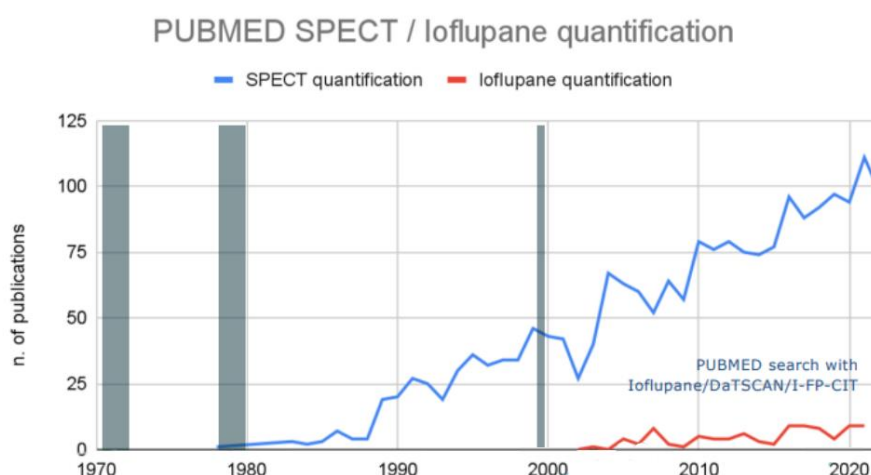
In the context of this project, two main target sectors are found. On the one hand, the main sector to which the tool is destined is **research**, since, as mentioned above, in relation to diagnosing diseases such as PD, quantification plays a very important role. Therefore, the different quantification methods must be validated, and one way to do this is to generate simulations that generate realistic data. Also, it will also be useful in terms of comparing different reconstruction methods before applying them into clinical practice. Finally, it could also be used in hospitals in time of designing acquisition protocols for imaging techniques with SPECT.

On the other hand, although in line with the above-mentioned, the other potential sector at which this project is aimed is **companies** focused on software or imaging medical device development, as this tool could be useful to validate the project they are developing. As said before, the idea of this project is to generate simulations of realistic data (meaning that we can simulate any situation and geometry desired), so it may be used for the project validation purposes.

#### 3.2 Historical evolution of the market

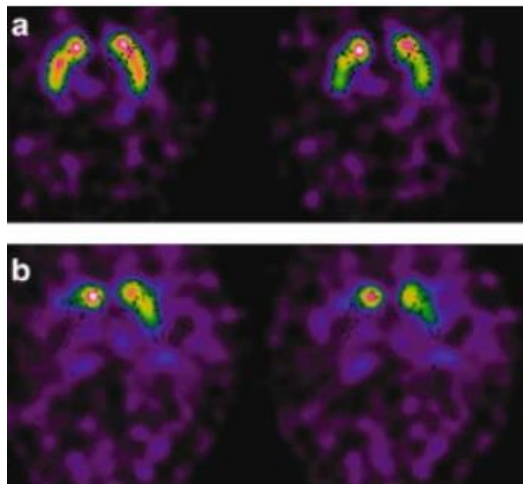
The classical interpretation of tomographic emission imaging has been by healthcare professionals' visual inspection. However, in recent years, an increase in the use and development of quantification methods has been observed. This is key to improve the specificity and reproducibility of diagnosis, as the interpretation does not rely on the observer visual analysis.

An increase among of quantification in SPECT imaging is seen from the beginning of the first methodological approaches, dating back to 1970. Moreover, loflupane quantification searches on PubMed have seen a little increase since the early 2000's until 2020. In *Figure 8*, searches on PubMed of both terms are shown, considering a time period from 1980 to 2020.



**Figure 8.** Graphic illustrating searches in PubMed of SPECT and Ioflupane quantification from 1980 to 2020. [23]

Therefore, a study of the quantification software available has been performed.



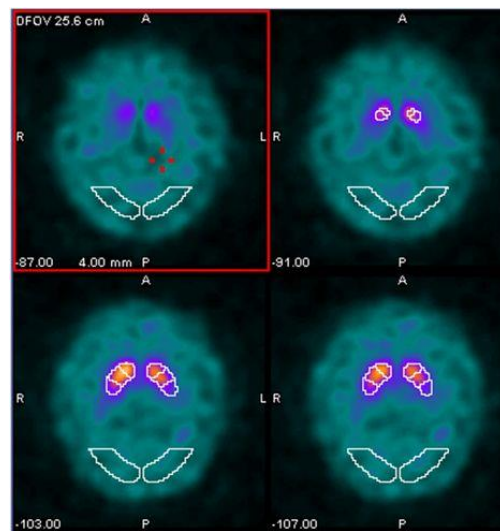
**Figure 9.** BasGan algorithm applied in (a) a control subject and in (b) a PD patient. The VOIs positioned by BasGan software are identified by black dotted lines [24].

### BasGan algorithm

The BasGan algorithm (*Figure 9*) is a basal ganglia matching tools Package for striatal uptake semi-quantification developed in the Department of Physics of the University of INFN, Genoa, in Italy. The algorithm automatically segmentate the striatal  $^{123}\text{I}$ -FP-CIT SPECT based on a striatal template derived from Talairach's atlas. It approximates striatal activity distribution and the radioactive counts [24].

### DaTQUANT

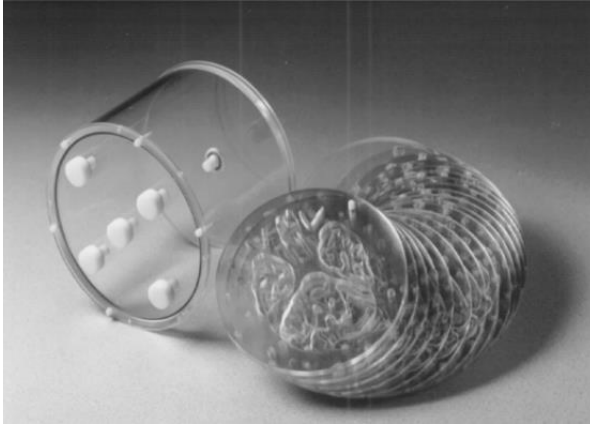
Launched by GE Healthcare in 2013, this software is available to use with any GE Healthcare DaT scan protocol and approved SPECT camera (*Figure 10*). It uses predefined voxels of interest (VOI) to automatically measure volumetrically putamen to caudate uptake ratios. The software also provides longitudinal image analysis letting the monitoring of the disease advancements [25].



**Figure 10.** DaTQUANT save screen. White VOI placement in the images segmenting striatal region. Normal database on top right, marked with red square [25].

However, as aforementioned (Section 2.2: State of the Situation), these methods lack standardisation between them and therefore need validation to demonstrate their reliability.

Nowadays, this validation may be done either with phantoms or simulation techniques like Monte Carlo. Hoffman 3-D brain phantom is one example of what is commercially available:



**Figure 11.** Hoffman 3-D Brain Phantom cylinder and 3-D Brain Insert [26]

## Hoffman 3-D Brain Phantom

Developed by CAPINTEC, this phantom provides an accurate 3D simulation of a radiopharmaceutical distribution found in the brain for SPECT and PET studies. Its main applications are the evaluation of acquisition and reconstruction techniques and quantitative methods [26].

## Striatal brain phantom

The Striatal Phantom (*Figure 12*) quantitatively optimizes PET or SPECT imaging in patients. “This phantom is based upon a standard RSD head with a calvarial cut to insert or remove the brain shell easily”. It is meant to evaluate image quality and validate quantification of the striatal uptake. It comes with a brain shell with five compartments to be filled separately: left and right nucleus caudate, left, and right putamen, and the remainder of the brain. [27].



**Figure 12.** Striatal Phantom [27]

### 3.3. Future market prospects

With the continuous advancement of technology in the field of medicine, research groups focused on functional medical imaging are consequently also evolving along this technological line. Quantification methods are essential in nuclear medicine to assess a quantitative and objective diagnostic. The final product of this project would be a major breakthrough for the validation of the quantification methods proposed by companies or research groups, as they also play a relevant role in the quantitative diagnosis of function SPECT imaging. So, the idea for the final product developed is to be integrated with SimPET program, which would be a useful software to generate simulated and reconstructed PET and SPECT data.

Therefore, it is important to study the potential market that will use this platform (**research** groups and **companies** are the main target considered).



Research groups which have used SimSET simulation package in some studies or projects are the following:

- University of Washington, that apart from being the developers of the program, have been also working in improvements like the implementation of block detectors into the package in *Schmitz, R. E et al. (2007)* [28]
- Department of Biomedical Engineering of the University of California-Devis studying the validation of SimSET to model the Siemens Biograph mCT PET scanner in *Poon, J. K., et al. (2015)* [29] and studying the performance of time-of-flight PET in improving the detection of lesions in *Cao, N. et al. (2010)* [30].
- Department of Biophysics and Bioengineering of the University of Barcelona to study quantification in DaT SPECT in *Cot, A., et al. (2005)* [31]

On the other hand, groups which have used STIR in some of their research lines are:

- Division of Imaging Sciences and Biomedical Engineering of the King's College London in the study of the comparative evaluation of scatter correction in PET for different scatter-levels in *Polycarpou, I., et al. (2011)* [32].
- Department of Physiological Sciences of the University of Barcelona integrating advanced SPECT modelling into STIR framework in *Fuster, B. M., et al. (2013)* [33].
- Institute for Particle Physics and Astrophysics, Department of Physics in Zurich with the implementation of cylindrical PET scanners with block detector geometry in the package in *Khateri, P., et al. (2019)* [34].

Moreover, some companies or research groups that present <sup>123</sup>I-lobupane quantification methods are:

- Siemens offers <sup>123</sup>I Xspect Quant for nuclear motion disorders [35].
- GE Healthcare offers DaTQUANT Software for PCS to quantify DaTscan images [36].
- Qubiotech offers commercially an integrated version QuantiDOPA, a Quantification Software for Dopaminergic Neurotransmission SPECT [37]

## 4. Concept engineering

The idea of this final degree project is to develop a program to simulate and reconstruct realistic SPECT data from three main files; a parameters file to edit the simulation options, an activity map, and an attenuation map. So, the main purpose is to select the correct tools to be able to develop the final project.

First, a solution study will be performed, followed by the final tools' choice to carry out the project and the justification for the election. Also, an alternative solution has been considered in case the chosen solution is not possible to be performed in case of lack of time.

### 4.1 Solutions study

In this section, research is carried out to study the possible solutions to take this project forward. First, by assessing the various packages used to perform Monte Carlo simulations, then also studying the different reconstruction packages. Then, a consideration of the different programming languages to develop the main program and graphical interface is done

#### 4.1.1 Monte Carlo simulation toolkits

Several packages and tools are available to perform MC simulations. All of them are free software packages which may be used for simulating emission tomography.

The Geant4 Application for Tomographic Emission, **GATE**, is a simulation toolkit developed by OpenGATE that includes Geant4 libraries to provide a modular, versatile, and scripted simulation adapted to the field of nuclear medicine. It allows to model time-dependent phenomena like detectors movements to achieve more realistic projections data [38]. Related to this package, **Geant4** is another toolkit to simulate the particles passed through matter. It was developed by a collaboration led by CERN (the European Organization for Nuclear Research), and it is used in physics fields including nuclear medicine or astrophysics.

**SimSET** is a popular simulation toolkit developed by the University of Washington Imaging Laboratory which models physical processes and instrumentation that are used in emission imaging [39]. It presents a great computational efficiency as it features fast simulation time and variance reduction tools, and it is on constant development by the group by adding new functionality and utilities [29]. However, it lacks on the modelled detector geometry, as for example, the detector ring cannot be subdivided into individual crystals, also the distance between them cannot be modelled.

The Monte Carlo N-particle Transport, **MCNP**, is a MC radiation transport code that is designed to track particles over ranges of energies developed by Los Alamos National Laboratory. It simulates particles interactions in various materials and offers reduction techniques to make the simulations more efficient [40]. However, MCNP has a lack of flexibility in specifying the geometry in the code [41].



#### 4.1.2 Reconstruction toolkits

Software other than STIR have not been considered, as it is already prepared to work in different computing environments. Other package like STIR has not been found, only reconstruction software platforms like

**STIR** is an open-source software used for the reconstruction in tomographic imaging, PET, and SPECT, being the successor of the PARAPET software library. *STIR is currently maintained by Kris Thielemans (University College London), with co-maintainers Nikos Efthimou (University of Pennsylvania) and Charalampos Tsoumpas (University of Leeds) [42]*

#### 4.1.3 Programming language to develop the main program

Two of the most used programming languages for software development are **Java** and **Python**, the two of them are object-oriented languages and count with extensive libraries, also lot of documentation is found on the internet for both options.

**Python** is a dynamically typed interpreted language, so variables' data type does not need to be declared, as it was thought to be easily readable. However, **Java** is a statically typed and compiled language, so the variables need to be explicitly declared, making this last language faster at runtime and easier to debug due to its previous compilation. Python tests the type of variable during the runtime, so it is not as stable as Java. On the one hand, **Java** main applications are web applications, desktop complex GUI apps or enterprise solutions, on the other, **Python** finds its main public in the scientific community for scientific and numeric computing, machine learning and artificial intelligence or image processing [43]

#### 4.1.4 Programming language to develop the graphical interface

In case of choosing to work with Python as the programming language, many libraries destined to develop graphical user interfaces (GUI) are available. After doing some research, three of them have been selected to analyse them and finally choose one. All of them are open-source packages, so they may be used for free by anyone.

**Tkinter** is the Python's default library to develop GUIs and is available for Unix and Windows, as it is maintained by ActiveState. It is based on the Tcl/Tk library, and it consists in different modules with different functionalities; Tcl, which is an interpreted and dynamic script programming language, and it is used to generate widgets and windows in Tk. Then, Tk, that is the library used to generate buttons and other widgets, and finally Ttk, which is a Tk extension that provides widgets with a more modern appearance [44].

**PyQt** is another Python libraries to create multiplatform application either for computers or mobile phones. It is based on Qt platform (a set of C++ libraries and development tools for the development of GUIs) and is dual licensed: free and fee-paying. It is supported by all platforms





and has a lot of components to make a GUI fully functional. However, the free license offers limited functions [45, 46]

**WxPython** is an open-source cross-platform tool to develop GUIs for Python. It is a library that include python modules to wrap GUI components of the wxWidgets platform library (written in C++). With it, rich and interactive graphical interfaces may be developed [47].

#### 4.2 Proposed solution

The main goal of the project is to develop a software to simulate and reconstruct SPECT data with the Monte Carlo computational technique, and it is important to make the proper choices to facilitate the project and ensure its functionality. The best option, as the idea is to develop a software would be to use Java, as it is its main function. But, based on personal experience, the chose programming language to develop the main program and the graphical interface will be **Python**. This decision is made because Python also offers the development of functional graphical interfaces and for the programming experience in this language. Also, SimPET main program is developed in Python, so having a basis could be very useful to fulfil the objectives of the project more precisely and not spend much time learning a programming language from the beginning, like it would be if the decision was to work with Java.

Regarding with the simulation software package, the decision is also made based on the experience and recommendation from the group GIB-UB. **SimSET** is the toolkit that will be used. The package was learned at university where we performed a simulation of DaTscan SPECT data using this program, so its structure and main programs are already known. It has four modules, the Photon History Generator (PHG) and the collimator, detector and binning modules, which will be explained in the following section. Also, as **STIR** is already prepared to run with SimSET, and it was the package used by the SimPET developers, it will be the one used for this project. Moreover, a lot of documentation and examples are found online. This last thing is very beneficial as the construction of the files needed for both software packages will be easier and more optimal.

For the GUI development, all solutions mentioned are equally good. However, **tkinter** library will be the option finally chose, as it is one of the most used tools to develop interfaces, so a lot of documentation and examples of code is available online, facilitating the work. Also, comparing to the other options, after some research, it has been concluded that is the more intuitive GUI library to program and the functionality of the objects is the same.

#### 4.3 Alternative solution

An alternative solution will be proposed in case there is not time to finish the reconstruction program using STIR software, as SimSET is already known, but the reconstruction package is not familiar, so it is possible there is not enough time to complete the task using it.

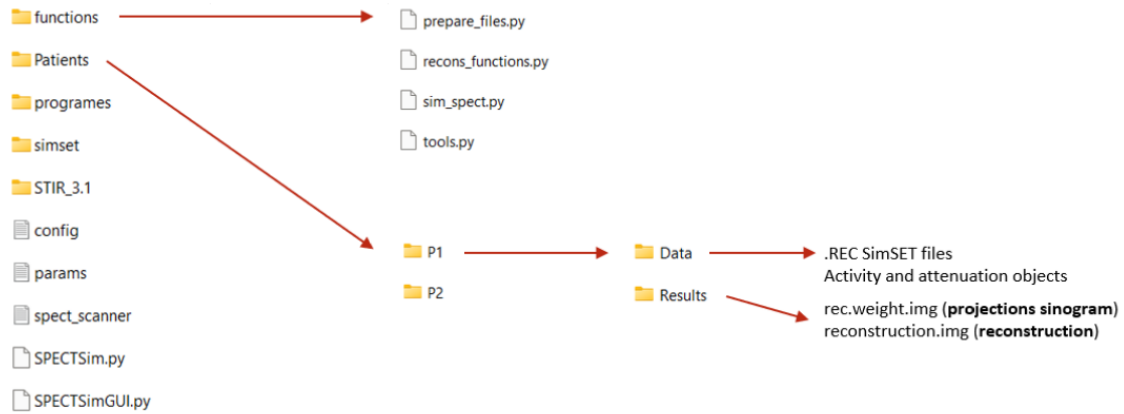


Therefore, in the proposed alternative solution the first part of the program would also use Python as programming language and the SimSET software package for the simulation program. What would change would be that instead of using STIR as the reconstruction package, only a filtered back projection filter (FBP) would be applied, since the group in which the project is being developed, GIB-UB, has a program written in C++ to apply the filter, therefore, this could be executed from Python or transcribe the code to this programming language.

## 5. Detailed engineering

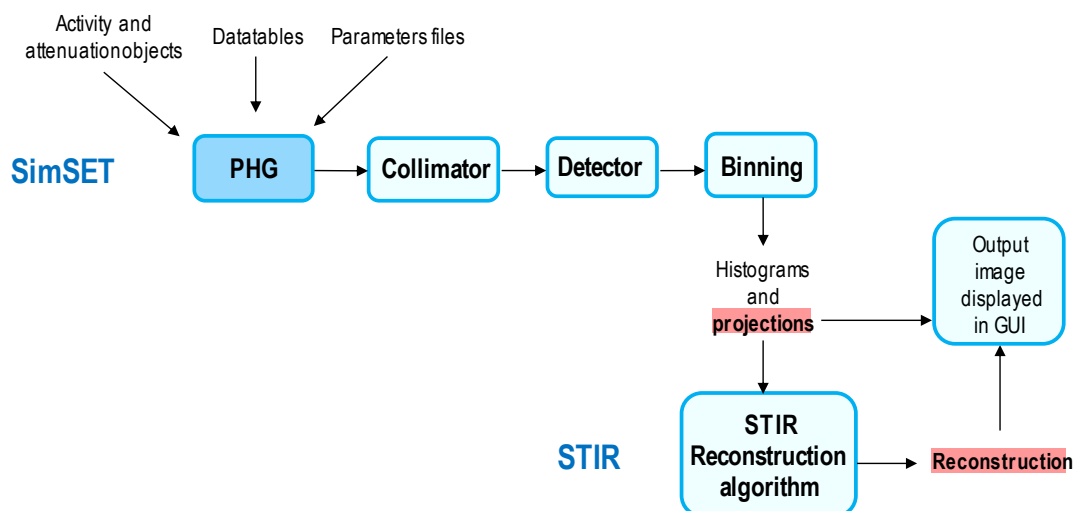
### 5.1 Workflow of simulation and reconstruction process

Before going into the workflow of the simulation and reconstruction a brief explanation of how the scripts and data are organised will be given. The scheme may be observed in *Figure 13*.



*Figure 13.* Data and scripts organisation in the working directory (Own source).

As it is seen, the data is divided in three groups, **input parameters**, python **scripts** of the program, and **outputs**. The input parameters are the three “config”, “params” and “spect\_scanner” yaml files, which contain the main parameters to perform the simulation. “params” file includes the runtime options for the simulation (number of photons to simulate, scan duration, main directory paths, etc.), “config” file includes parameters for the PHG module that should not be changed, and “spect\_scanner” file that has the geometry traits of the scanner, including the collimator and detector specifications, and the binning parameters (pixel and matrix size, etc).



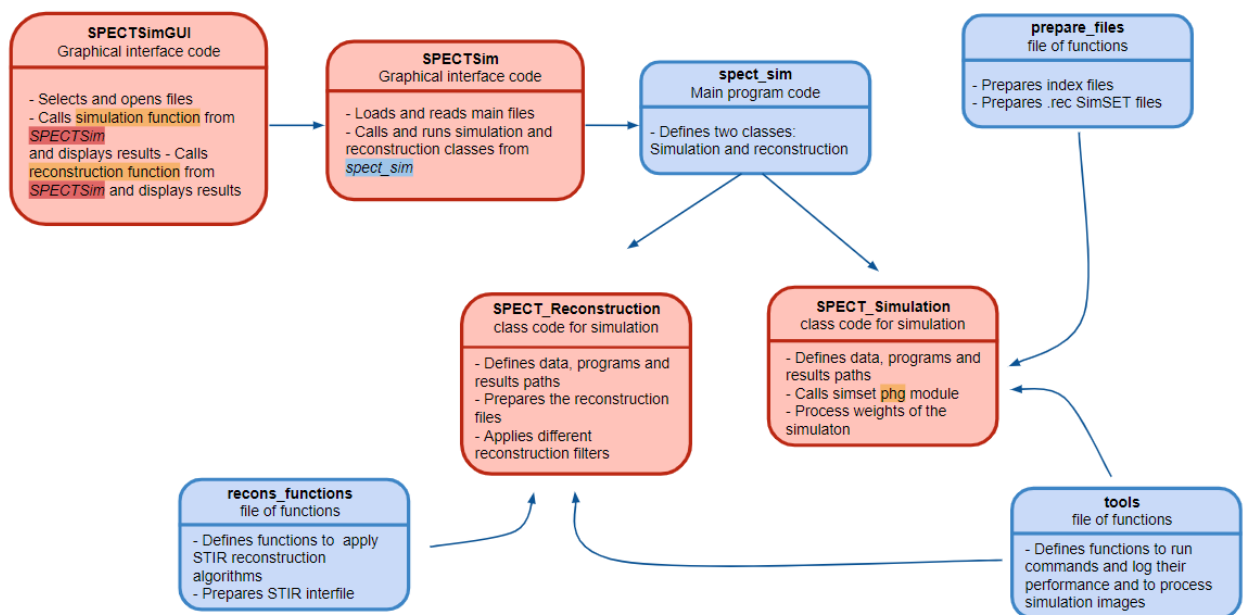
*Figure 14.* Workflow of the simulation and reconstruction programs using SimSET and STIR packages (Own source).

The workflow of the simulation and reconstruction is the one seen in *Figure 14*. The core module of the SimSET simulation program (used for the simulation program) is the **Photon History Generator** (PHG), which generates a decay and the resulting photons and their tracks through the input activity map. It also counts with the **Collimator Module**, which receives the photons from the PHG module and tracks them when simulating SPECT data, the **Detector Module**, that tracks the photons through the simulated detector, and finally, the **Binning Module**, that histograms the detected photons and provides an output array that is user-formatted (See Annexes 1 for more detailed information regarding the SimSET package).

On the other hand, a reconstruction algorithm of STIR is applied to the projections. The flow will be explained in detail in section 5.2.2, but in general terms, to reconstruct with STIR a generation of two files must be done: an interfile, and a file which calls the reconstruction algorithm – which collects the parameters specified in the interfile – and defines other variables specific to the chosen algorithm (See Annexes 2 for more detailed information regarding the STIR package).

## 5.2 Python code for the back end (main program)

To develop the back end, different files have been programmed. Two types of files have been generated; one type meant to be as a file with auxiliary functions needed to perform some processes in the program flow. On the other hand, the second type of python file created is meant to be part from the main program flow. There, classes to create object to group data and functionality are generated. The three main classes found in the back end of this project are: **SPECTSim**, **SPECT\_Simulation** and **SPECT\_Reconstruction**, the three of them will be explained in detail in the next subsections.



**Figure 15.** UML diagram showing the different classes and files (main and auxiliary) to accomplish SPECTSim performance (Own source).

A Unified Model Language (UML) diagram has been carried out to visualize the main blocks of the program and their main functionality. It may be seen in *Figure 15*. Red boxes correspond to classes, blue boxes correspond to files and the orange underlined refer to file-specific functions or programs.

In general lines, the main flow of the program works this way: The **SPECTSimGUI** class is called, and it generates the graphical interface window, which has several buttons to call the different classes and upload the different files, choose parameters and display both simulation and reconstruction results (See 5.3 section). It imports the **SPECTSim** class, that loads and reads the main files selected and calls the *spect\_sim* file, where both simulation (SPECT\_Simulation) and reconstruction classes (SPECT\_Reconstruction) are written. These two classes contain the program to perform the simulation and reconstruction, respectively, and both call the auxiliary files *tools* to develop some tasks for the simulation performance. Then, simulation class calls *prepare\_files* file for simulation program tasks like the creation of specific SimSET files and reconstruction class uses *recons\_functions* file to perform specific tasks for the reconstruction program.

Summarizing, SPECTSim includes four classes and three auxiliary files that define the skeleton of the program, with respect to the four **classes**, they are:

- *class* SPECTSimGUI: Definition of the graphical interface.
- *class* SPECTSim: Main file, it loads yaml files and defines two functions; simulation and reconstruction, which call simulation and reconstruction classes, respectively.
- *class* SPECT\_Simulation of *file spect\_sim*: Defines the main code for the simulation calling PHG module from SimSET and a run function to execute the process.
- *class* SPECT\_Reconstruction of *file spect\_sim*: Defines the main code for the reconstruction calling reconstruction file and a run function to execute the process.

Then, the **auxiliary files** include a set of functions helpful for the code. These are made to organise the code and make it clear. The files are:

- *file sim\_spect.py*: contains the main program code, as it is where simulation and reconstruction classes are defined.
- *file prepare\_files.py*: Defines functions to write .rec SimSET files and index files, necessary for PHG module.
- *file tools.py*: Defines necessary functions to load images and run commands (Developed by SimPET authors).
- *file recons\_functions.py*: Defines functions that include the main reconstruction algorithms and filters from STIR software package. (Adaptation from *stir\_tools* file developed by SimPET authors).

Depending on the use the user wants to give to the program, the SPECTSimGUI class may be excluded. If the user wants to perform the simulation and visualize the results on the graphical interface, they need to call the GUI class. But, if the user desires to edit parameters or lines of code they must generate a file to call the *spect\_sim* file and run it on terminal. The rest of the program works the same for both options. (See Annexes 3 *Installation and User Guide*).

In the next sub-sections, all files and classes will be explained in more detail. The code is mainly based mainly on available shell scripts files to perform the simulation written in bash, so the migration from this language to python will be explained. In all classes, an `__init__` function that defines the main variables, objects and files used in the functions must be written.

### 5.2.1 Definition of the program

#### SPECTSim class

The SPECTSim file defines a class called also SPECTsim where an `__init__` function is defined. There, the three parameter files with *yaml* extension are uploaded to save their data into a dictionary for each one. Having a dictionary lets us access the data in an easy way, with the method “*get*”. The upload of *yaml* files is done with *safe\_load* function from *PyYaml* library. Also, in the initializing function, main paths are also defined, these are:

- *path spect\_dir*: Working directory.
- *path patients\_dir*: Path to patients' folder.
- *path simulation\_dir*: Path to patient data folder, corresponds to “Data” folder inside the selected patient directory. It contains the necessary data to perform the simulation.
- *path results\_dir*: Path to patient results folder, corresponds to “Results” folder inside the selected patient directory. It is where the resulting images are stored.

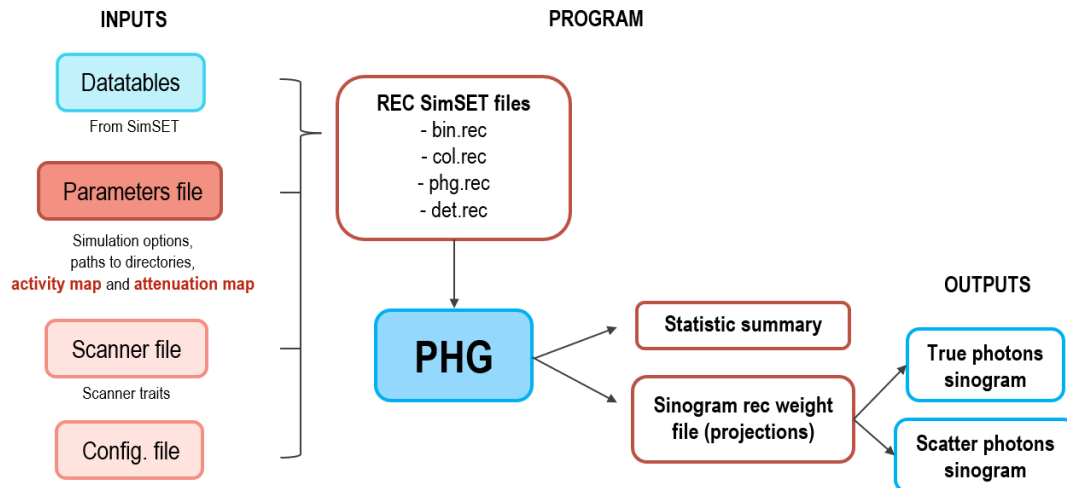
### 5.2.2 Main program

The main program is defined in *sim\_spect* file, it is where the simulation and reconstruction classes are found. In both functions, several libraries are used to make it functional, the most important are:

- *library os*: It allows us to define paths to create folders, files and run commands own of bash like *cp* (copy), *rm* (remove), etc.
- *library multiprocessing*, specifically, the *function* *Process* to run defined functions

#### SPECT simulation class

Simulation class is contained in *sim\_spect* file and defines the program to call the SimSET PHG module. The flow may be seen in *Figure 16*.



**Figure 16.** Block diagram showing the flux of the simulation program. The inputs, the program flow and the outputs (Own source).

The idea of the program is that several inputs are provided: set of **data tables** from SimSET, a **parameters** file that contains the simulation runtime options, the paths to the main directories, the activity map and the attenuation map, a **scanner** file that has the scanner traits (collimator, detector, and binning parameters), and a **configuration** file that also includes PHG module parameters that are not expected to be changed.

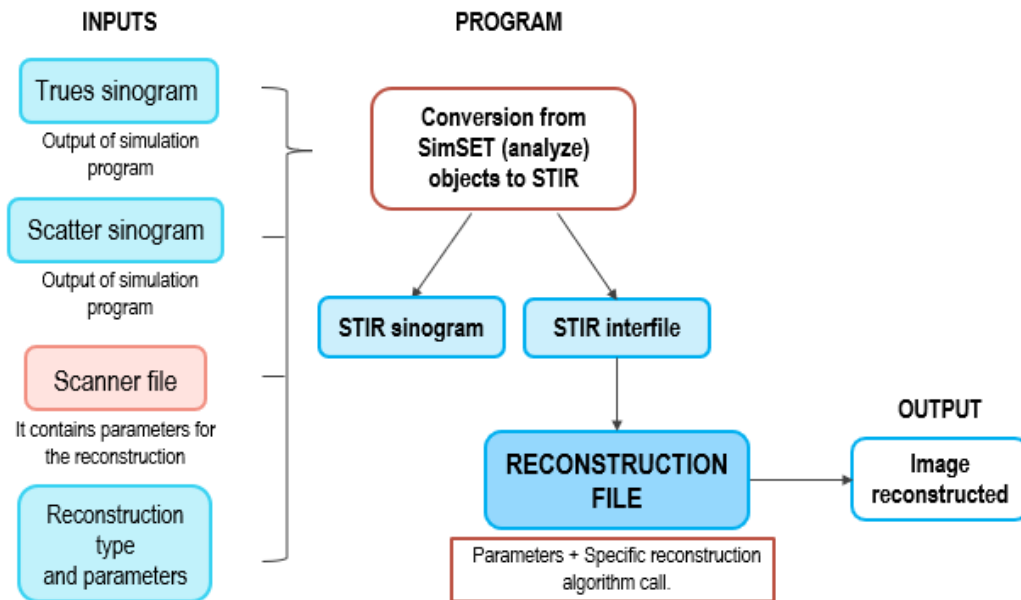
With the inputs, rec SimSET files that are constructed and introduced into the PHG module. When the simulation is over, the outputs are a statistic summary and an image with the projections. The image with the projections is in analyze format, so it is composed by a header (.hdr) and the image file itself (.img). This image is finally divided in true photons sinogram and scatter photons sinogram.

To do that, three functions (apart from the `__init__` one) are defined:

- *function* run: Uses the tool “Process” from multiprocessing library to run `simset_simulation` function.
- *function* `simset_simulation`: Calls PHG module from SimSET. To adapt the bash command to python language, the `system` function from `os` library is used (See 5.2.3.2 for information about `tools.osrun`). This function also divides the final weight image in trues and scatter sinogram objects.
- *function* `prepare_simset_files`: Calls the `prepare_files` file to write the rec files (`phg.rec`, `col.rec`, `det.rec`, and `bin.rec`) and index file, necessary to run the SimSET PHG module.

## SPECT Reconstruction class

Reconstruction class is also contained in *sim\_spect* file and defines the program to generate the STIR sinograms that later will be reconstructed with an algorithm program from the package. The flow may be seen in *Figure 17*.



**Figure 17.** Block diagram showing the flux of reconstruction program. The input, the process, and the outputs. Also the reconstruction algorithms considered for the program (Own source)

As it is seen in the block diagram, several inputs are provided. Coming from the other program the trues and scatter **sinograms** are found, also, the **scanner** file (as it specifies binning data and parameters useful for the reconstruction with STIR package) is introduced. On the other hand, other inputs like the reconstruction algorithm chosen and its parameters are found.

The reconstruction algorithms and their specific parameters considered are **FBP2D** and **FBP3D**, the parameters of which are: cut off for ramp filter, and **OSEM2D** and **OSEM3D**, where the variables are: number of subsets and number of iterations

With the inputs, a conversion of the sinograms from analyze to STIR format is done, also generating an interfile with the necessary information for the reconstruction. Some data described in the interfile is: the type of data (tomographic), the number of projections (120), the matrix size ([128,32]) or the start angle (0°). Having the interfile, then a reconstruction algorithm parameter file is written (See Annexes 2: STIR to see examples of both interfile and algorithm parameter files).

To do that, three functions (apart from the `__init__` one) are defined:





- *function* run: Uses the tool “Process” from multiprocessing library to run *reconstruction* and *prepare\_recons* function.
- *function* prepare\_recons: This function converts all SimSET data to STIR data and generates the interfile
- *function* reconstruction: Depending on the specified reconstruction algorithm, functions from *recons\_functions* file are called, and reconstruction parameter file is written, and the reconstruction is performed.

### 5.2.3 Auxiliary function files

The auxiliary functions are essential for the program, as they define useful functions that are called to perform some tasks during the simulation or reconstruction. So, they are essential for the program to work correctly.

#### Prepare\_files functions

This file of functions is used to generate the files SimSET needs to work. So, it consists of several functions to write the index file and the .rec files.

- *function* prepare\_simset\_det: Writes in a .rec extension file the detector module parameters from the variables of the yaml scanner file.
- *function* prepare\_simset\_bin: Writes in a .rec extension file the binning module parameters from the variables of the yaml scanner file.
- *function* prepare\_simset\_col: Writes in a .rec extension file the collimator module parameters from the variables of the yaml scanner file.
- *function* prepare\_simset\_phg: Writes in a .rec extension file the photon history generator module parameters from the variables of the yaml params and config files.
- *function* process\_weights: Divides the final weight image in trues and scatter sinograms.

#### Tools file main functions

This file was developed by SimPET authors [48], but some functions have been adapted or removed according to this project. The main functionalities of the defined functions in the file are the following:

- Function to run commands with os library and generate a logging file to visualise the performance.
- Functions to do conversions between image formats (i.e., Nifti1 to Analyze).
- Functions to operate between images (i.e., sum two sinograms).
- Function to generate analyze images from data.

## Recons\_stir file main functions

This file includes functions used for the reconstruction class which write the algorithm files needed by STIR. They are an adaption from SimPET functions for PET (*stir\_tools* file [49]), changing the specifications and parameters so they work for SPECT data.

The four functions to call STIR algorithms are the following:

- *function* FBP2D: Creates a file to introduce it to the STIR FBP2D program.
- *function* FBP3D: Creates a file to introduce it to the STIR FBP3D program.
- *function* OSEM2D: Creates a file to introduce it to the STIR OSEM2D program.
- *function* OSEM3D: Creates a file to introduce it to the STIR OSEM3D program.

Also, a function to generate the interfile needed by STIR was defined.

### 5.3 Python code for the front end (GUI development)

Tkinter (tk) was the python library selected to develop the GUI for the simulation and reconstruction program because of its great range of components and functions to make an equipped GUI and the great quantity of documentation and code examples available online. The widgets used to develop SPECTSim GUI are the following:

- *tk.Tk()*: To create main windows.
- *tk.Button()*: To open a popup window or execute an action.
- *tk.Label()*: To display text in a window.
- *tk.filedialog.askopenfilename()*: To select files from the explorer.
- *tk.Toplevel()*: To create popup windows.
- *tk.Scale()*: To create a slider.
- *tk.ttk.Combobox()*: To create a list of options.
- *tk.entry()*: To let the user introduce data.

The GUI consists of a main window and two popup windows to display results from simulation and reconstruction separately. The flux diagram showing the widgets (blue boxes) and windows (red boxes) generated of the GUI are observed in *Figure 18*.

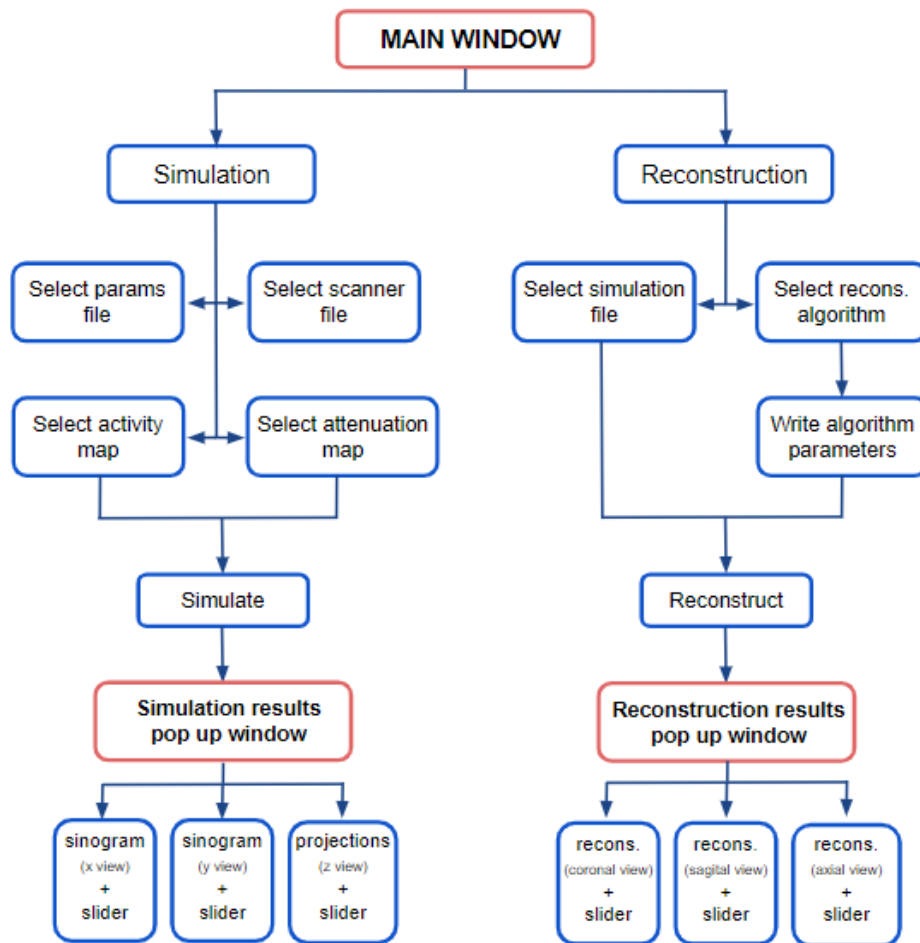


Figure 18. Widgets and buttons diagram of SPECTSim graphical interface (Own source).

### Main window

The main window of the GUI includes two buttons to select the process to perform either simulation or reconstruction process.

When “simulation” button is clicked, four buttons appear to select four files: Params file, scanner file, activity map and attenuation map. When all four files are selected, the “simulate” button should be clicked. This calls the simulation function from *sim\_spect* file, so the simulation process starts for the introduced input data. It also opens the simulation popup window where the results are displayed showing the x and y views from the sinogram and the projections (view z). Each image has a slider to choose the slide of the view displayed, as the result is a 3D image with a determined shape (i.e. [128, 128, 32]). (See section 5.4: Results)

Then, when “reconstruction” button is clicked, two buttons appear; one to select the simulation file previously generated and another one that is a menu bar to choose the reconstruction algorithm to apply. When the algorithm is chosen, text entries to introduce the parameters for the filter

appear. When all this is defined, the “reconstruct” button should be clicked. This calls the reconstruction function from *sim\_spect* file, so the reconstruction process starts for the introduced input data. It also opens the reconstruction popup window where the results are displayed showing the coronal (x), sagittal (y) and axial (z) views. As before, each image has a slider to choose the slide of the view displayed. (See section 5.4: Results)

Depending on the reconstruction algorithm chosen, input labels to write the desired parameters will appear, the considered ones are the following:

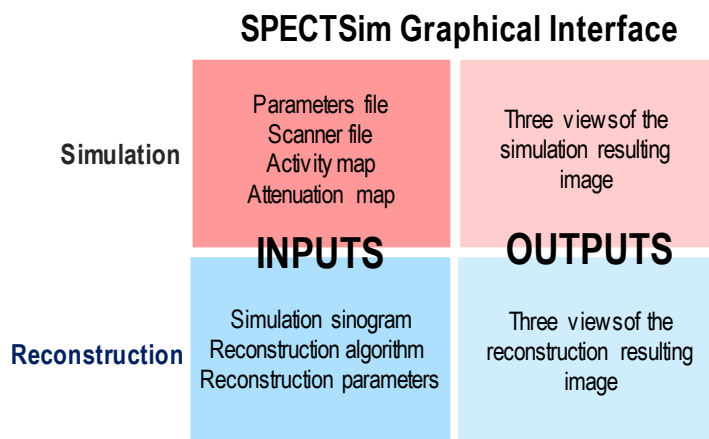
For **FBP2D** and **FBP3D**:

- Cut off for ramp filter

For **OSEM2D** and **OSEM3D**:

- Number of subsets
- Number of iterations

In summary, the inputs and outputs required by the GUI to function correctly are the ones collected in *Figure 19*.



*Figure 19.* Scheme showing the inputs and outputs needed for both simulation and reconstruction flows (Own source).

## 5.4 Results

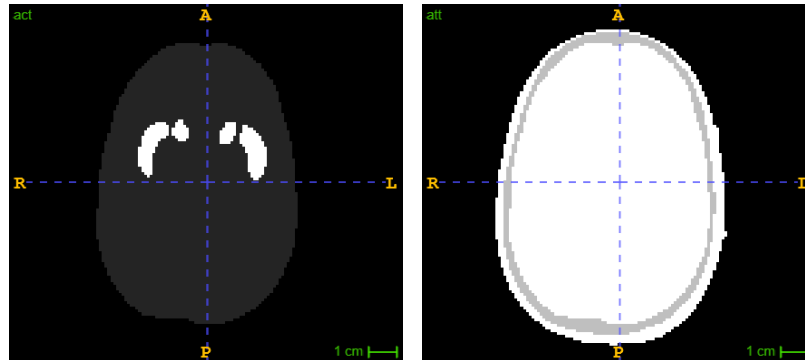
To visualize the final result of the project, an example has been used.

### Input data

As aforementioned (Section 5.1), the program needs several input data to accomplish its functionality, the main inputs are: a parameters file including runtime options for PHG module, an activity distribution map, an attenuation map and an scanner file including the collimator, detector, binning, and reconstruction information.

Then, the program also needs other reconstruction specifications: the reconstruction algorithm and specific parameters for the chosen algorithm.

The activity and attenuation maps used are the ones seen in *Figure 20* (Seen in axial view)

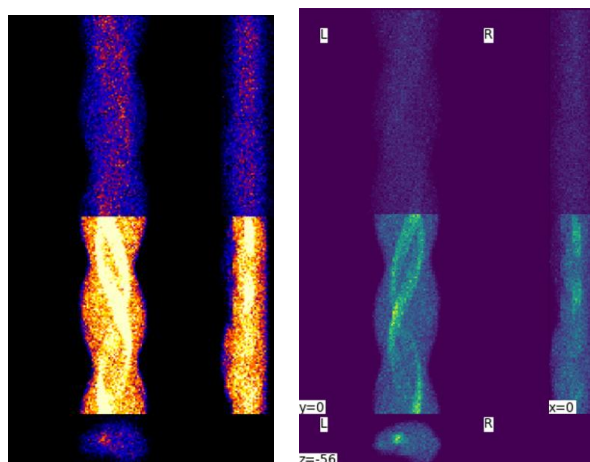


**Figure 20.** Activity distribution map showing striatal nuclei (left) and attenuation map (right) of striatal phantom in axial view. (Source: ITK-SNAP)

Other parameters of interest regarding the example used to show the results are the following. All the other parameters may be seen in Annexes 2, in the configuration of the four module file sections.

- Number of photons: 100.000.000 photons
- Scan duration: 100 seconds
- Shape of the image: Number of views: 128; Number of bins: 240; Number of slices: 32
- Hole radius of the collimator: 0.0615
- Photon energy: 140 keV
- Hole geometry of the collimator: Parallel

## Results



**Figure 21.** Simulation results visualised with (left) MRICro and (right) Nilearn library from Python (Own source)

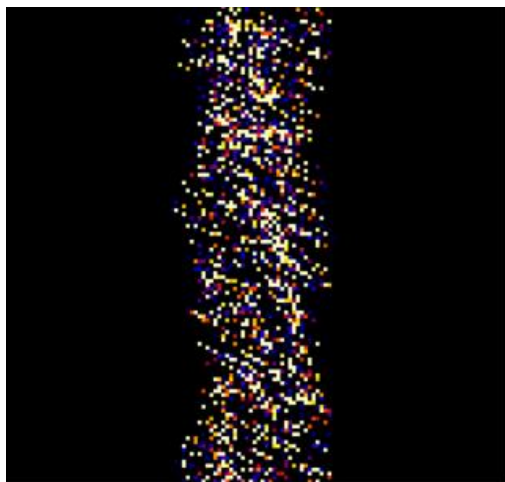
The images above correspond to the view of the sinogram on the x (right) and y (left) axes and below we see the projections. In addition, in the sinogram we see two differentiated parts, in the upper part we see the result of a common degrading effect in nuclear medicine: the photons that have undergone a dispersion. In the lower part we see the real photons.

The statistical summary for different number of photons to simulate considered is collected in *Table 1*. Results studied were the photons passing the collimator-detector-binning module and the total accepted photons. All simulations were executed for the same activity and attenuation objects, also the other parameters apart from the number of photons were the same.

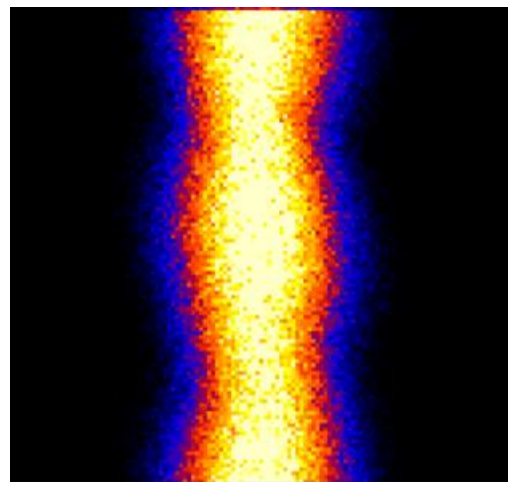
*Table 1. Statistical summary of simulation for different numbers of simulated photons*

Photons simulated	Total counts
5.000.000	1,271747e+06
15.000.000	1.279572e+06
100.000.000	1.897213e+06

Finally, in the following images, the coronal view of the simulated sinogram for each different value of simulated photons considered is seen.



**Figure 22.** Coronal (y) view of sinogram for 5.000.000 photons simulated. (Source: MRICro).



**Figure 23.** Coronal (y) view of sinogram for 15.000.000 photons simulated. (Source: MRICro).

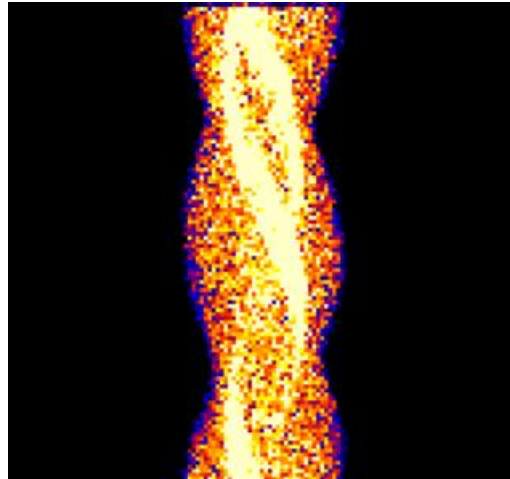


Figure 24. Coronal (y) view of sinogram for 100.000.000 photons simulated. (Source: MRlcro).

As it may be observed, the signal to noise ratio improves, as for less photons generated the image quality is inferior. Also, when the number of photons is increased (See Figure 24), the striatum activity is appreciated in the middle of the sinogram.

On the other hand, examples of FBP3D (Figure 25) and OSEM3D (Figure 26) reconstructions are seen.

For **FBP**: Cut off for ramp filter = 0.5 for a simulation.

For **OSEM**: Number of subsets = 8; Number of iterations = 10 for a simulation.

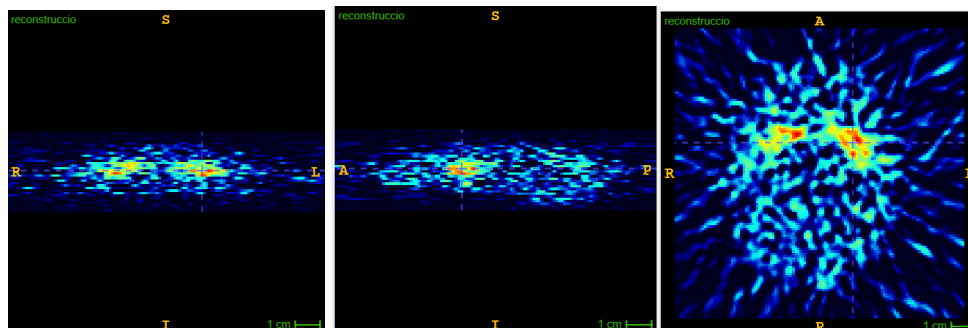


Figure 25. Reconstruction with FBP algorithm. CSA order plotted in jet colour map. The colour map has been altered to distinguish better the brain regions with the background. Zoom of 1.5 applied. ((Source: ITK-SNAP)

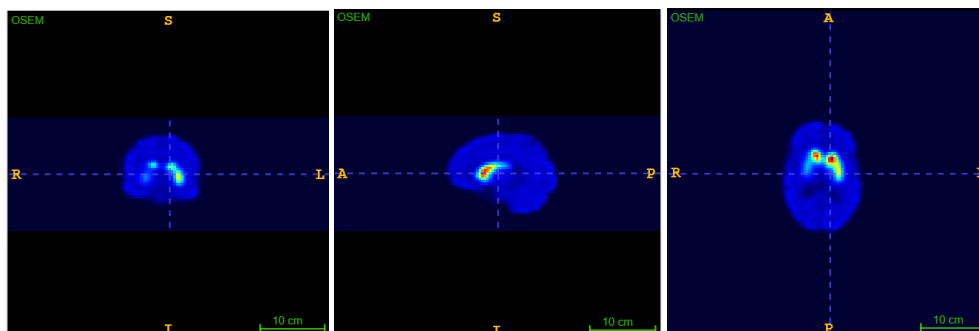
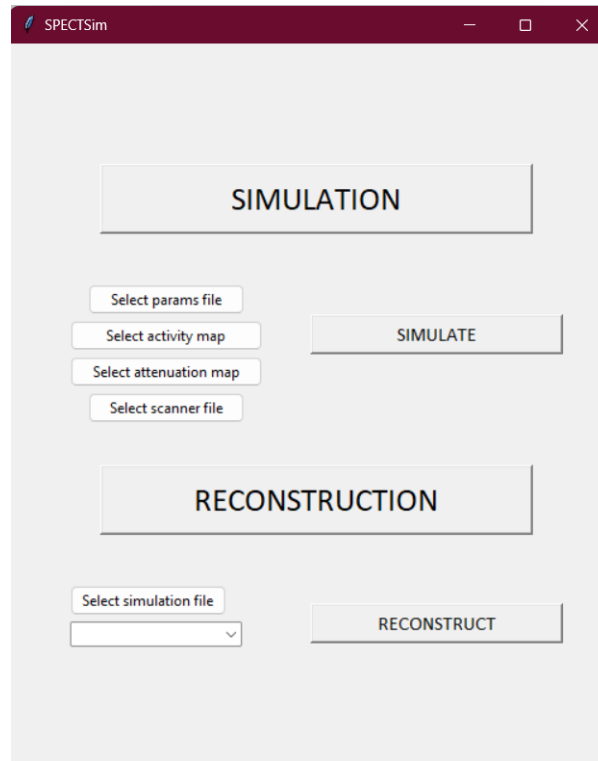


Figure 26. Reconstruction with OSEM algorithm. CSA orde, plotted in jet color map r. (Source: ITK-SNAP)

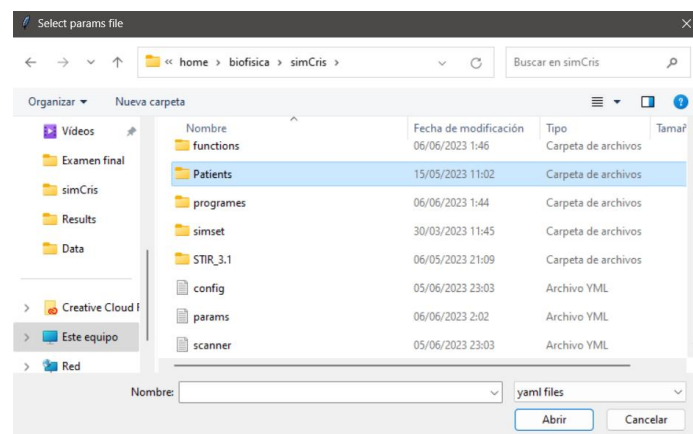
## Results in graphical interface

In *Figure 27*, the main window of the GUI is shown. As it may be seen, all the buttons specified in previous section appear. In the top, there are the buttons to control the simulation flow, and in the bottom the buttons to control the reconstruction flow appear.



*Figure 27.* Main window of GUI (Own source).

When the “select” buttons are clicked, the file explorer of the computer is opened, letting the user choose the specified file (*Figure 28*).



*Figure 28.* File explorer window opened when the select buttons are clicked (Own source).

Then, in *Figure 29*, the simulation pop-up window showing the results of the Monte Carlo simulation for the input parameters is seen. Top left, the y view from the sinogram is seen,



observing both the real and scatter sinograms, top right, the x view from the sinogram is visualised. Left bottom, the projections (z view) are seen. At each side of the three images, the slider to control the slice shown is also seen.

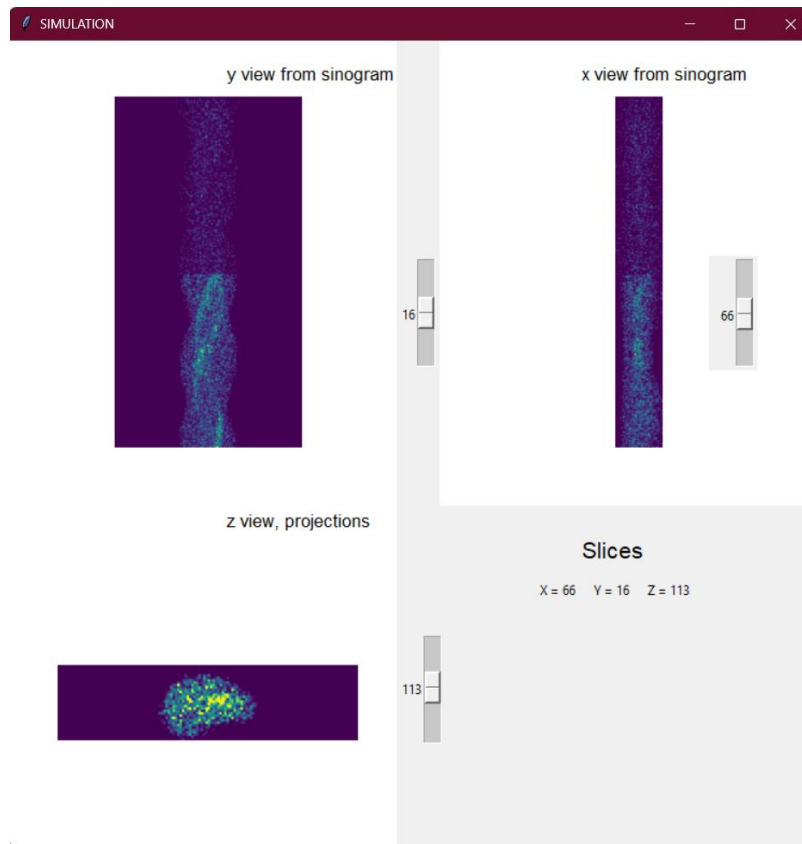


Figure 29. Simulation pop-up window (Own source).

Then, when the "Reconstruction" option is chosen, and the simulation file selected (the output from the simulation process), the list that lets the user choose the reconstruction algorithm appears (Figure 30).

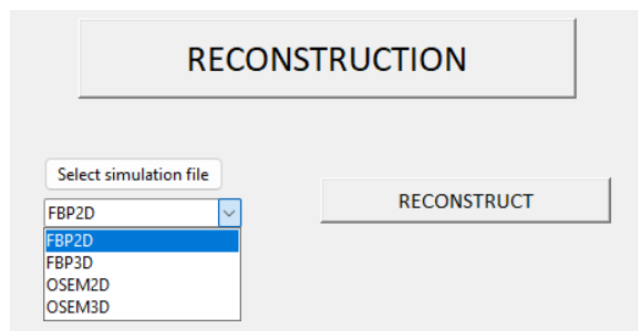


Figure 30. List of options to choose the reconstruction filter or algorithm preferred (Own source).

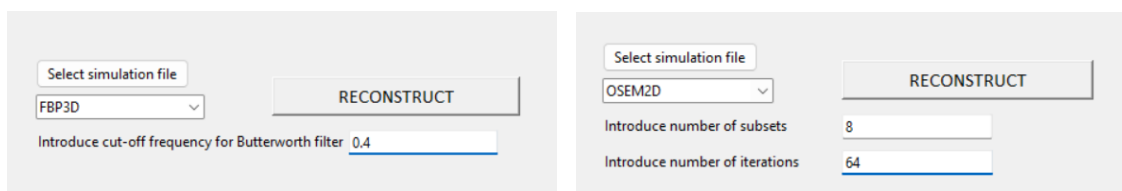
When reconstruction algorithm selected, text entries to let the user choose the specific parameters appear (*Figure 31*). As aforementioned, the parameters considered are:

For **FBP2D** and **FBP3D**:

- Cut off for ramp filter

For **OSEM2D** and **OSEM3D**:

- Number of subsets
- Number of iterations



**Figure 31.** Entries to introduce the reconstruction parameters for (left) FBP filters and for (right) iterative OSEM algorithm (Own source).

All the selected inputs will be introduced into the reconstruct program calling SPECTsim class, specifically the reconstruction class.

Finally, in *Figure 32*, the reconstruction pop-up window showing results for the chosen reconstruction algorithm and specific views. The organisation of the displayed image is top left–coronal view, top right–sagittal view, and left bottom–axial view, following the CSA display. At each side of the three images, the slider to control the slice shown is also seen.

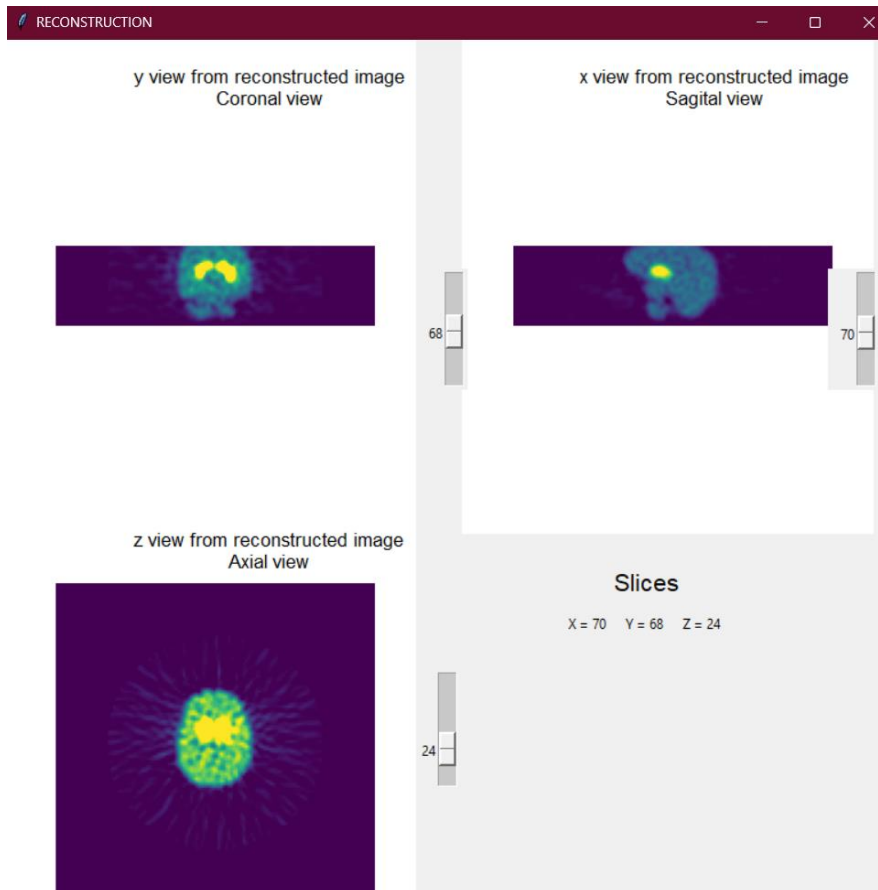


Figure 32. Reconstruction pop-up window (Own source).

## 5.5 Discussion

In order to carry out this project, the group in which the project has been developed proposed the use of SimSET and STIR as the software on which the development of the programme was to be based. Even so, different options of software packages that also work with Monte Carlo simulation, programming languages and libraries to generate the graphic interface were studied. In the end, the final choice was to follow the initial recommendation, as SimSET software had already been previously and there were reference scripts that could be adapted to Python. In addition, STIR is already prepared for use with the studies generated by SimSET. Finally, the Python tkinter library was chosen to build the graphical interface for its simplicity and functionality. To generate the back end of the programme, it was decisive for the final choice to use the same tools as SimPET to integrate the final product into their software.

The collection of accepted photons, shown in *Table 2*, summarizes the statistical summary obtained from each simulation process. As it is seen, the number of accepted photons is generally a tenth part from the original value of photons to simulate, decreasing this ratio when the number of photons increases. This is expected because Monte Carlo technique is a probabilistic model, so the number of photons tracking the collimator module and the number of accepted photons will decrease. However, this great loss of photons was not the expected, so to obtain feasible results, the number of simulated photons was augmented to show the result. The



problem with the simulation still was not identified, a deeper look to all the parameters introduced should be made.

Furthermore, a great difference between reconstruction applied with FBP (*Figure 25*) or OSEM (*Figure 26*) is seen, OSEM reconstruction performs better. In FBP reconstruction, the striatal uptake is seen, but the other algorithm suppresses the noise better and the brain is more differentiated, in FBP algorithm some brain regions are lost. This problem could be because of a missing parameter or a wrong one. However, in SPECT, nowadays OSEM algorithm is more used in the reconstruction process, as it is a much more complete method, so the results correspond to expected ones.

Moreover, as it may be seen in the results, the final graphical interface is functional and considers all files, data, and parameters necessary for the program. The images are shown in a Coronal-Sagittal-Axial (CSA) order and each one has a slider to be able to choose the slice of the image shown, achieving this way an interactive result where the user may see the whole resulting displayed image. Also, the simulation and reconstruction program work fine, but the considerations implemented in the back-end code always could be improved. For example, the simulation program does not consider the simulation of random photons, so the simulated images are not as realistic as they could be. Also, the parameters used were obtained from already existing examples of SimSET files developed by GIB-UB and other ones used in SimPET. A validation with other scanners and parameters should be done in future lines of work.

The time limitation should be considered when analysing the results. However, the results are great in general terms, as the development of the program – including both processes of simulation and reconstruction – and of a functional graphical interface has been achieved.

## 6. Execution schedule

### 6.1 Tasks and time definition.

This section is aimed to have a clear idea of the different tasks needed to complete the project. To organise these, a work breakdown structure has been made. In the following subsections, the diagram and the WBS dictionary may be observed.

#### 6.1.1 Work Breakdown Structure (WBS)

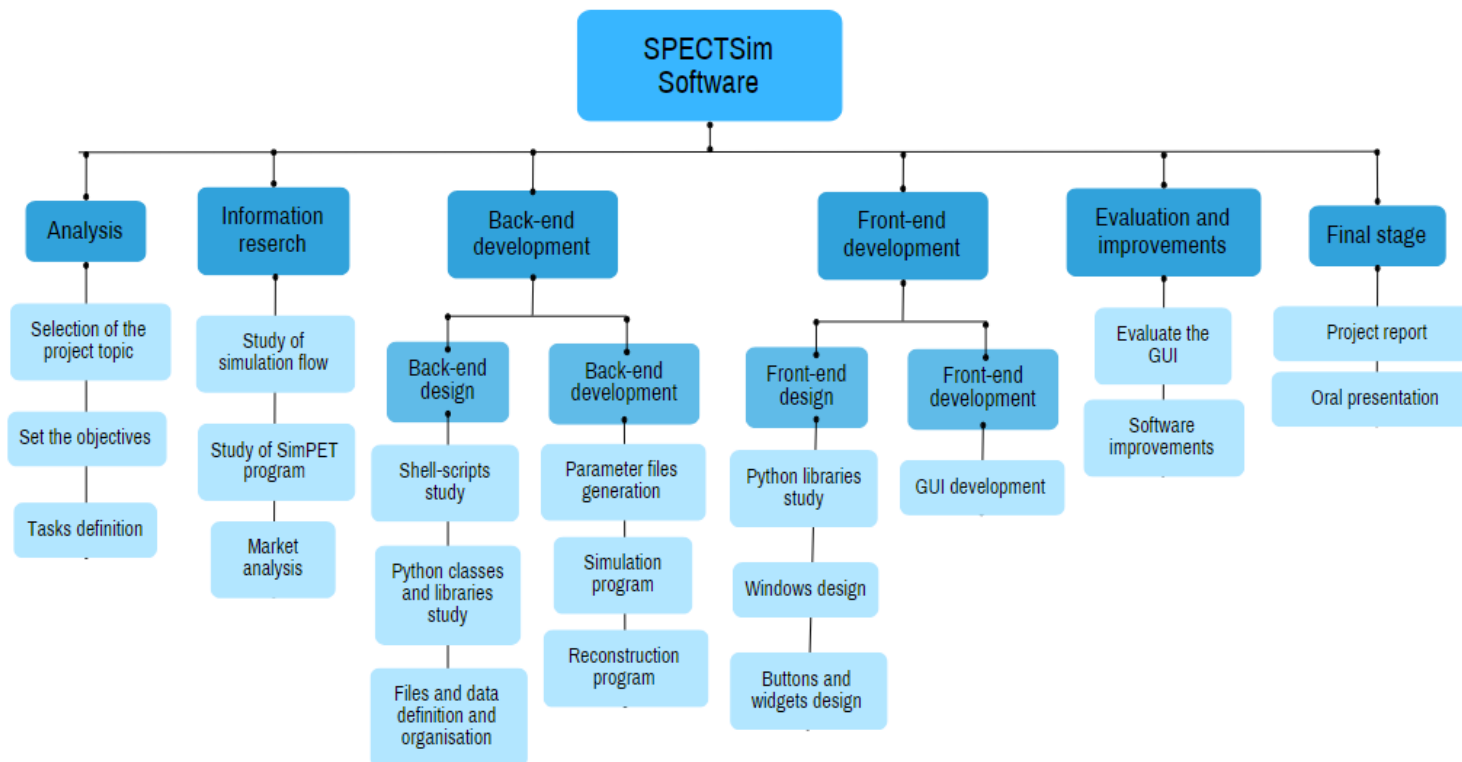


Figure 33. Work Breakdown Structure (WBS) of the project (Own source)

### 6.1.2 WBS dictionary

The project is divided into six main parts, each one includes several activities to achieve the final goal of the project; develop a functional graphical interface to generate simulations and their reconstruction and see the results. The distinct levels of activities, the task name and its description are compiled in the WBS dictionary, seen in *Table 2*.

*Table 2. WBS dictionary describing each task.*

Level	Task name	Description	Duration
<b>1</b>	<b>ANALYSIS</b>		
1.1	Selection of project topic	Presentation of possible themes, the final topic is then selected.	7 days Deliverable: 20/01/2023
1.2	Set the objectives of the project	Setting of the main aims concerning the project and final product.	2 days
1.3	Taks definition	Organisation and definition of the tasks needed to be done to fulfil the objectives.	4 days
<b>2</b>	<b>INFORMATION RESEARCH</b>		
2.1	Study of simulation flow	Study of SimSET and STIR programs to understand their flow and parameters requirement	5 days
2.2	Study of SimPET program	Study of SimPET available code on GitHub	5 days
2.3	Research about the state of the technology	Research about SPECT neuroimaging technique, DaTscan technique, quantification methods and reconstruction algorithms	4 days
2.4	Market analysis	Research about the sector the project is directed to and its main applications	4 days
<b>3</b>	<b>BACK-END DEVELOPMENT</b>		
3.1	Back-end design		
3.1.1	Study of shell scripts and programs developed by GIB-UB	Study of shell scripts codes where the simulation and reconstruction are performed, also the programs used	9 days
3.1.2	Python classes and libraries study	Study of classes generation and python libraries to fulfil the objectives of the project	5 days
3.1.2	Files and data definition and organisation	Definition of main and auxiliary files and organisation of the files and data (input and output)	5 days
3.2	Back-end development		
3.2.1	Parameter files generation	Write three main parameters files that are introduced into the program	5 days

3.2.2	Simulation program	Define classes, paths, and functions to construct the simulation program	30 days
3.2.3	Reconstruction program	Define classes, paths, and functions to construct the simulation program	33 days
<b>4</b>	<b>FRONT-END (GUI) DEVELOPMENT</b>		
<b>4.1</b>	<b>Front-end design</b>		
4.1.1	Python libraries study	Study of different python libraries available to develop a graphical interface	3 days
4.1.2	Windows design	Visual design of the three windows (main, simulation and reconstruction)	3 days
4.1.2	Buttons and widgets design	Design of the buttons and widgets for the windows and their functionality	4 days
<b>4.2</b>	<b>Front-end development</b>		
4.2.1	GUI development	Development of the buttons and widgets, implementation of the program and organise the visualisation of results	18 days
<b>5</b>	<b>EVALUATION AND IMPROVEMENTS</b>		
5.1	Evaluate the graphical interface	Evaluate the functioning of the GUI and test for different data	3 days
5.2	Software improvements	Improvements of errors and optimizing the code	16 days
<b>6</b>	<b>PRESENTATION AND REPORT</b>		
6.1	Project report	Throughout the execution of the project, a memory of the process performed will be edited	60 days Deliverable: 07/06/2023
6.2	Preparation of the oral presentation	Design of the oral presentation (10 minutes) and organise the information to expose	6 days Deliverable: 14/06/2023

## 6.2 Precedence analysis and critical path method

To determine the time (in days) that each task will need, a precedence analysis has been performed. This way, the PERT time may be known following the following expression:

$$t_{PERT} = \frac{t_{OPTIMISTIC} + 4t_{NORMAL} + t_{PESSIMISTIC}}{6} \quad (\text{Equation 2})$$

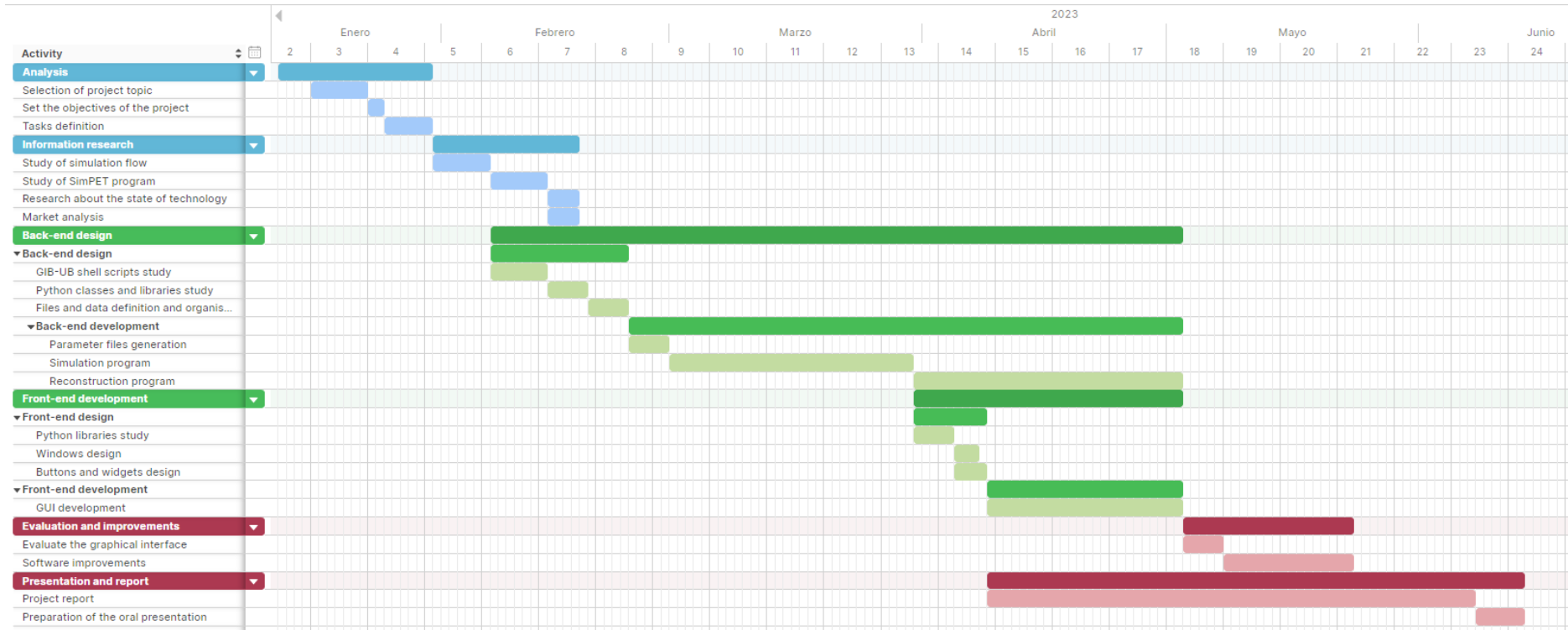
*Table 3. PERT, Optimistic, normal and pessimistic time of each activity.*

Tasks	ID	PERT time	Optimistic time	Normal time	Pessimistic time
1.1 Selection of the project topic	A	7	6	7	8
1.2 Set the objectives of the project	B	2	1	2	3
1.3 Tasks definition	C	4	3	4	5
2.1 Study of simulation flow	D	5,16	4	5	7
2.2 Study of SimPET program	E	5,16	4	5	7
2.3 Research about state of technology	F	4	3	4	5
2.4 Market analysis	G	4,16	3	4	6
3.1.1 GIB-UB shell scripts study	H	9	7	9	11
3.1.2 Python classes and libraries study	I	4,83	2	5	7
3.1.3 Files and data definition and organisation	J	5	3	5	7
3.2.1 Parameter files generation	K	5	4	5	6
3.2.2 Simulation program	L	30,33	27	30	35
3.2.3 Reconstruction program	M	32,83	30	33	35
4.1.1 Python libraries study	O	3	2	3	4
4.1.2 Windows design	N	2,83	1	3	4
4.1.3 Buttons and widgets design	P	3,6	1	4	5
4.2.1 GUI development	Q	20	16	18	20
5.1 Evaluate the graphical interface	R	3	2	3	4
5.2 Software improvements	S	16	14	16	18
6.1 Project report	T	60	50	60	70
6.2 Preparation of the oral presentation	U	6	5	6	8



### 6.3 Schedule; GANTT diagram

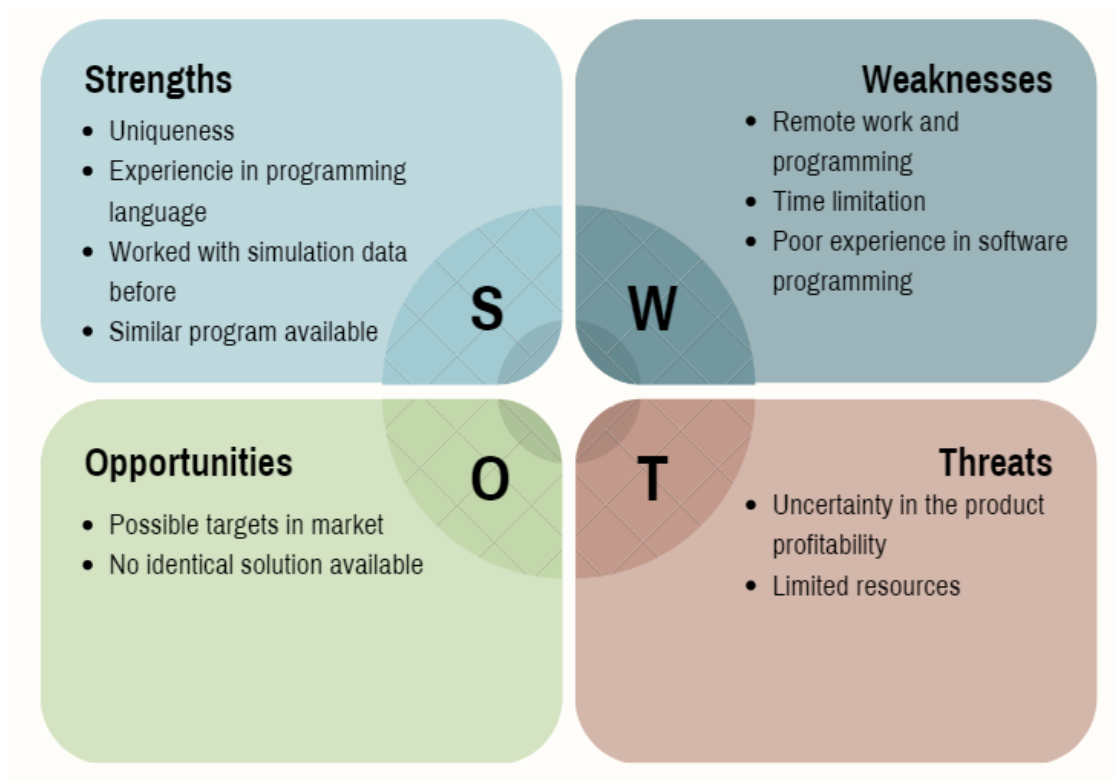
To control and schedule the different tasks for this project over time, a GANTT diagram has been created, seen in *Figure 34*.



*Figure 34.* GANTT diagram (Own source).

## 7. Technical viability

In this part a technical viability study will be performed by using the SWOT analysis technique, a diagram where the strengths, weaknesses, opportunities, and threats of the project are listed. This provides a general idea of the feasibility of the project and have an idea of the inconveniences that may appear when developing the chosen solution. The SWOT diagram is observed in *Figure 35*.



*Figure 35. SWOT diagram to study technical viability (Own source).*

In relation to the **strengths** of the project, the most important is the previous experience both with the programming language chosen, Python, and with SPECT simulated data. This reduces the time required to develop the project, as there is no need to learn a new programming language. Also, the DaTscan simulation and reconstruction was used in a lecture in the Biomedical Engineering degree in University of Barcelona, so the flow, programs, inputs, and outputs of the process are known. Besides, there is SimPET, and the code is available, so this is also an important strength for the project, as having a base of a program on which to resort when encountering a problem is beneficial to reduce time.

As far as the **opportunities** of the product are concerned, there are possible targets in the market to direct the results of the project, as it is a new solution which does not have an identical product available.



As for **weaknesses**, the three main aspects considered are the remote working and programming, time limitation and poor experience on software programming. Remote working and programming are a drawback with respect to the project slowdown, as there is a dependence on the power of the internet connection, and it has not a graphical interface to visualize the results of the code execution. However, it permits to work on Linux in every computer without this computer system. Also, time limitation is a fact to consider, as this program is a Final Degree project, so it has strict time limits. Finally, not having developed a software before may cause problems to be encountered more easily. But, as there is a lot of free available documentation of the chosen libraries, they could be solved.

Finally, regarding the **threats** of the project, the uncertainty of the product profitability is considered, as the usage of the project results by the sectors to which it is directed is unknown. But considering the uniqueness of the project, and its utility in the research field, there is hope that it happens to be a useful tool for investigation.

## 8. Economic viability

The aim of this section is to study the required costs with respect to the execution of the project. The materials, services and personnel costs are included into this study.

Regarding the **materials**, the Brain CT and the Alderson striatal brain phantom have been considered, as the activity and attenuation objects used to test the software are obtained from a CT scan. Moreover, personal materials like the computer used to develop the project are considered. *Table 4* shows the costs of the above-mentioned.

*Table 4. Materials cost of the project.*

Material	Unit price (€)	Units	Total (€)
Brain CT	210,00€	1	210,00€
Striatal brain Phantom (Alderson)	~1000,00€	1	1000,00€
Lenovo Yoga 530 -14 IKB (i7)	1.099,00€	1	1.099,00€
<b>Total</b>			<b>2.309,00€</b>

Also, a consideration of all software tools and licenses used in this project is done. In the list, the software packages used, Anaconda support platform and Office 365 license appear. The costs regarding the mentioned services are collected in *Table 5*.

*Table 5. Services cost of the project.*

Services	Unit price (€)	Units	Total (€)
Anaconda spyder	Free software (0€)	1	0€
SFTP drive	Free software (0€)	1	0€
SimSET	Free source (0€)	1	0€
STIR	Free software (0€)	1	0€
Office 365 license	5,60€/month	6	33,60€
<b>Total</b>			<b>33,60€</b>

With respect to the **personnel** cost, both the tutor and student have been considered. The salary data is retrieved from the Glassdoor web [50], searching for the mid salaries of a junior engineer and of a senior one.

*Table 6. Personnel cost of the project*

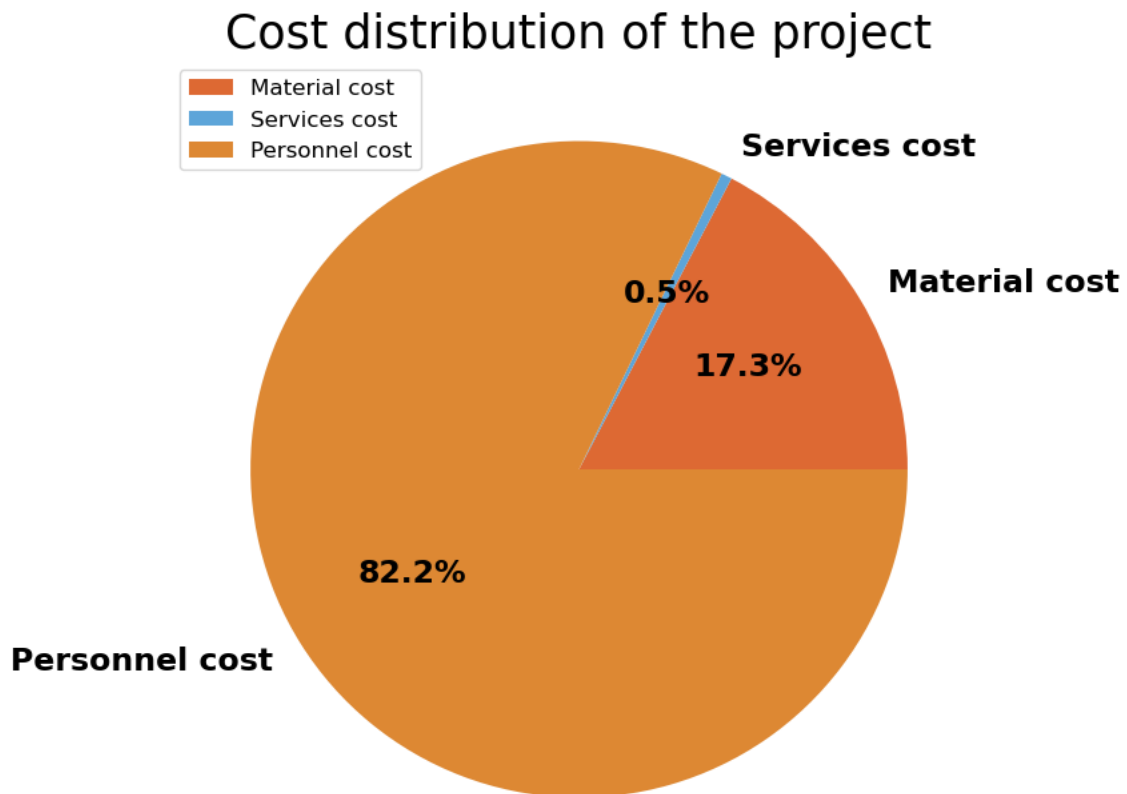
Personnel	Salary	Hours	Total (€)
Biomedical engineer	30€/h.	80	2.400,00€
Junior biomedical engineer	9,375€/h.	300	2.812,50€
<b>Total</b>			<b>5.212,50€</b>

So, the final budget to develop the project is the following seen in *Table 7*. The final cost would be of **7555,10€**

*Table 7. Complete cost of the project*

	Price (€)
Material cost	2.309,00€
Services cost	33,60€
Personnel cost	5.212,50€
<b>Total</b>	<b>7555,10€</b>

The distribution of the cost of the project is seen in a pie chart, dividing the costs in services, material, and personnel (*Figure 36*).



*Figure 36. Pie chart showing the cost distribution of the project (Own source)*

## 9. Legislation and regulation

As this software does not imply the use of medical data from patients or subjects, the legislation and regulation for medical devices does not apply. However, some regulations have been considered to ensure the software has a reliable legal background.

This project is directed to research groups and companies to validate acquisition, reconstruction, and quantitative protocols and methods, being also useful to characterize different SPECT cameras. To do that, phantom or even patient activity and attenuation objects may be used. As the project is also used for research, **data protection and privacy** regulation must be considered. Personal data collected and used for research is covered by the UK General Data Protection Regulation (GDPR) and the Data Protection Act 2018 (DPA). Also, the code of integrity in research of the University of Barcelona should be considered, as this project is being developed in the Biomedical Imaging Group (GIB-UB) of this University.

Moreover, it is key to mark that this project is an adaption from SimPET open-source code to be able to generate SPECT data. Also, it uses SimSET and STIR software packages, so a consideration of **intellectual property rights** must be done. All packages used in this project are open-source software, so their license guarantee their use, study, change and distribution. Each of the three software have a concrete license, therefore a brief study will be made to know what has to be taken into consideration to carry out this project.

SimPET and SimSET have a GNU GPLv3 license (General Public Use General Public License version 3). This license guarantees the user to execute, study, share and modify the software. In addition, GPL uses copyleft mechanism, meaning that any derivative software based on the licensed one must also be licensed under GPL, ensuring that rights to access, modify and distribute the source code are maintained [51]. Also, a mention of the use of such software for crediting purposes should be made.

On the other hand, STIR has an Apache 2.0 license, which is a permissive license but with more conditions than the previous. The permissions to use guarantee users a license to use the software either for commercial or non-commercial use. With respect to the source code modification, derivatives from it are not required to be distributed under Apache 2.0 license, so the combination of the source software with other programmes may be under different terms. Finally, regarding the copyright, the license mentions that *“the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work”* [52]

As the final product is intended to be integrated with SimPET platforms, a **collaboration agreement** would maybe be needed, as the initial software idea is of their own.



## 10. Conclusions and future work

The aim of this project was to develop a Monte Carlo simulation software for the generation of realistic SPECT data. Therefore, three main objectives were set: to generate a program that would define two classes (the simulation class and the reconstruction class), and then to integrate them in a graphical interface that would allow the user to choose the main files and display the result.

Looking at the final product, it can be concluded that the three main objectives of the project have been achieved, since a functional graphic interface has been developed that integrates the simulation and reconstruction programmes and allows the visualisation of the results generated based on the inputs chosen by the user.

Even so, there is still some work to do before the application could be used by other users. In the first place, the construction of a web to be able to use the product online should be made, making the software cross-platform as it could be used in all systems. The aforementioned involves obtaining a web domain and building a database to store the results generated. If the final project could be integrated with SimPET platform, the generation of activity and attenuation would be possible.

As other future lines of work, it would also be convenient, on the one hand, to introduce the images generated in a quantification software and compare them with the theoretical values, Moreover, consideration of other SPECT scanners should be made, as the parameters used in this project were obtained from the available file from GIB-UB group.

Finally, it is key to conclude that even if the main focus of this project was the simulation of DaTscan SPECT data because of the absence of standardization among its quantitative methodology, the final result of this project could be used to simulate other nuclear imaging SPECT scans like cardiac SPECT, ictal and interictal images for epilepsy, etc. This way, a validation of the processing software could be made with other type of SPECT studies.

## Bibliography

- [1] *Parkinson's Disease: Causes, Symptoms, and Treatments*. (2022). National Institute on Aging. <https://www.nia.nih.gov/health/parkinsons-disease>
- [2] NHS (2022, 11 noviembre). *Causes*. <https://www.nhs.uk/conditions/parkinsons-disease/causes/>
- [3] Haider, A., Elghazawy, N.H., Dawoud, A. et al. Translational molecular imaging and drug development in Parkinson's disease. *Mol Neurodegeneration* 18, 11 (2023). <https://doi.org/10.1186/s13024-023-00600-z>
- [4] Kangasmaa, T., Hippeläinen, E., Constable, C. et al. Quantitative Monte Carlo-based brain dopamine transporter SPECT imaging. *Ann Nucl Med* 35, 17–23 (2021). <https://doi-org.sire.ub.edu/10.1007/s12149-020-01532-0>
- [5] Paredes-Pacheco, J., López-González, F. J., Silva-Rodríguez, J., Efthimiou, N., Niñerola-Baizán, A., Ruibal, Á., Roé-Vellvé, N., & Aguiar, P. (2021). SimPET-An open online platform for the Monte Carlo simulation of realistic brain PET data. Validation for 18 F-FDG scans. *Medical physics*, 48(5), 2482–2493. <https://doi.org/10.1002/mp.14838>
- [6] Gnesin, S., Leite Ferreira, P., Malterre, J., Laub, P., Prior, J. O., & Verdun, F. R. (2016). Phantom Validation of Tc-99m Absolute Quantification in a SPECT/CT Commercial Device. *Computational and mathematical methods in medicine*, 2016, 4360371. <https://doi.org/10.1155/2016/4360371>
- [7] Bahreyni Toossi, M. T., et al. (2010). SIMIND Monte Carlo simulation of a single photon emission CT. *Journal of medical physics*, 35(1), 42–47. <https://doi.org/10.4103/0971-6203.55967>
- [8] Yandrapalli, S. (2022, October 3). *SPECT Imaging*. StatPearls - NCBI Bookshelf. <https://www.ncbi.nlm.nih.gov/books/NBK564426/>
- [9] Martín-Comin, J. et al. (2012). *Medicina Nuclear en la práctica clínica* (2.<sup>a</sup> ed., pp. 59–83). Aula Médica.
- [10] *SPECT scan - Mayo Clinic*. (2022, July 27). <https://www.mayoclinic.org/tests-procedures/spect-scan/about/pac-20384925>
- [11] Themes, U. (2016, February 26). *Tomographic Reconstruction in Nuclear Medicine*. Radiology Key. <https://radiologykey.com/tomographic-reconstruction-in-nuclear-medicine/>
- [12] Ema. (2022, February 10). DaTSCAN - European Medicines Agency. European Medicines Agency. <https://www.ema.europa.eu/en/medicines/human/EPAR/datscan>
- [13] Niñerola, A. (2015). Quantification of striatal dopamine transporter SPECT in animal models and clinical research. *Universitat de Barcelona*.
- [14] Zhu Y. M. (2018). Ordered subset expectation maximization algorithm for positron emission tomographic image reconstruction using belief kernels. *Journal of medical imaging* (Bellingham, Wash.), 5(4), 044005. <https://doi.org/10.1117/1.JMI.5.4.044005>



- [15] <https://aapm.onlinelibrary.wiley.com/doi/pdfdirect/10.1118/1.598559>
- [16] SimPET - Simulation of PET studies. (n.d.). <http://simpetweb.uma.es:8020/manual>
- [17] Silva-Rodríguez, J. (2019). *simpet*. GitHub. <https://github.com/txusser/simpet>
- [18] Morbelli, S., et al. (2020). EANM practice guideline/SNMMI procedure standard for dopaminergic imaging in Parkinsonian syndromes 1.0. *European journal of nuclear medicine and molecular imaging*, 47(8), 1885–1912. <https://doi.org/10.1007/s00259-020-04817-8>
- [19] López-González, F. J., et al. & Alzheimer’s Disease Neuroimaging Initiative (2019). Spill-in counts in the quantification of 18F-florbetapir on A $\beta$ -negative subjects: the effect of including white matter in the reference region. *EJNMMI physics*, 6(1), 27. <https://doi.org/10.1186/s40658-019-0258-7>
- [20] Dickson, J. C., Tossici-Bolt, L., Sera, T., Erlandsson, K., Varrone, A., Tatsch, K., & Hutton, B. F. (2010). The impact of reconstruction method on the quantification of DaTSCAN images. *European journal of nuclear medicine and molecular imaging*, 37(1), 23–35. <https://doi.org/10.1007/s00259-009-1212-z>
- [21] Dickson, J. C., et al. (2017). The impact of reconstruction and scanner characterisation on the diagnostic capability of a normal database for [123I]FP-CIT SPECT imaging. *EJNMMI research*, 7(1), 10. <https://doi.org/10.1186/s13550-016-0253-0>
- [22] Koch, W., et al. (2005). Clinical testing of an optimized software solution for an automated, observer-independent evaluation of dopamine transporter SPECT studies. *Journal of nuclear medicine : official publication, Society of Nuclear Medicine*, 46(7), 1109–1118.
- [23] Valteri Kaasinen (2023). How to semi-quantify <sup>123</sup>I-Ioflupane. ESMIT 2023 Live Webinar Series
- [24] Calvini, P., et al. (2007). The basal ganglia matching tools package for striatal uptake semi-quantification: description and validation. *European journal of nuclear medicine and molecular imaging*, 34(8), 1240–1253. <https://doi.org/10.1007/s00259-006-0357-2>
- [25] Brogley J. E. (2019). DaTQUANT: The Future of Diagnosing Parkinson Disease. *Journal of nuclear medicine technology*, 47(1), 21–26. <https://doi.org/10.2967/jnmt.118.222349>
- [26] Mirion Technologies. (n.d.). *Hoffman 3-D brain phantom*. Retrieved May 30, 2023, from <https://www.mirion.com/products/medical/nuclear-medicine-instrumentation/quality-assurance/phantoms/hoffman-3-d-brain-phantom>
- [27] Radiology Support Devices Inc. (2022, June 15). *Striatal Phantom - Radiology Support Devices Inc.* Retrieved May 30, 2023, from <https://rsdphantoms.com/product/striatal-phantom/>
- [28] Schmitz, R. E., et al. (2007). Expanding SimSET to include block detectors: performance with pseudo-blocks and a true block model. *IEEE Nuclear Science Symposium conference record. Nuclear Science Symposium*, 6, 4275–4278. <https://doi.org/10.1109/NSSMIC.2007.4437061>



- [29] Poon, J. K., et al. (2015). Validation of the SimSET simulation package for modeling the Siemens Biograph mCT PET scanner. *Physics in medicine and biology*, 60(3), N35–N45. <https://doi.org/10.1088/0031-9155/60/3/N35>
- [30] Cao, N., et al. (2010). Detection performance analysis for time-of-flight PET. *Physics in medicine and biology*, 55(22), 6931–6950. <https://doi.org/10.1088/0031-9155/55/22/021>
- [31] Cot, A., et al. (2005). Absolute quantification in dopaminergic neurotransmission SPECT using a Monte Carlo-based scatter correction and fully 3-dimensional reconstruction. *Journal of nuclear medicine: official publication, Society of Nuclear Medicine*, 46(9), 1497–1504.
- [32] Polycarpou, I., et al. (2011). Comparative evaluation of scatter correction in 3D PET using different scatter-level approximations. *Annals of nuclear medicine*, 25(9), 643–649. <https://doi.org/10.1007/s12149-011-0514-y>
- [33] Fuster, B. M., et al. (2013). Integration of advanced 3D SPECT modeling into the open-source STIR framework. *Medical physics*, 40(9), 092502. <https://doi.org/10.1118/1.4816676>
- [34] Khateri, P., et al. (2019). Implementation of cylindrical PET scanners with block detector geometry in STIR. *EJNMMI physics*, 6(1), 15. <https://doi.org/10.1186/s40658-019-0248-9>
- [35] Siemens Healthineers. (2017). *Xspect Quant enables standardized quantification*. Retrieved June 3, 2023, from <https://www.siemens-healthineers.com/es/molecular-imaging/mi-clinical-corner/clinical-case-studies/xspect-quant-enables-standardized-quantification.html>
- [36] GE Healthcare. (2020). *DaTQUANT 2.0 Quantitative analysis of dopamine transporter (DaT) SPECT images at your fingertips*. Retrieved June 3, 2023, from <https://www.gehealthcare.com/products/nuclear-imaging-agents/datquant>
- [37] Niñerola, A. et al. (2014). QuantiDOPA: A Quantification Software for Dopaminergic Neurotransmission SPECT. In: "XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013", 25/09/2013 - 28/09/2013, Seville, Spain. pp. 443-446. [https://doi.org/10.1007/978-3-319-00846-2\\_110](https://doi.org/10.1007/978-3-319-00846-2_110).
- [38] Geant4 Collaboration (2022). Introduction to Geant4. Release 11.1. Retrieved, May 1, 2023, from <https://geant4userdoc.web.cern.ch/UsersGuides/IntroductionToGeant4/fo/IntroductionToGeant4.pdf>
- [39] *Simulation System for Emission Tomography (SimSET) Home Page*. (n.d.). Retrieved, May 1, 2023, from [https://depts.washington.edu/simset/html/simset\\_main.html](https://depts.washington.edu/simset/html/simset_main.html)
- [40] MCNP® Website. (n.d.). Retrieved, May 1, 2023, from <https://mcnp.lanl.gov/>
- [41] Bozkurt, Ahment (2005). Issues related to the used of MCNP code for an extremely large voxel model VIP-MAN. *American Nuclear Society Topical Meeting*. Retrieved from [https://mcnp.lanl.gov/pdf\\_files/TechReport\\_2004\\_LANL\\_LA-UR-04-8671\\_BozkurtXuEtAl.pdf](https://mcnp.lanl.gov/pdf_files/TechReport_2004_LANL_LA-UR-04-8671_BozkurtXuEtAl.pdf)
- [42] *STIR Home Page*. (n.d.). Retrieved, May 1, 2023, from <https://stir.sourceforge.net/>



- [43] Berga, M. (2023). *Python vs Java: key differences and code examples*. Blog | Imaginary Cloud. Retrieved, May 1, 2023, from <https://www.imaginarycloud.com/blog/python-vs-java/#Python>
- [44] *tkinter — Interface de Python para Tcl/Tk*. (n.d.). Python Documentation. <https://docs.python.org/es/3/library/tkinter.html#module-tkinter>. Retrieved, May 1, 2023, from
- [45] *Introduction — PyQt Documentation v5.15.4*. (n.d.). Retrieved, May 1, 2023, from <https://www.riverbankcomputing.com/static/Docs/PyQt5/introduction.html>
- [46] Facialix. (2020). *Las mejores bibliotecas de código para crear interfaces gráficas en Python*. Facialix. Retrieved, May 1, 2023, <https://blog.facialix.com/las-mejores-bibliotecas-de-codigo-para-crear-interfaces-graficas-en-python/>
- [47] Team, W. (2021, August 2). *Welcome to wxPython!* wxPython. Retrieved, May 10, 2023, from <https://www.wxpython.org/>
- [48] Silva-Rodríguez, J. (2019). *simpet/tools.py at master · txusser/simpet*. GitHub. Retrieved, March 2023, from <https://github.com/txusser/simpet/blob/master/utis/tools.py>
- [49] Silva-Rodríguez, J. (2019). *simpet/tools.py at master · txusser/simpet*. GitHub. Retrieved, March 2023, from [https://github.com/txusser/simpet/blob/master/src/stir/stir\\_tools.py](https://github.com/txusser/simpet/blob/master/src/stir/stir_tools.py)
- [50] Glassdoor. (s. f.). *Búsqueda de empleo en Glassdoor*. Retrieved, May 10, 2023, from <https://www.glassdoor.es/>
- [51] *The GNU General Public License v3.0 - GNU Project - Free Software Foundation*. (n.d.). Retrieved, May 29, 2023, from <https://www.gnu.org/licenses/gpl-3.0.html>
- [52] *Apache License, Version 2.0*. (n.d.). Retrieved, May 29, 2023, from <https://www.apache.org/licenses/LICENSE-2.0>



# ANNEXES

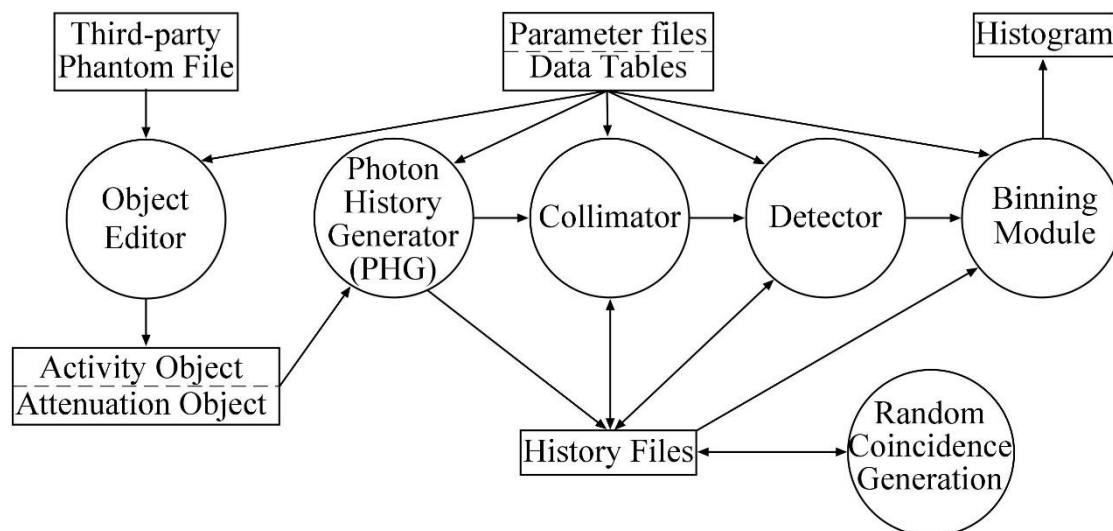
## ANNEX 1 SimSET package

### A1.1 SimSET

SimSET program is a software package designed to perform Monte Carlo simulations for emission tomography imaging, simulating photon generation and transport through attenuators. It includes four modules, each one with its respective file defining the specific parameters.

- Collimator module → Configured with file *col.rec*
- Detector module → Configured with file *det.rec*
- Binning module → Configured with file *bin.rec*
- Photon History Generator (PHG) module → Configured with file *phg.rec*

The general overview of the program is the one seen in *Figure A1*, where a diagram showing the interaction between modules is observed.



**Figure A1.** Block diagram illustrating the fit between the different modules of SimSET package.

To install SimSET, you can visit their repository (Link: <https://depts.washington.edu/simset/>). Refer to SimSET user and installation manual to find detailed information about how to install and the program.

### A1.2 SimSET modules

As aforementioned, SimSET has four modules to generate de emission tomography simulated data, which will be now explained. All this information is retrieved from their webpage, corresponding to Ref. 39.

### A1.2.1 Photon History Generator (PHG) module

#### Description of the module

The Photon History Generator (PHG) is a software application designed to perform Monte-Carlo simulations of photon creation and transport through heterogeneous attenuators for both SPECT and PET.

In Figure A2, a diagram showing PHG inputs and outputs is shown.

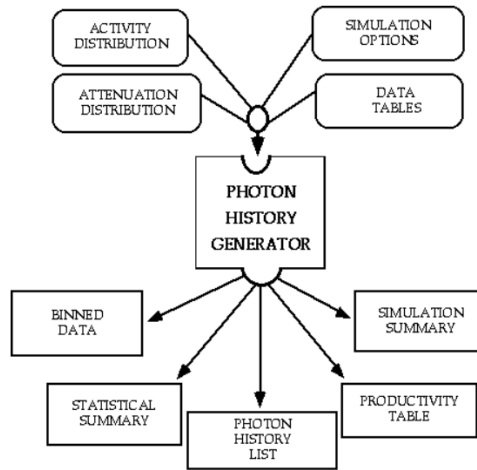


Figure A2. PHG inputs and outputs.

From the activity object and the specified number of photons (specified in the phg.rec file with *num\_to\_simulate* parameter), the number of decays to generate for each voxel is computed. The photon tracking algorithm is seen in Figure A3.

As it is seen, the decay is generated, and photons are produced. Then, the photons are projected to the target cylinder within a user-specified acceptance angle. The photons that reach the cylinder will be passed to the collimator/detected/binning modules and included in the history file.

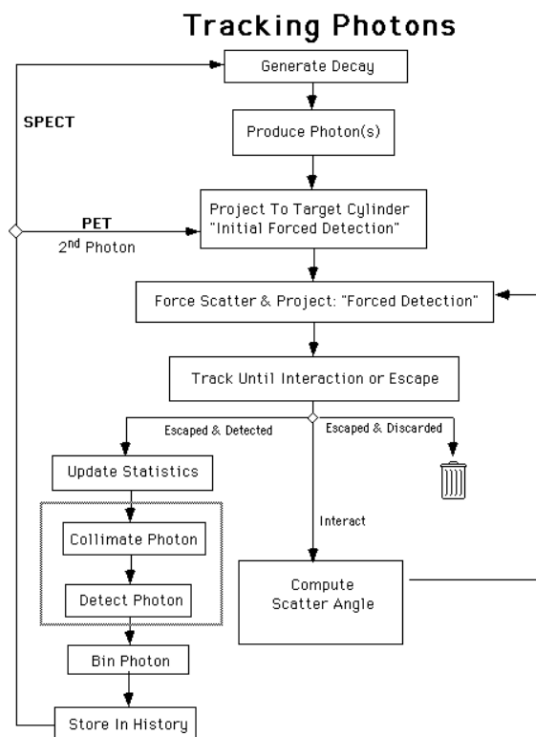


Figure A3. Control flow for tracking photons in PHG module

## Configuration of the PHG module

The module uses a parameter file to define the simulation options, define the target cylinder, and activity and attenuation maps in ascii text file. Therefore, the entries syntax is in the following

DATA\_TYPE    parameter\_name = value

In the following table, the parameters considered to configure PHG module in this project is seen. All parameters are fitted in *phg.rec* file.

**Table A1.** Considered parameters to configure PHG module

DATA TYPE	PARAMETER NAME	VALUE
<b>RUNTIME OPTIONS</b>		
INT	num_to_simulate	50.000.000
REAL	length_of_scan	100
BOOL	simulate_SPECT	True
REAL	minimum_energy	100.0 keV
REAL	photon_energy	140.0 keV
BOOL	simulate_stratification	False
BOOL	simulate_forced_detection	False
BOOL	adjust_for_positron_range	False
BOOL	adjust_for_collinearity	False
REAL	weight_window_ratio	1.0
BOOL	point_source_voxels	True
INT	random_seed	0
BOOL	model_coherent_scatter	False
BOOL	model_coherent_scatter_in_obj	False
INT	num_slices	32
<b>TARGET CYLINDER INFORMATION</b>		
REAL	target_zMin	0.0
REAL	target_zMax	16.0
REAL	radius	13.0
<b>COHERENT ANGULAR DISTRIBUTION FILES</b>		
STR	coherent_scatter_table	
STR	activity_indexes	
STR	activity_table	
STR	activity_index_trans	
STR	activity_image	
STR	attenuation_indexes	
STR	attenuation_table	
STR	attenuation_index_trans	
STR	attenuation_image	
STR	productivity_input_table	
STR	productivity_output_table	
STR	forced_detection_table	
STR	collimator_params_file	
STR	detector_params_file	
STR	bin_params_file	
STR	history_file	

For the coherent angular distribution files, the value must be the path of the specific file. Note that the files own from SimSET are marked with bold letter.

### A1.2.2 Collimator module

#### Description of the module

For UNC SPECT simulations, the collimator is modelled using a geometric transfer function and three types of hole geometries are available: parallel hole, fan beam and cone beam. or parallel hole collimation the hole is a right circular cylinder perpendicular to the collimator face; for fan-beam the holes are tapered, slanted transaxially to focus on a focal line; for cone-beam the holes are tapered, slanted both axially and transaxially to focus on a focal point (Figure A4)

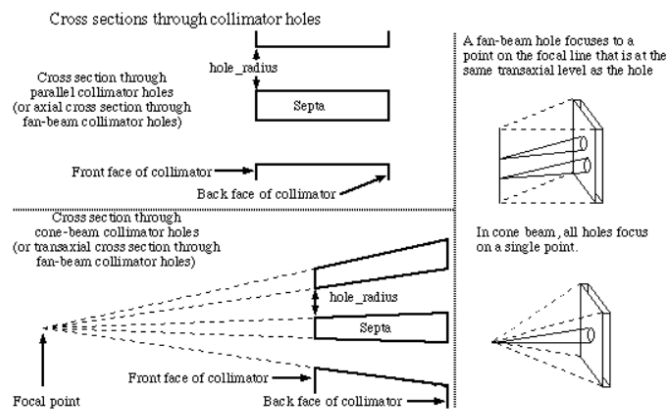


Figure A4. Fan- beam and Cone-beam collimation in UNC SPECT simulation

A hexagonal design is used to simulate the collimator holes. The holes do not correspond to fixed places because the collimator response is computed as though the pattern were moved randomly over the collimator face. The user-specified hole radius and septal thickness determine the density of the holes. The probability that a photon will travel through a hole without striking the collimator is computed for each photon. Photons whose weights are multiplied by this probability and for which this probability is non-zero are accepted.

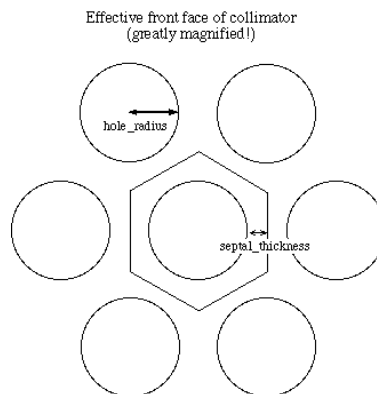


Figure A5. Front face of collimator in a UNC SPECT simulation



### Configuration of collimator module

In the following table, the parameters considered to configure collimator module in this project is seen. All parameters are fitted in *col.rec* file.

**Table A2.** Considered parameters to configure collimator module

DATA TYPE	PARAMETER NAME	VALUE	DESCRIPTION
ENUM	hole_geometry	parallel	Parallel holes, fan, or cone beam
ENUM	collimator_type	unc_spect	Type of collimator
INT	focal_length	0.0	Non-zero if fan or cone beam
BOOL	radius_of_rotation	13.00	Inside radius of collimator rotation
REAL	thickness	4.8	Desired depth of collimator
REAL	hole_radius	0.0866	Hole radius
REAL	septal_thickness	0.02	Half the desired distance between holes
REAL	min_z	0.0	Desired minimum axial extent of the collimator
REAL	max_z	16.0	Desired maximum axial extent of the collimator
REAL	start_angle	0.0	First rotation angle
REAL	stop_angle	360.0	Last rotation angle
INT	num_views	120	Number of discrete sampling positions between start and stop angles
STR	history_file	“ “	History file, if empty, no history file

#### A1.2.3 Detector module

### Description of the module

*The Detector module receives photons either directly from the PHG module or from the Collimator module. It tracks photons through the specified detector, recording the interactions within the detector for each photon. The interactions are used to compute a detected location and total energy deposited. Detected Photon Data is passed from the detector module to either the binning module or the history file module depending on the options specified in the PHG Parameter File.*

The detector modelled in this project is a simple SPECT.

### Configuration of detector module

In the following table, the parameters considered to configure detector module in this project is seen. All parameters are fitted in *det.rec* file.

**Table A3.** Considered parameters to configure detector module

DATA TYPE	PARAMETER NAME	VALUE	DESCRIPTION
INT	detector_type	2	Simple SPECT
REAL	energy_resolution_percentage	10	Energy resolution of the detector (% of FWHM)
REAL	reference_energy_keV	140	Reference energy
STR	history_file	“ “	History file, if empty, no history file

### A1.2.3 Binning module

#### Description of the module

The Binning Module is used to process photon and detection records. It can be used during the simulation as each record is accepted (on-the-fly), or it can be used after the simulation via processing a standard history file. It "bins" data into multi-dimensional histograms. The binning module will create histograms based on several different parameters which are described below.

#### Configuration of binning module

In the following table, the parameters considered to configure detector module in this project is seen. All parameters are fitted in *bin.rec* file.

**Table A4.** Considered parameters to configure binning module

DATA TYPE	PARAMETER NAME	VALUE	DESCRIPTION
INT	scatter_param	1	Simulation with primary + scatter photons
INT	min_s	0	As scatter param is 1, needs to be set to 0
INT	max_s	9	Number of scatter bins
INT	num_aa_bins	120	Number of views
INT	num_z_bins	32	Number of slices
REAL	min_z	0.0	Minimum value for z binning
REAL	max_z	16.0	Maximum value for z binning
INT	num_td_bins	128	Required number of transaxial distance bins
REAL	min_td	-28.3136	Minimum value of transaxial distance
REAL	max_td	28.3136	Maximum value of transaxial distance
INT	num_e1_bins	1	Binning by photon energy
INT	num_e2_bins	1	Ignored for SPECT
REAL	min_e	120.0	Minimum energy
REAL	max_e	250.0	Maximum energy
INT	weight_image_type	2	4-bytes
INT	count_image_type	2	4-bytes
BOOL	add_to_existing_img	false	Write into existing simulation
STR	weight_image_path	rec.weight	Path to weight image
STR	weight_squared_image_path	rec.weight2	Path to square weight image
STR	history_file	" "	History file, if empty, no history file

## ANNEX 2 STIR package

### A2.1 STIR

STIR is an open-source library written in C++ that facilitates the reconstruction and processing of emission tomographic data. It offers various algorithms and scanner geometries, and it is designed to be compatible with different programming platforms. The license of STIR is under Apache 2.0 regulation and some parts under the PARAPET license, as it was initially developed as a part of this project.

To install STIR, you can visit their website (Link: <https://stir.sourceforge.net/>) and register. Once downloaded, STIR directory and subdirectories will be found where you extract the file. Refer to STIR user manual to find detailed information about the compilation of the program.

### A2.1 SPECT reconstruction

To perform a SPECT reconstruction using STIR two files must be created: an interfile and an algorithm parameter file. In the interfile, the stir sinogram and other variables like the start angle, the matrix size, the radius of the scanner, the type of data, etc. must be specified (An example is seen in *Figure A6*).

```
!INTERFILE :=
!imaging modality := nucmed
name of data file :=/NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/stir_sinogram.s
!version of keys := 3.1
!GENERAL DATA :=
!GENERAL IMAGE DATA :=
!type of data := Tomographic
patient rotation := prone
patient orientation := feet_in
imagedata byte order := LITTLEENDIAN
!SPECT STUDY (General) :=
!number format := float
!number of bytes per pixel := 4
!number of projections := 120
!extent of rotation := 360
process status := acquired
!SPECT STUDY (acquired data):=
!direction of rotation := CW
start angle := 0.0
Radius := 65
!matrix size [1] :=128
!scaling factor (mm/pixel) [1] := 3.32
!matrix size [2] := 32
!scaling factor (mm/pixel) [2] := 6.64
!END OF INTERFILE :=
```

**Figure A6.** Example of a STIR interfile for a SPECT reconstruction.

On the other hand, a parameter file for the reconstruction algorithm must be written in interfile-like syntax, specifying the inputs and output files, the matrix used, and specific reconstruction parameters for each algorithm. An example of file for each algorithm considered in the project is seen: In *Figure A7*, the file for a FBP2D filter, in *Figure A8*, for FBP3D filter, in *Figure A9*, for OSEM2D algorithm and in *Figure A10* for OSEM3D algorithm.

```
fbp2dparameters := :=

input file := /NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/stir_sinogram.hs
output filename prefix := /NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/recons_FBP2D

zoom := 0.5
xy output image size (in pixels) := 128

alpha parameter for ramp filter := 1
cut-off for ramp filter (in cycles) := 0.5

Back Projector type:= matrix
Back Projector Using Matrix Parameters :=
Matrix type := SPECT UB
Projection Matrix By Bin SPECT UB Parameters:=psf type:= Geometrical
attenuation type := No
mask type := Cylinder
keep all views in cache := 0
End Projection Matrix By Bin SPECT UB Parameters:=
End Back Projector Using Matrix Parameters :=
end:=
```

**Figure A7.** Example of STIR file to apply a FBP2D reconstruction

```
fbp3drpparameters :=

input file := /NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/stir_sinogram.hs
output filename prefix := /NFSROOT/kerrighed/home/biofisica/simCris/STIR_3.1/install/bin/FBP3DRP
;;;;; output image parameterszoom := 1
; defaults to cover whole FOV
xy output image size (in pixels) := 180
maximum absolute segment number to process := 2
;;;;; parameters for initial imagenum segments to combine with ssrb := -1

alpha parameter for ramp filter := 1
cut-off for ramp filter (in cycles) := 0.4

alpha parameter for colsher filter in axial direction := 1
cut-off for colsher filter in axial direction (in cycles) := 0.5
alpha parameter for colsher filter in planar direction := 1
cut-off for colsher filter in planar direction (in cycles) := 0.5

stretch factor for colsher filter definition in axial direction := 2
stretch factor for colsher filter definition in planar direction := 2

transaxial extension for fft := 1
axial extension for fft := 1

save intermediate images := 0
display level := 0

end :=
```

**Figure A8.** Example of STIR file to apply a FBP3D reconstruction

```
OSMAPOSParameters :=
objective function type:= PoissonLogLikelihoodWithLinearModelForMeanAndProjData
PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
input file := /NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/stir_sinogram.hs
projector pair type := Matrix

Projector Pair Using Matrix Parameters :=
Matrix type := SPECT UB

Projection Matrix By Bin SPECT UB Parameters:=
minimum weight:= 0.001
maximum number of sigmas:= 2.0
spatial resolution PSF:= 0.00001
psf type:= 2D
collimator slope := 0.0163
collimator sigma 0(cm) := 0.1466
attenuation type := Simple
keep all views in cache:=1

End Projection Matrix By Bin SPECT UB Parameters:=
End Projector Pair Using Matrix Parameters :=
end PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
number of subsets:= 8
number of subiterations:= 10
Save estimates at subiteration intervals:= 10

END :=
```

**Figure A9.** Example of STIR file to apply a OSEM2D reconstruction

```
OSMAPOSParameters :=
objective function type:= PoissonLogLikelihoodWithLinearModelForMeanAndProjData
PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
input file := /NFSROOT/kerrighed/home/biofisica/simCris/Patients/P1/Results/stir_sinogram.hs
projector pair type := Matrix

Projector Pair Using Matrix Parameters :=
Matrix type := SPECT UB

Projection Matrix By Bin SPECT UB Parameters:=
minimum weight:= 0.001
maximum number of sigmas:= 2.0
spatial resolution PSF:= 0.00001
psf type:= 3D
collimator slope := 0.0163
collimator sigma 0(cm) := 0.1466
attenuation type := Simple
keep all views in cache:=1

End Projection Matrix By Bin SPECT UB Parameters:=
End Projector Pair Using Matrix Parameters :=
end PoissonLogLikelihoodWithLinearModelForMeanAndProjData Parameters:=
number of subsets:= 8
number of subiterations:= 10
Save estimates at subiteration intervals:= 10

END :=
```

**Figure A10.** Example of STIR file to apply a OSEM3D reconstruction



## ANNEX 3 Installation and User guide

SPECTSim is an open-source software to perform Monte Carlo simulations to generate SPECT data. It uses SimSET package to perform simulations and STIR package to reconstruct the resulting simulated data.

By now, the project is not completely finished, as some other modules would like to be implemented. However, to try its advances and be able to perform some modifications an installation guide and user guide have been developed.

At the moment, the project only works with Linux/Unix/Ubuntu systems.

### DEPENDENCIES NEEDED

As SPECTSim uses external software packages, they need to be correctly installed.

#### 1. Install SimSET

To install SimSET, follow the instructions available at:

[https://depts.washington.edu/simset/old\\_websites/html.2.9.1/installation\\_guide/installation\\_guide\\_index.html](https://depts.washington.edu/simset/old_websites/html.2.9.1/installation_guide/installation_guide_index.html)

Version used in this project is **2.6.2.5** but is also compatible with other versions.

#### 2. Install STIR

To install STIR, follow the instructions available at:

[https://stir.sourceforge.net/wiki/index.php/Installing\\_STIR\\_with\\_CMake](https://stir.sourceforge.net/wiki/index.php/Installing_STIR_with_CMake)

Version used in this project is **3.1** but other versions should work.

**Install both SimSET and STIR packages in the folder the SPECTSim repository is going to be cloned**

## INSTALLATION

### 1) Clone SPECTSim repository

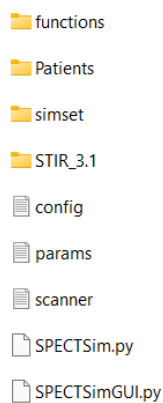
The project is available on a GitHub repository. Clone it in the directory is preferred to work.

- a) Open the Ubuntu terminal
- b) Move
- c) Make sure an Ubuntu Git installation exists, if not, type in the console:  

```
sudo apt install git
```
- d) When git installed, type in the console  

```
git clone https://github.com/crmartinb/SPECTSim.git
```

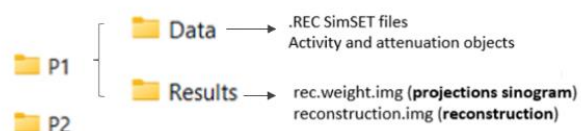
### 2) Folders organisation



To run SPECTSim correctly, the selected directory must have the folders organised like seen in the next figure. *SimSET* and *STIR3.1* folders correspond to the ones created when installing each package. *Functions* folder is the one with the functions of the SPECTSim software, the main skeleton of the program. *Patients'* directory is the one where all data and results stored for each study generated.

### 3) Creating *Patients* folder

To function, SPECTSim needs a patient's folder where the data and results from each case. In the following figure, the organisation of data that must be followed is shown. The "Patients" directory must be named like this, the environment patient directory name may be changed, "Data" and "Results" folder must also be named after these.



## USER MANUAL

### 1) Specify simulation options

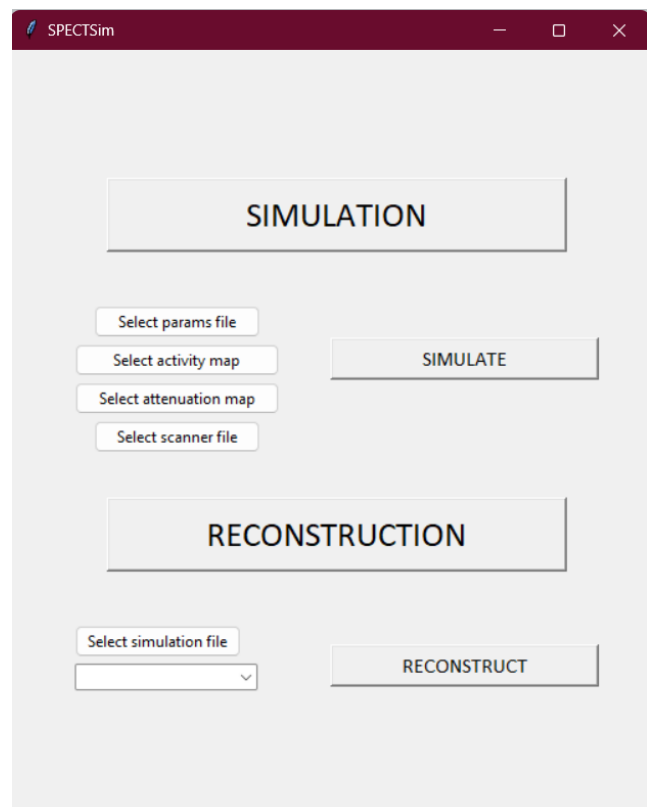
Change the wanted parameters to change simulation options. Change the parameter “input\_dir” to the name of your patient folder. Preferrable only edit the number of photons to simulate, the scan duration and the energy of the photons.

### 2) Running SPECTSim

To run SPECTSim program you have two options: run it executing a python code specifying the files and parameters or by using the software.

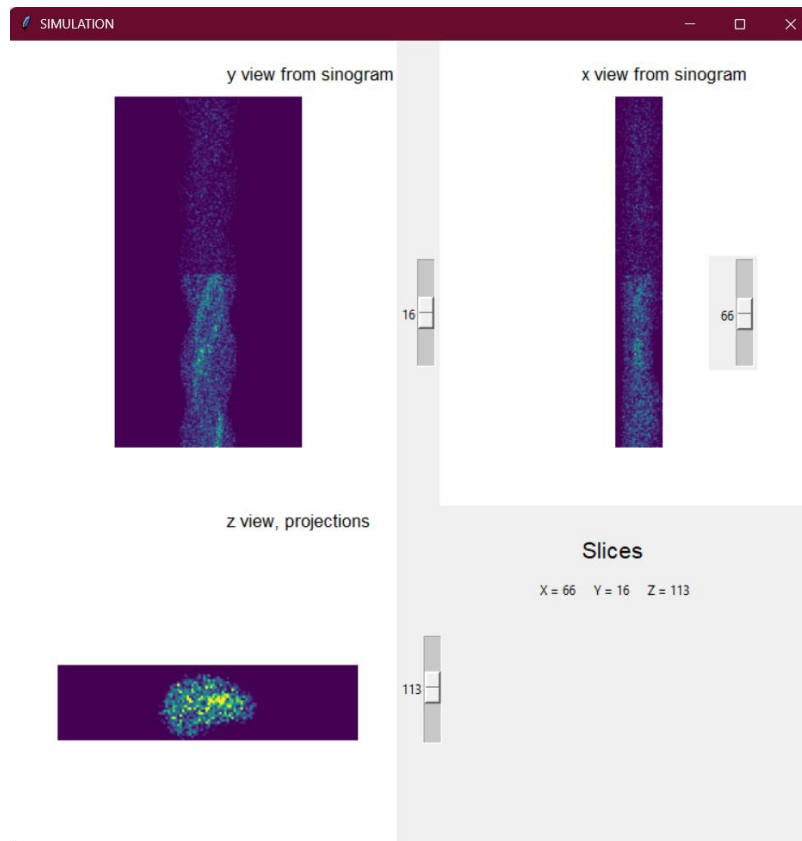
#### 2.1 - Through the software...

The software looks like seen in the picture. As it may be seen, all the buttons specified in previous section appear. In the top, there are the buttons to control the simulation flow, and in the bottom the buttons to control the reconstruction flow appear.

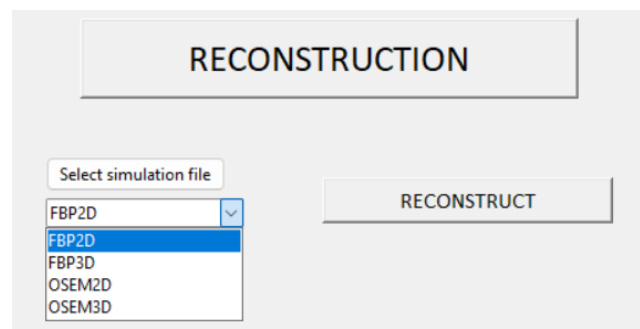




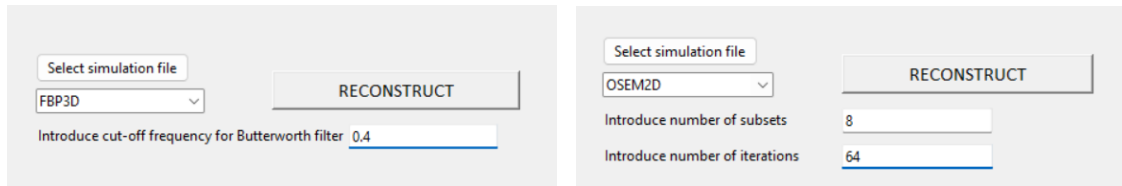
In the simulation pop-up window showing the results of the Monte Carlo simulation for the input parameters is seen. Top left, the y view from the sinogram is seen, observing both the real and scatter sinograms, top right, the x view from the sinogram is visualised. Left bottom, the projections (z view) are seen. At each side of the three images, the slider to control the slice shown is also seen.



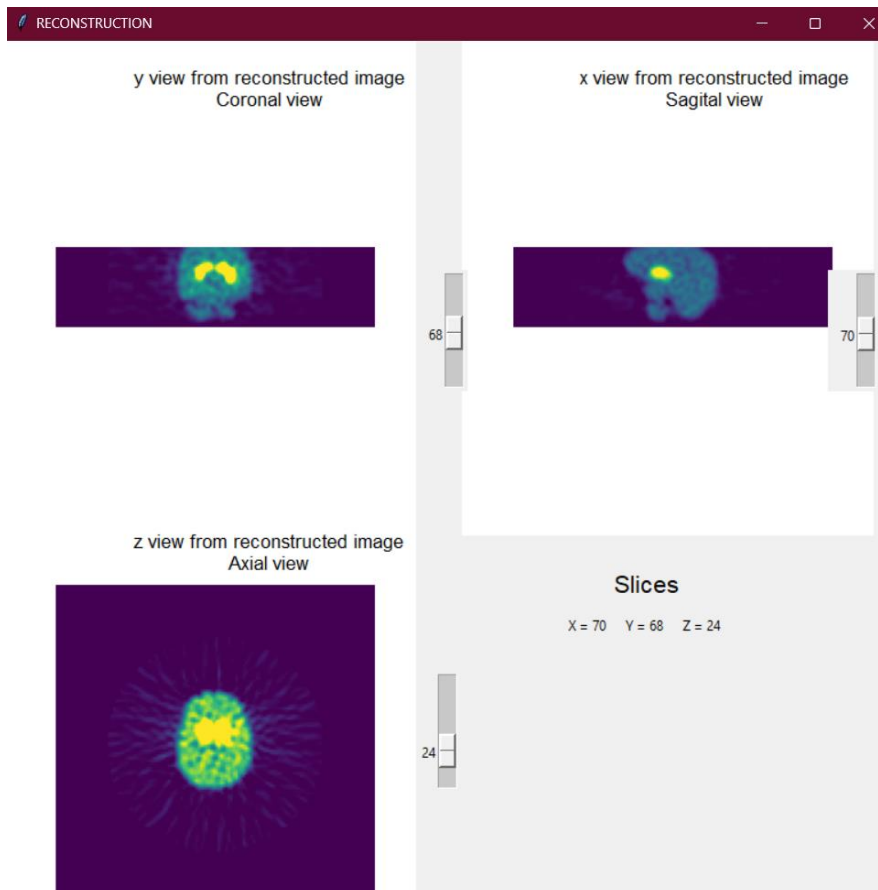
Then, when the "Reconstruction" option is chosen, and the simulation file selected (must be the output from the simulation process called *simset\_sinogram.hdr*), the list to choose the reconstruction algorithm appears



When reconstruction algorithm selected, you have to enter the desired reconstruction parameters for the algorithm. The parameters are For **FBP2D** and **FBP3D**: Cut off for ramp filter and or **OSEM2D** and **OSEM3D**: Number of subsets and number of iterations



In the reconstruction pop-up window showing results for the chosen reconstruction algorithm and specific views. The organisation of the displayed image is top left–coronal view, top right–sagittal view, and left bottom–axial view, following the CSA display. At each side of the images, the slider to control the slice shown is seen.



## 2.2 – Through a python file...

To run the simulation and reconstruction processes through a python file, it must be generated in the working directory (the main one) following the structure presented below.

```
import SPECTSim
from os.path import join
import os

cd=os.getcwd()
params_file=join(cd,"params.yml")
scanner_file=join(cd,"scanner.yml")
act_map=join(cd, "Patients", "P1", "act")
att_map=join(cd, "Patients", "P1", "att")

recons_type='OSEM2D'

# flux control, if recons type is FBP, subsets and iterations will be written as 0
# the same for OSEM filters, the cut off frequency will be written as 0
if recons_type == "FBP2D" or recons_type == "FBP3D":
    cutoff = 0.4
    num_subsets = 0
    num_iterations = 0
elif recons_type == "OSEM2D" or recons_type == "OSEM3D":
    cutoff = 0
    num_subsets = 8
    num_iterations = 10

simu = SPECTSim.SimsSPECT(params_file, scanner_file, act_map, att_map, recons_type, cutoff, num_subsets, num_iterations)
simu.run_simu()

recons = SPECTSim.SimsSPECT(params_file, scanner_file, act_map, att_map, recons_type, cutoff, num_subsets, num_iterations)
recons.run_recons()
```

Then, type in the terminal the following command:

```
python name_python_file.py
```

Where *name\_python\_file* file is the name under the file is saved.

**You will find the results in the Results directory generated**