Katarzyna Grygorowicz 307346

Kamera wirtualna – sprawozdanie

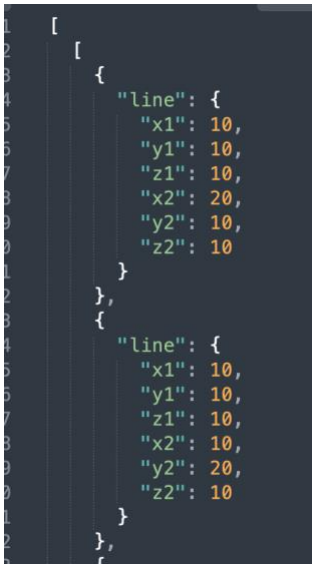
Kod: https://github.com/kasiagrygorowicz/virtual-camera

Cel projektu

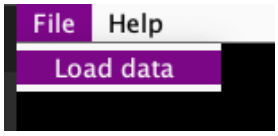
Celem projektu było zaimplementowanie wirtualnie kamery poruszającej się w przestrzeni 3d i pozwalające na obserwacje obiektów w tej przestrzeni. Kamera umożliwia:

- Poruszanie się w przestrzeni wzdłuż osi
- Zmianę przybliżenia obiektów
- Obracanie

Dane wejściowe



Po uruchomieniu programu należy wybrać plik z danymi opisującymi obiekty. Plik musi być w formacie JSON. Zdecydowałam się na JSONa ze względu na jego czytelność. Każdy obiekt opisywany jest przez listę krawędzi, które go tworzą. Na krawędź składają się 2 punkty. Aby wybrać plik z danymi wybieramy w menu opcję **File**, a potem **Load data**:



Zastosowane wzory

Rzut perspektywiczny

Aby punkty 3D przenieść na płaszczyznę 2D należało wykonać rzutowanie perspektywiczne względem punktu rzutowania jakim jest kamera.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{d} & 0 \end{bmatrix} = \begin{bmatrix} \frac{xd}{z} \\ \frac{yd}{z} \\ d \\ 1 \end{bmatrix}$$

Zastosowanie tych wzorów pozwala uzyskać znormalizowany wynik.

Katarzyna Grygorowicz 307346

Zoom

Dla operacji zoomowania nie trzeba wykonywać transformacji punktów, ale trzeba zmienić wartość ogniskowej, która w kodzie ma nazwę **D**.

Translacja

Program ma zdefiniowaną wartość kroku o jaki punkty w osach x, y lub z zostaną przesunięte.

$$\begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x+X \\ y+Y \\ z+Z \\ 1 \end{bmatrix}$$

Rotacja

Obrót kamery następuje poprzez zmianę pozycji każdego wierzchołka o zdefiniowaną wartość kroku.

Rotacja w osi X

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\phi & -sin\phi & 0 \\ 0 & sin\phi & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ ycos\phi - zsin\phi \\ ysin\phi + zcos\phi \\ 1 \end{bmatrix}$$

Rotacja w osi Y

$$\begin{bmatrix} cos\phi & 0 & sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -sin\phi & 0 & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} xcos\phi + zsin\phi \\ y \\ -xsin\phi + zcos\phi \\ 1 \end{bmatrix}$$

Rotacja w osi Z

$$\begin{bmatrix} cos\phi & -sin\phi & 0 & 0 \\ sin\phi & cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} xcos\phi - ysin\phi \\ xsin\phi + ycos\phi \\ z \\ 1 \end{bmatrix}$$

Katarzyna Grygorowicz 307346

Działanie programu

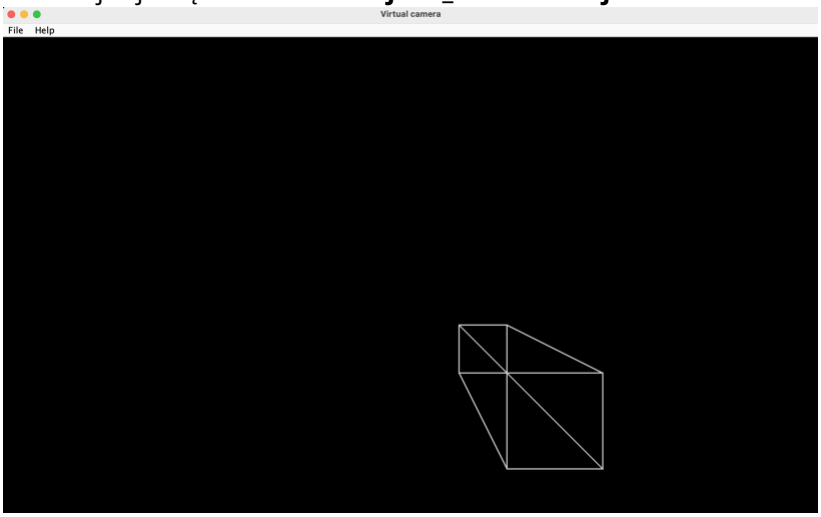
Program znajduje się w pliku **virtual-camera.jar** w folderze **/target/** w repozytorium.
Układ jest przesuwany.

Sterowanie programem

- Strzałki - translacja w osiach X i Y
- Q i W - translacja w osi Z
- I i O – przybliżenie i pomniejszenie
- F i G – rotacja w osi X (do przodu i do tyłu)
- H i J – rotacja w osi Y (w prawo i w lewo)
- K i L – rotacja w osi Z (dół-lewo i dół-prawo)

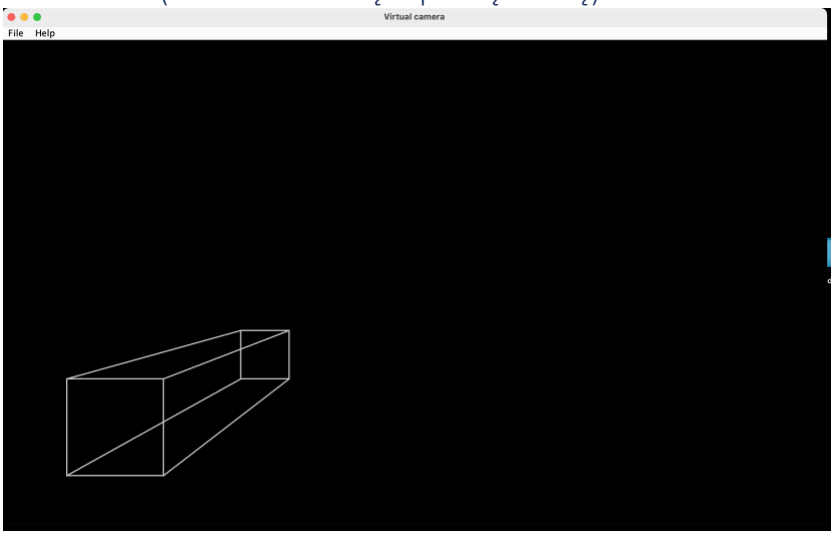
Początkowe położenie obiektu

Plik znajduje się w **/resources/json_files/data1.json**



Translacja

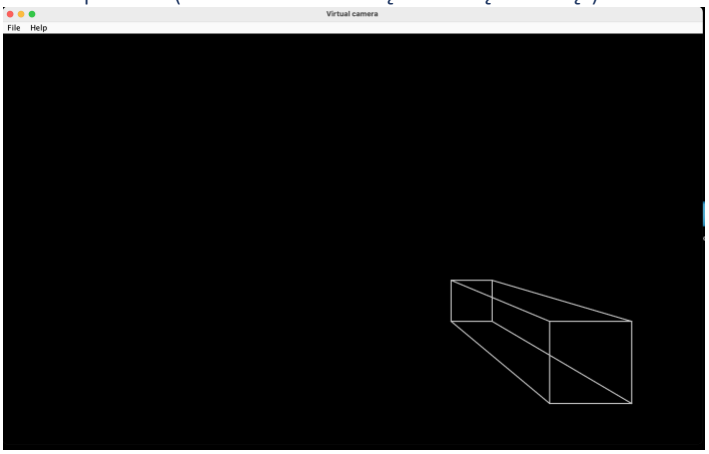
Oś X – lewo (kamera rusza się w prawą stronę)



Względem pierwotnego ustawienia widać, że przy przesunięciu kamery w prawo odsłania się prawa strona obiektu.

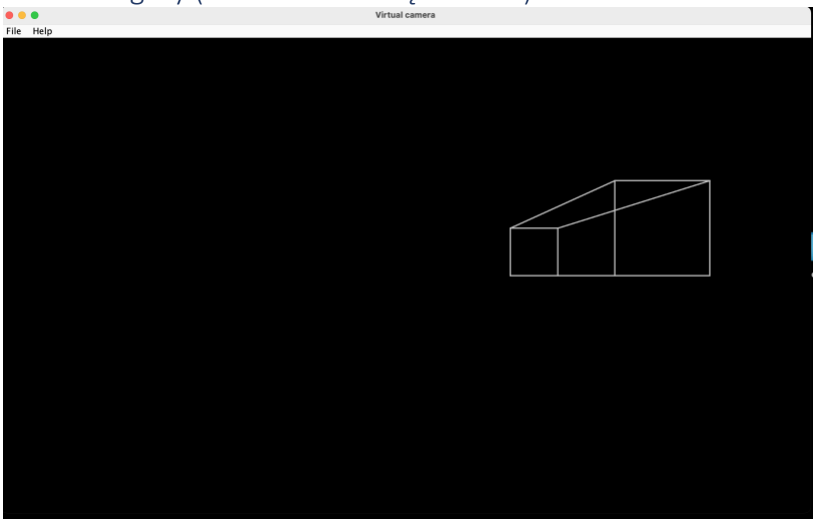
Katarzyna Grygorowicz 307346

Oś X –prawo (kamera rusza się w lewą stronę)



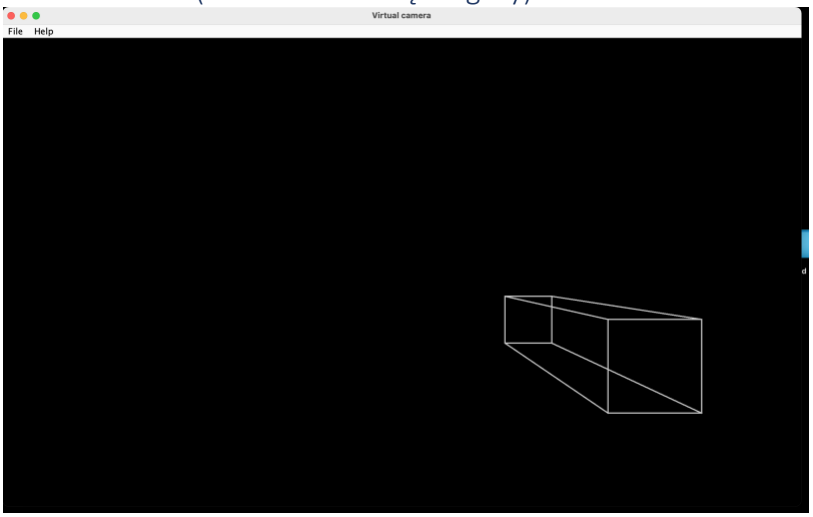
Przesuniętą kamerę w prawo przesuwam w lewo i widać że odsłania się lewa strona bryły.

Oś Y – do góry (kamera rusza się do dołu)



Przesunięcie kamery do dołu względem wyjściowej pozycji.

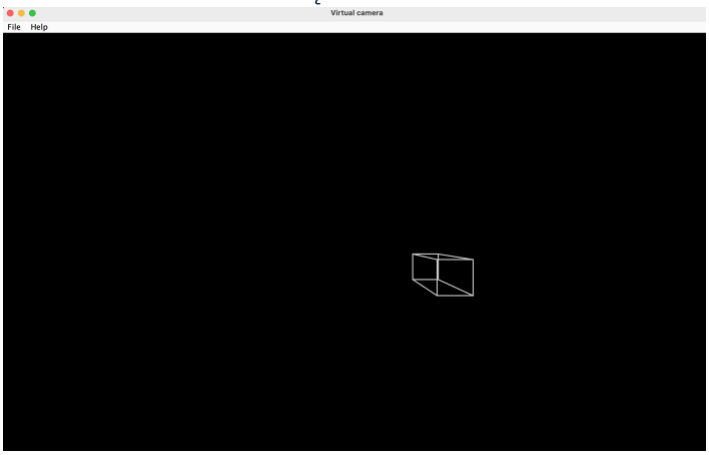
Oś Y – do dołu (kamera rusza się do góry)



Kamera zostaje ponownie przesunięta do góry i widzimy obiekt od góry.

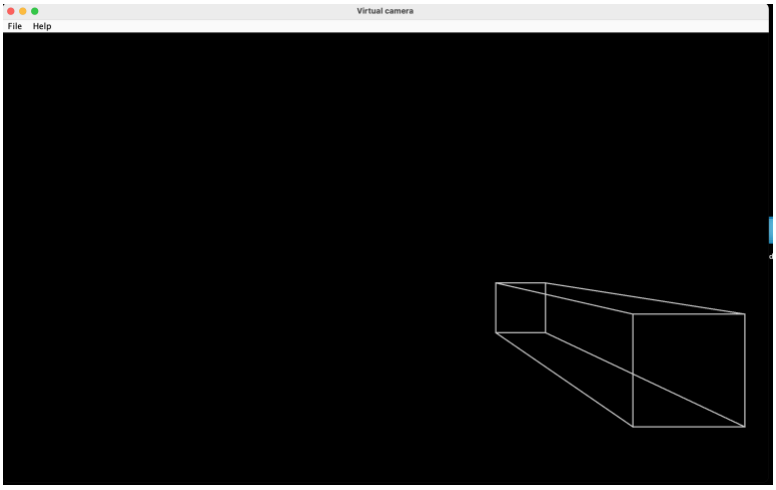
Katarzyna Grygorowicz 307346

Oś Z – kamera oddala się



Obiekt się pomniejsza.

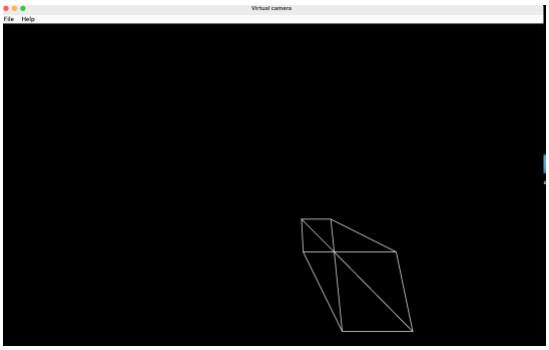
Oś Z – kamera przybliża się



Katarzyna Grygorowicz 307346

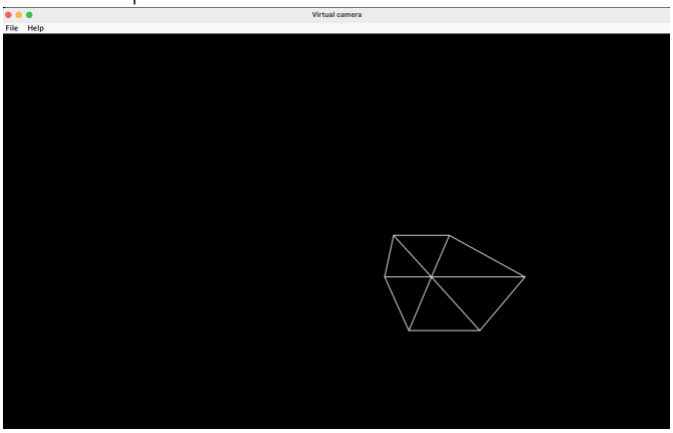
Rotacja

Oś X – do tyłu



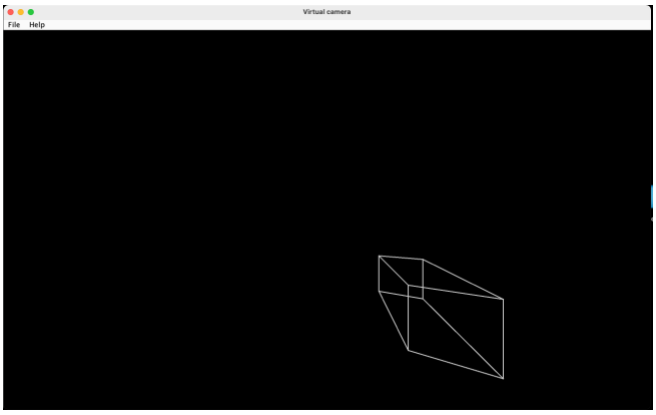
Widać wyraźnie pochylenie do tyłu obiektu względem pierwotnego położenia. Sprawia to wrażenie jakby kamera wchodziła na obiekt od góry.

Oś X – do przodu



Widać wyraźnie pochylenie do przodu obiektu względem pierwotnego położenia. Sprawia to wrażenie jakby kamera wchodziła pod obiekt od dołu.

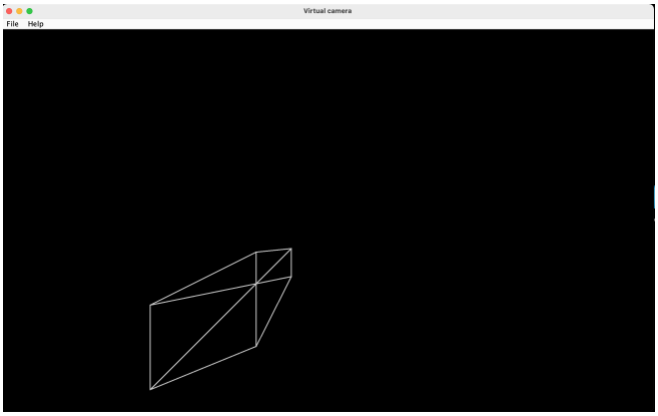
Oś Y – w prawo



Względem pierwotnego położenia widzimy, że prawa strona obiektu wydłuża się.

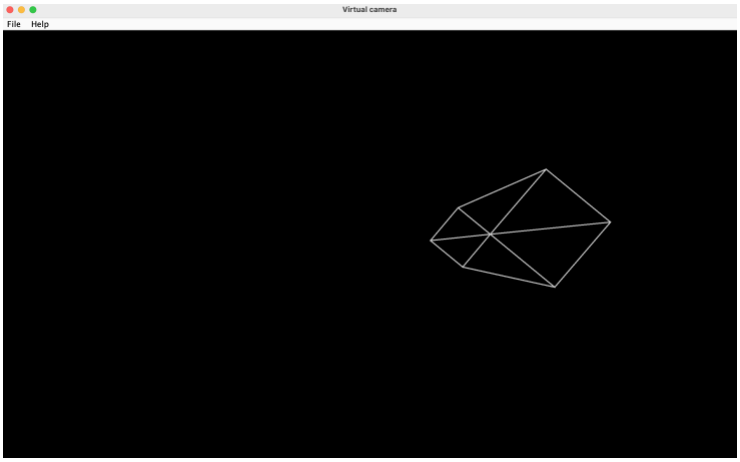
Katarzyna Grygorowicz 307346

Oś Y – w lewo

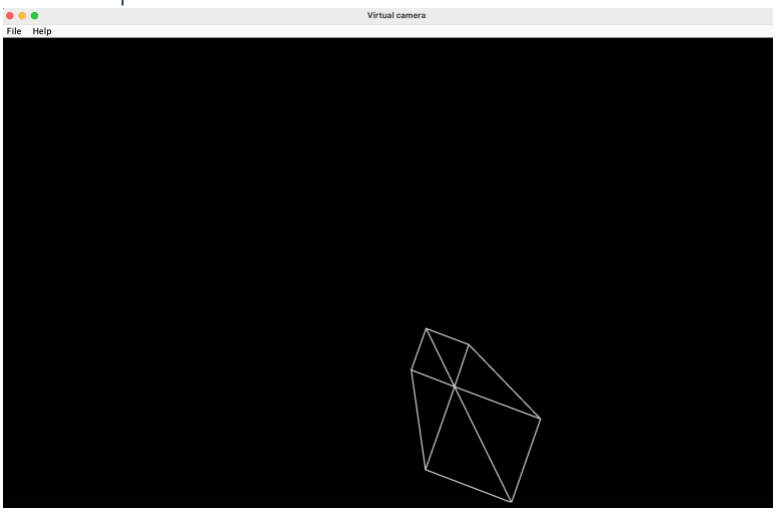


Względem pierwotnego położenia widzimy, że lewa strona obiektu wydłuża się.

Oś Z – w lewo

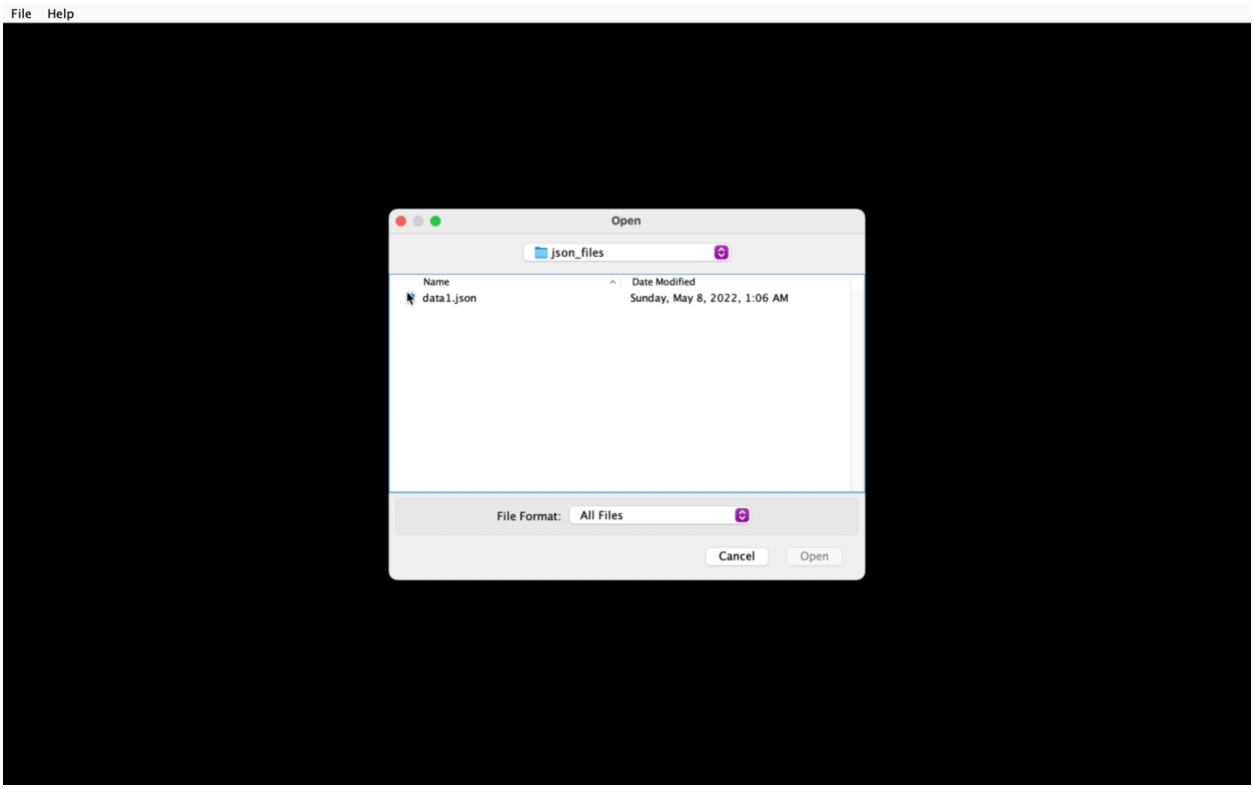


Oś Z – w prawo



Katarzyna Grygorowicz 307346

Nagranie



Technologie

Java, json.org, lombok, slf4j