

PLANET GENERATOR

Projekt końcowy kursu SDA Data Science

ZESPÓŁ



TOMASZ MULARCZYK

Data Scientist Wannabe



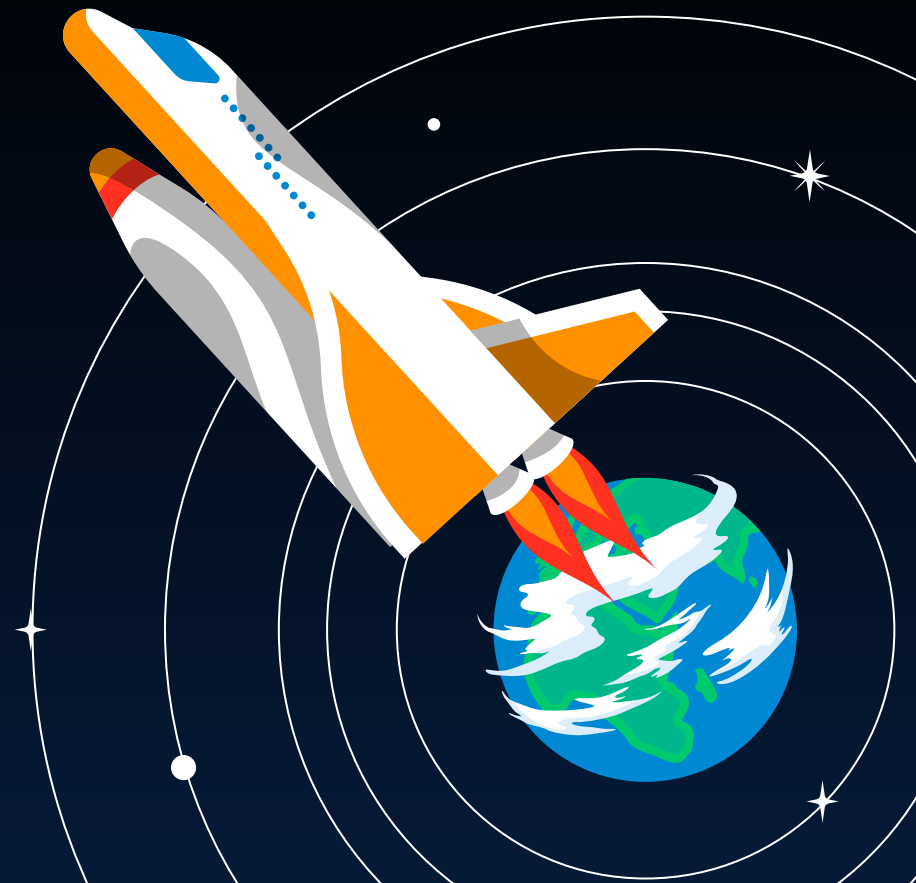
PIOTR PAWŁOWSKI

Data Scientist Wannabe

CEL PROJEKTU

Na potrzeby gry typu ekspansja kosmosu:

Stworzenie generatora nowych planet do osiągnięcia
efektu Otwartego Świata tzw. OpenWorld



CO CHCEMY UZYSKAĆ I JAKI EFEKT OCZEKUJEMY

1.

Stworzenie Generatora AI

do przetestowania wystawionego API



2.

Stworzenie API

do odpytywania o klasyfikację danego obiektu, bądź wygenerowanie wizualizacji nowej planety

3.

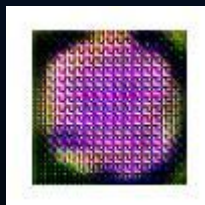
Stworzenie interfejsu użytkownika

do przetestowania wystawionego API



DZIAŁANIE SIECI GAN

Dane generowane



Discriminator

FAKE

REAL

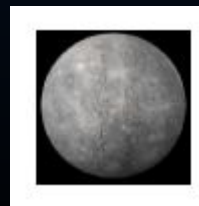
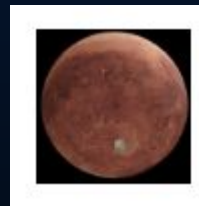
FAKE

REAL

FAKE?

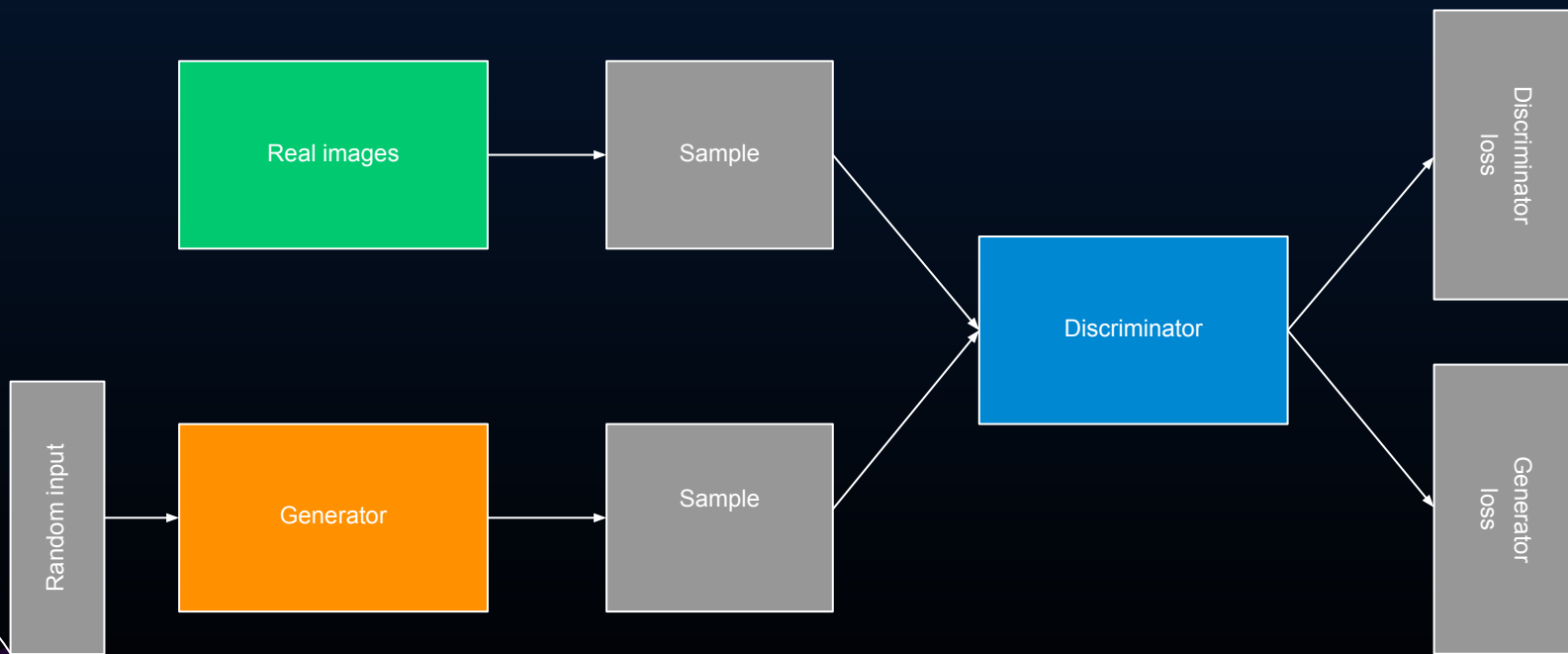
REAL

Dane prawdziwe



DZIAŁANIE SIECI GAN

Schemat działania



TIMELINE PROJEKTU

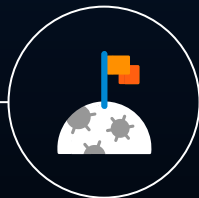
21.01

- Zebranie danych
- Obróbka surowych danych



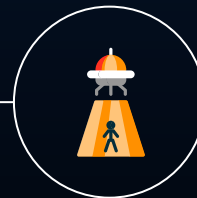
22.01-29.01

- Wytrenowanie modelu
- Utworzenie API
- Utworzenie strony internetowej generatora



22.01

- Zaplanowanie pracy
- Wczytanie danych
- Utworzenie pierwszej sieci neuronowej



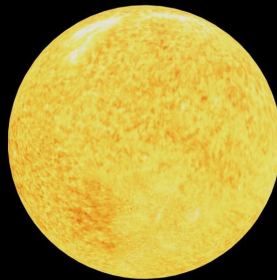
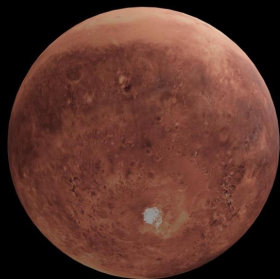
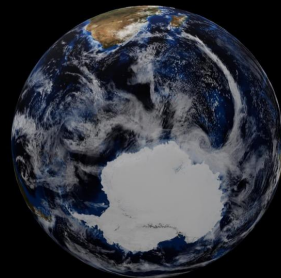
29.01-03.02

- Podsumowanie projektu
- Modelowanie 512px Datasetu

DATA UNDERSTANDING

DANE WEJŚCIOWE

- Jakość kontra ilość
- Wykorzystanie setów zdjęciowych ze strony NASA
- Nasz wstępnie przygotowany dataset obejmuje 100 zdjęć dla każdej z planety (x12)



DATA PREPARATION

DANE WEJŚCIOWE

- Jakość kontra ilość
- Wykorzystanie setów zdjęciowych ze strony NASA
- Nasz wstępnie przygotowany dataset obejmuje 100 zdjęć dla każdej z planety (x12)

Screengrab

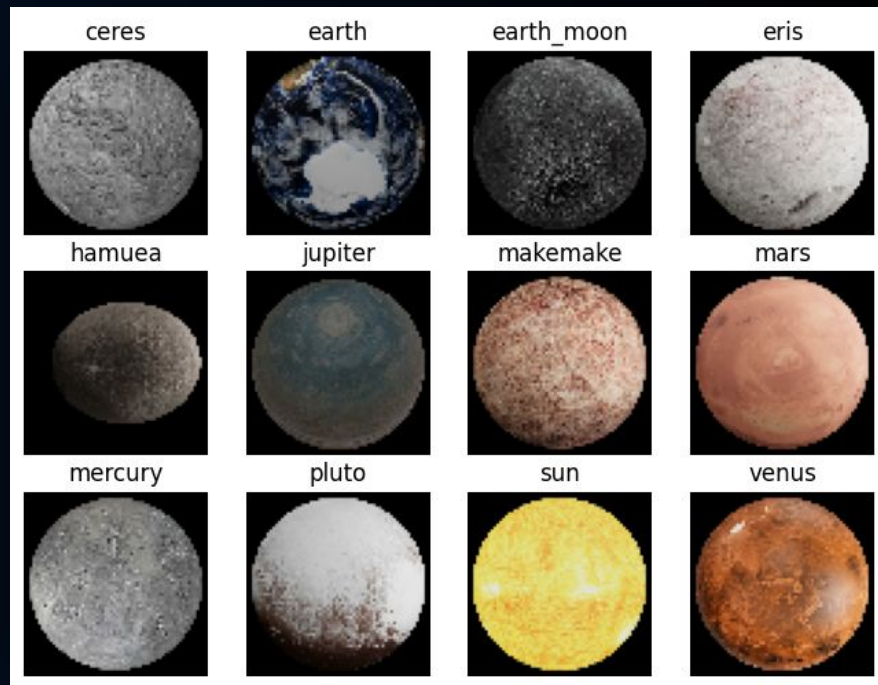
→
resize
remove watermarks
rename

Batch
processing

DATA PREPARATION

DANE WEJŚCIOWE

- Jakość kontra ilość
- Wykorzystanie setów zdjęciowych ze strony NASA
- Nasz wstępnie przygotowany dataset obejmuje 100 zdjęć dla każdej z planety (x12)
- **Wstępnie przygotowaliśmy scaling do 64px, 128px, 256px, 512 px**



MODELING

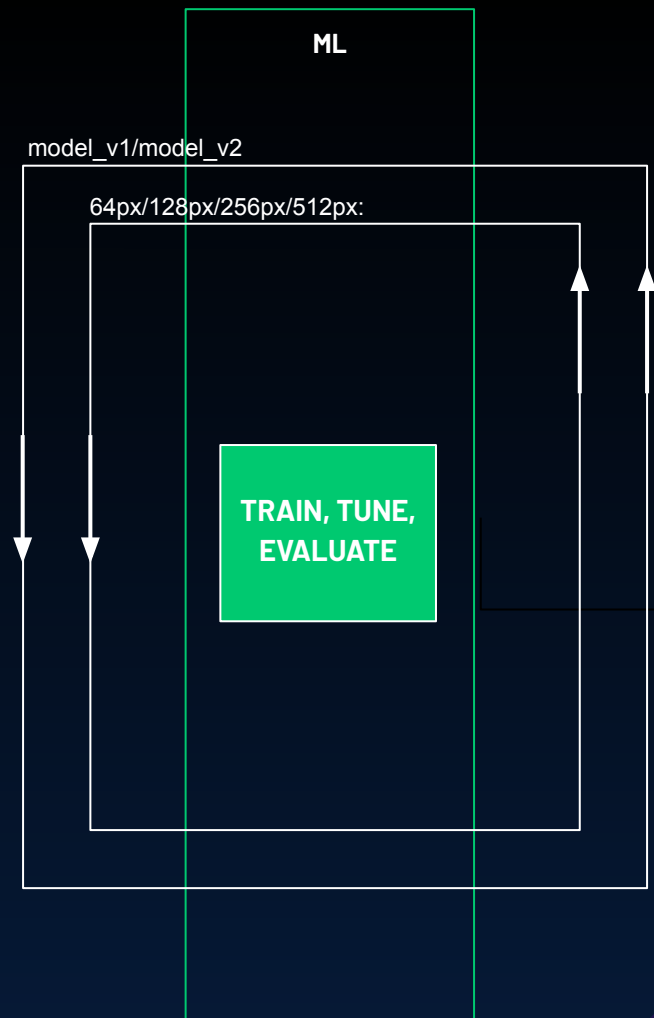
- Utworzono 8 niezależnych modeli (po 2 na każdą rozdzielczość)
- Każdy model składa się z generatora i dyskryminatora; generator z 4 warstw, dyskryminator z 3 (w tym po dwie konwolucyjne)
- model_v2 wyróżnia się większą liczbą zastosowanych filtrów
- Dla modelu 512px ograniczyliśmy ilość danych wejściowych do 4 (z 12) planet

model_v1:

Gen: 4L, 64/32F, 7mln params
Disc: 3L, 32/32CNN, 18k params

model_v2:

Gen: 4L, 128/64F, 440mln params
Disc: 3L, 64/128CNN, 2mln params



EVALUATION

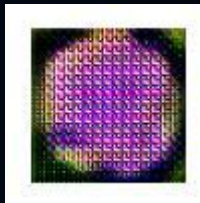
- 8 przygotowanych modeli wytrenowano na 3 równolegle pracujących maszynach wirtualnych pracujących w chmurze Azure
- Wraz z rozdzielczością zdjęć i ilością parametrów modelu wzrastała zasobożerność (czas, ram, storage)
- Przetwarzanie modelu o największej rozdzielczości i tylko dla 4 planet trwało ok. 5 dni
-

Data set 512x512px, model_v2

Czas liczenia jednej epoki ~ 20 min;



1 dzień, 10 epoka



3 dzień, 250 epoka



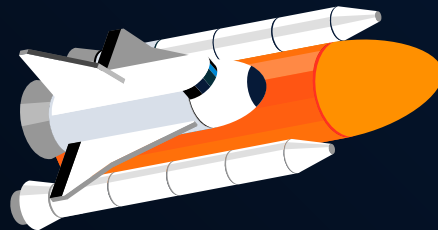
5 dzień, 500 epoka

DEPLOYMENT



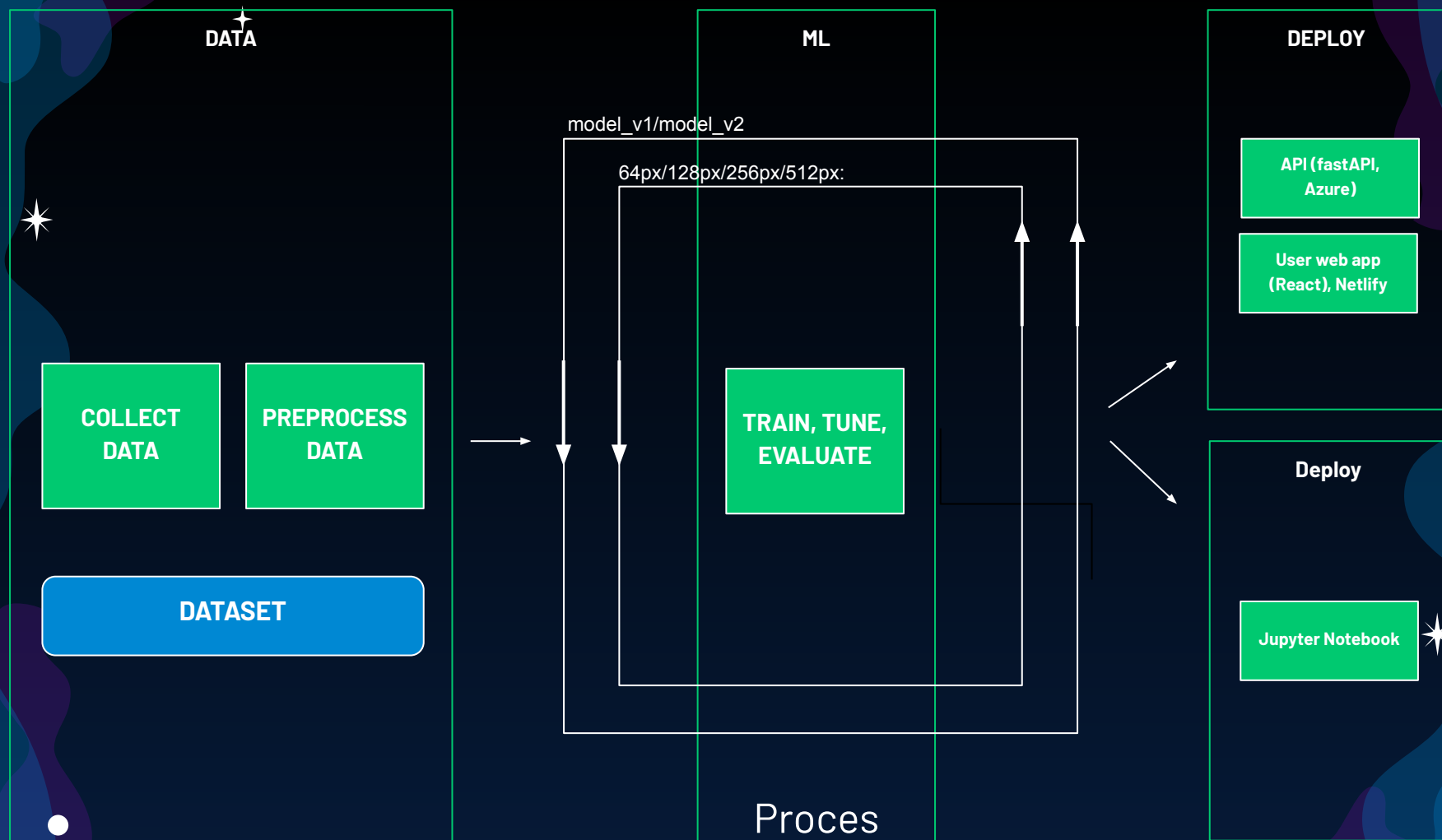
MODEL (API)

Deploy modelu GAN generatora umieszczono w serwisie FastAPI, **endpoint generuje nową planetę**. Wykorzystano platformę Azure.



WEB APP

Layout wykonano w Figma. Gotowy generator (**front aplikacji testującej API**) zbudowano w React i umieszczono na publicznym serwerze.



HARDWARE



Windows 2019 Server
8vPC, 32GB Ram

DATASET

64px

128px

256px

512px

MODELS

Model_v1:

Gen: 4L, 64/32F, 7mln params
Disc: 3L, 32/32CNN, 18k params

Model_v2:

Gen: 4L, 128/64F, 440mln params
Disc: 3L, 64/128CNN, 2mln params

DCGAN

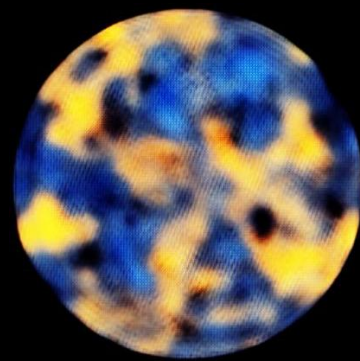
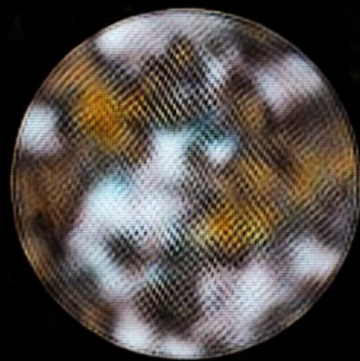
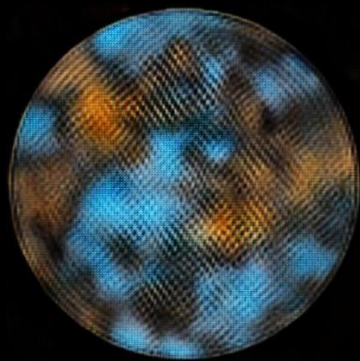
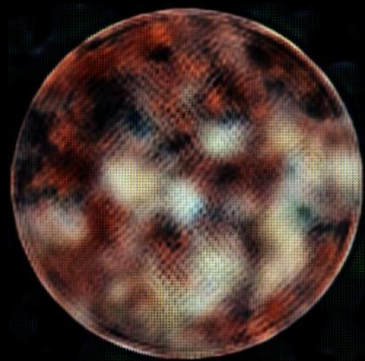
DEPLOY

API (fastAPI,
Azure)

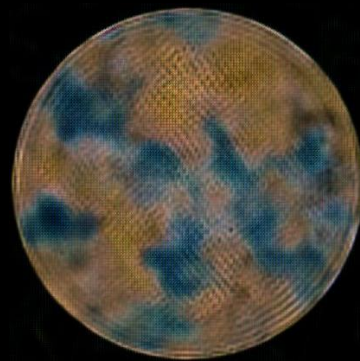
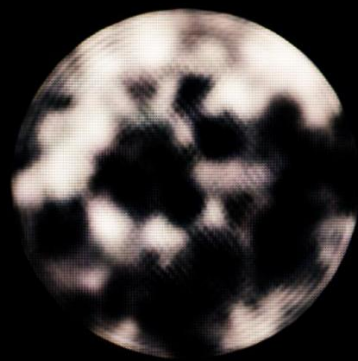
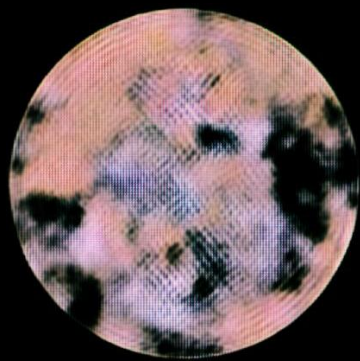
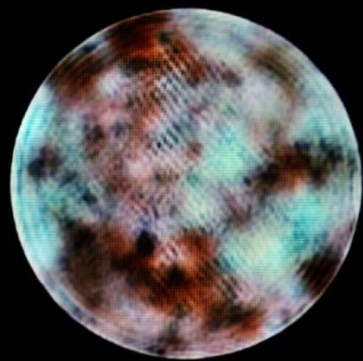
User web app
(React), Netlify

Jupyter Notebook

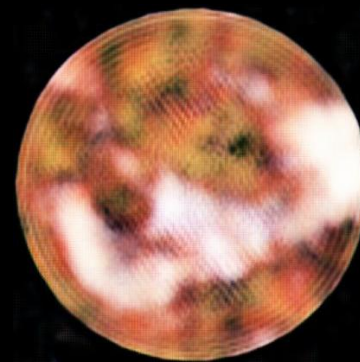
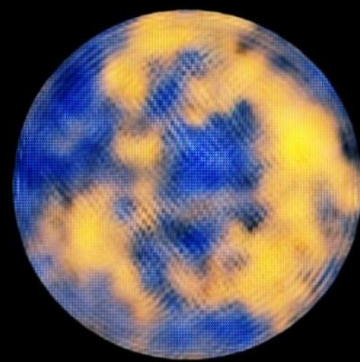
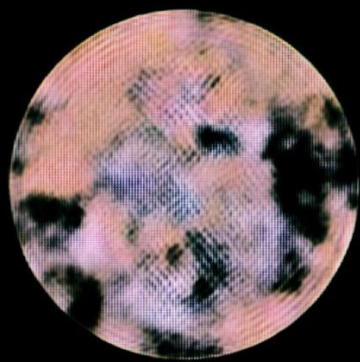
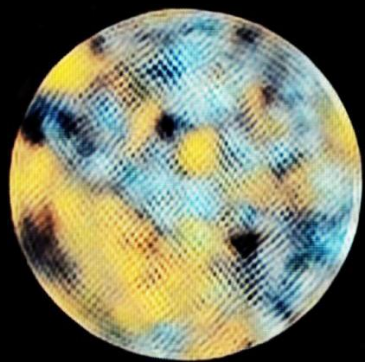
Infrastruktura



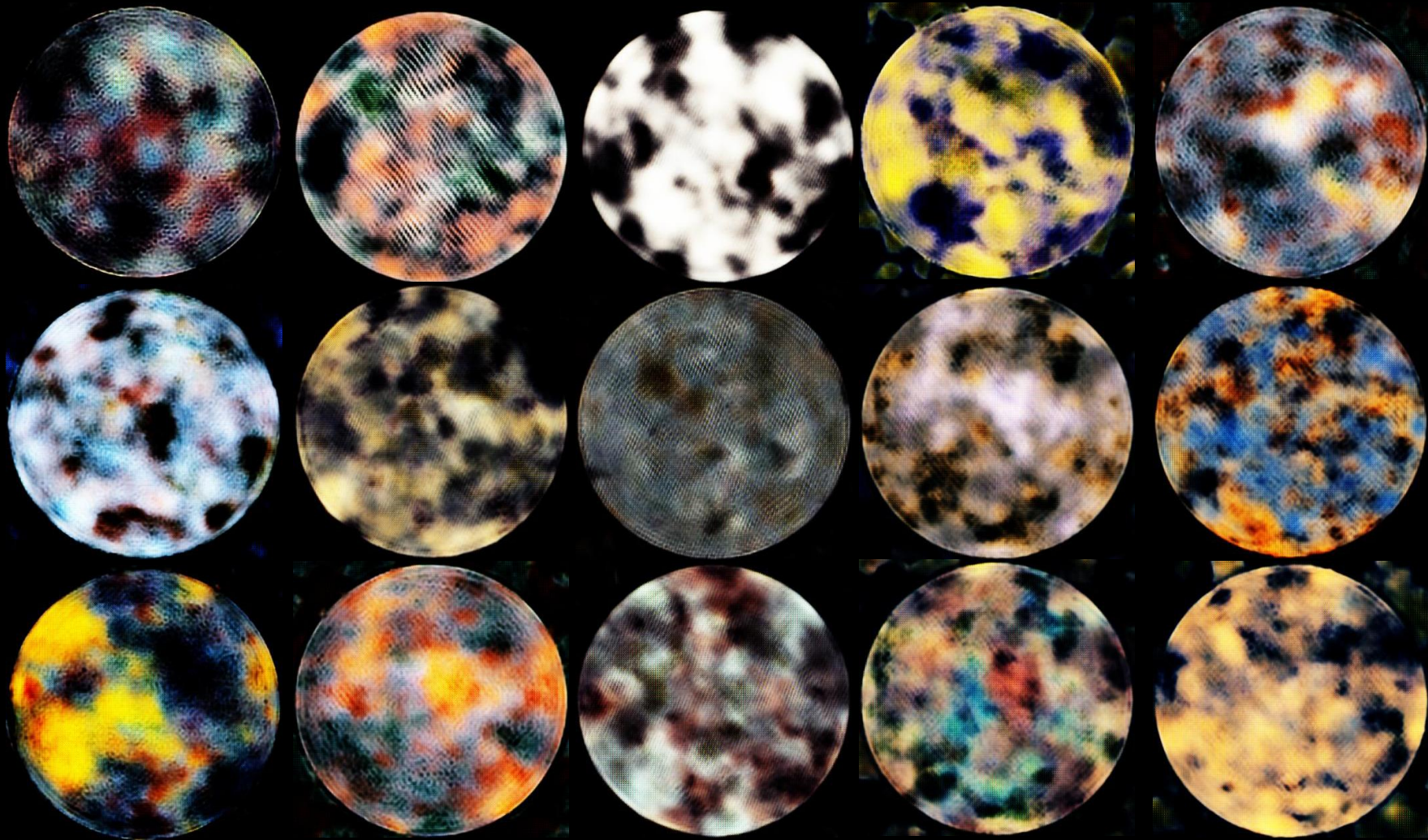
Wygenerowane planety



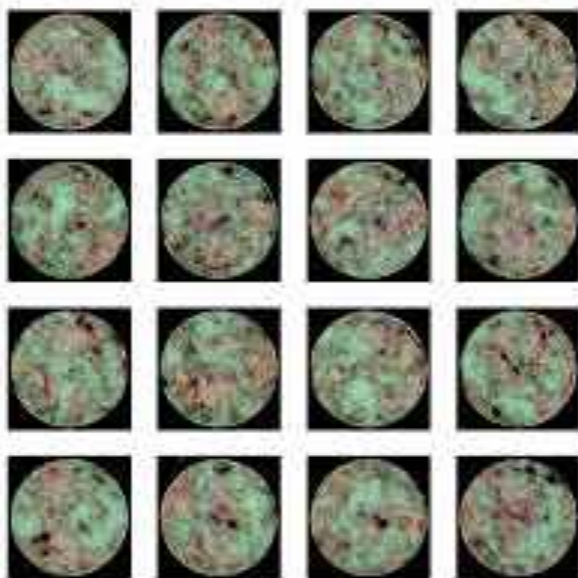
Wygenerowane planety

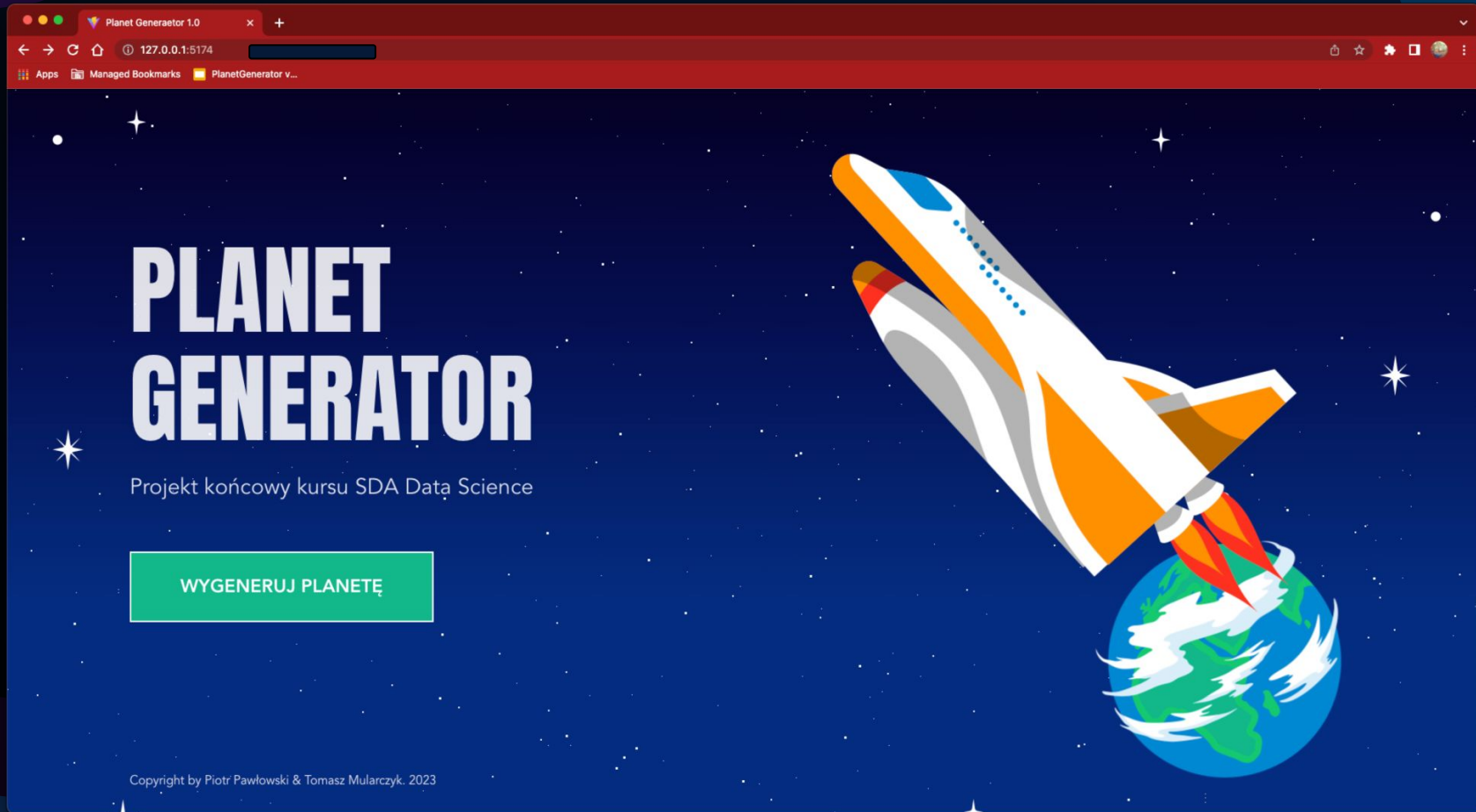


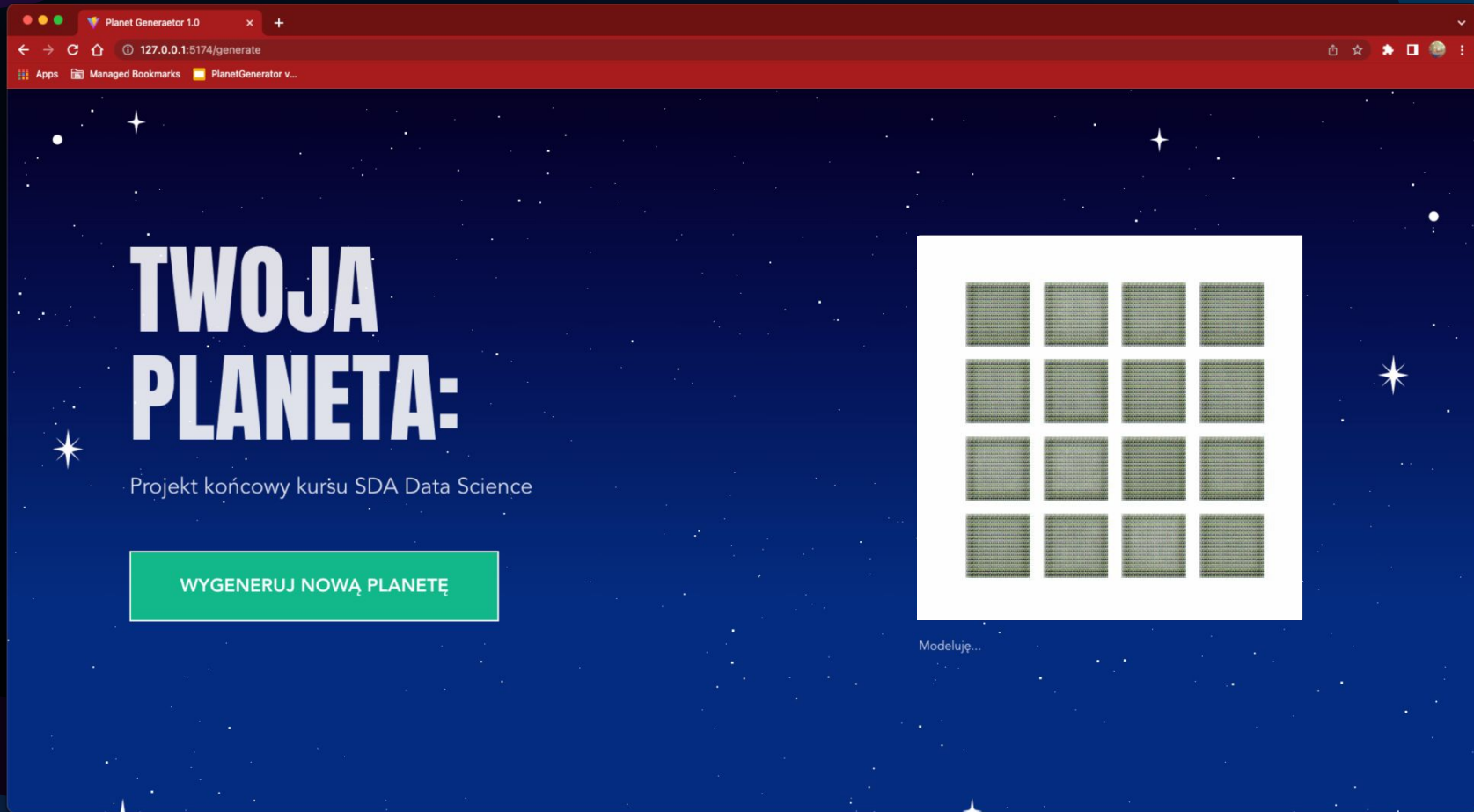
Wygenerowane planety

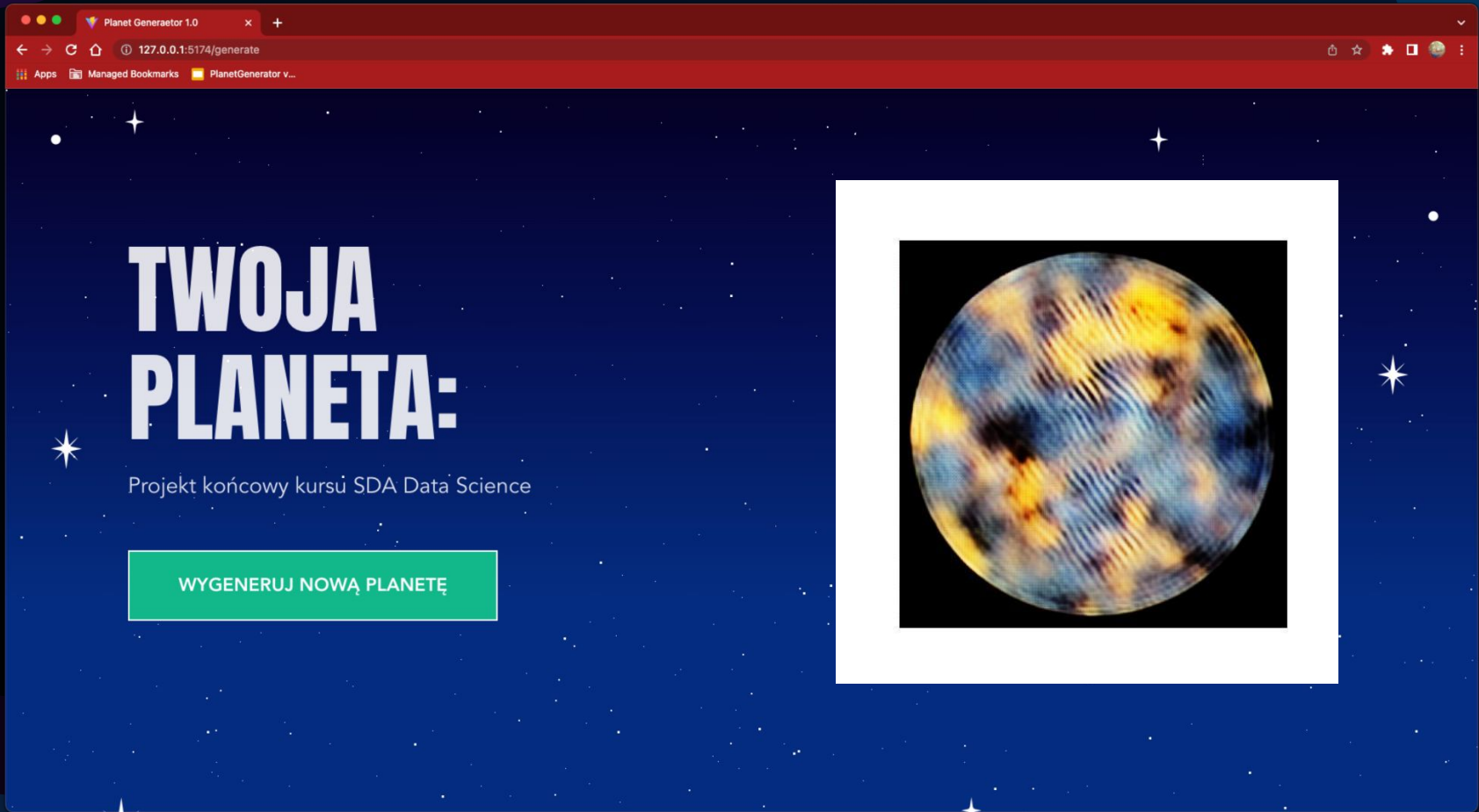












ŹRÓDŁA:

- Aplikacja: <http://tomsoft1.pl>
- Github: [mulatom1/PlanetGenerator](https://github.com/mulatom1/PlanetGenerator)
- Projekt aplikacji: [figma.com](https://www.figma.com)

