

## Индивидуальное задание

### Работа с библиотекой *TensorAeroSpace*

#### Шаги работы:

1. Выполнить *Fork* библиотеки в свою учетную запись.
2. Изучить материалы открытой библиотеки.
3. Провести установку библиотеки на свой локальный ПК.
4. Провести апробацию (например, с использованием представленных в репозитории примеров).

#### Порядок работы:

1. Выполнил *Fork* проекта в свою учетную запись, как показано на рисунках 1-2.

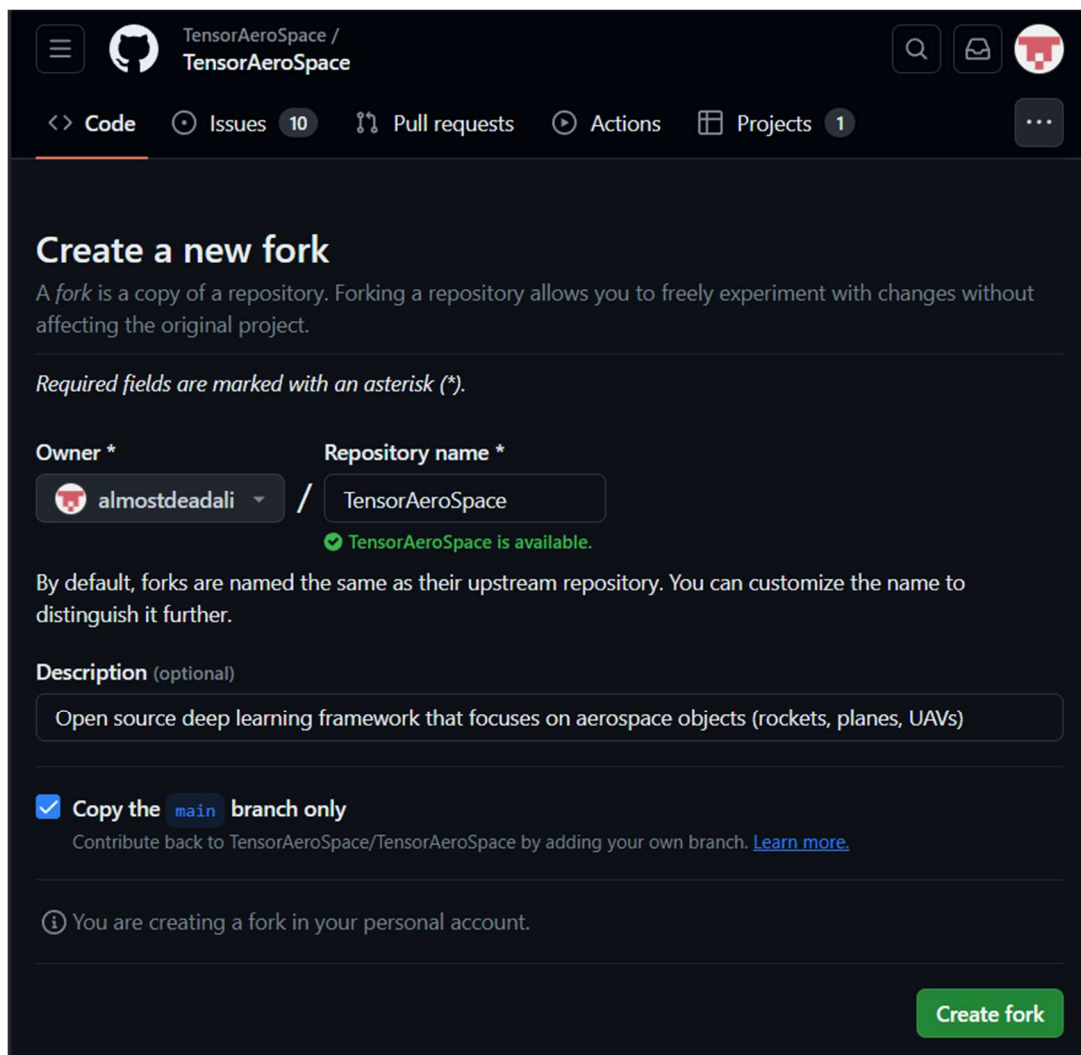


Рисунок 1 – *Fork* библиотеки

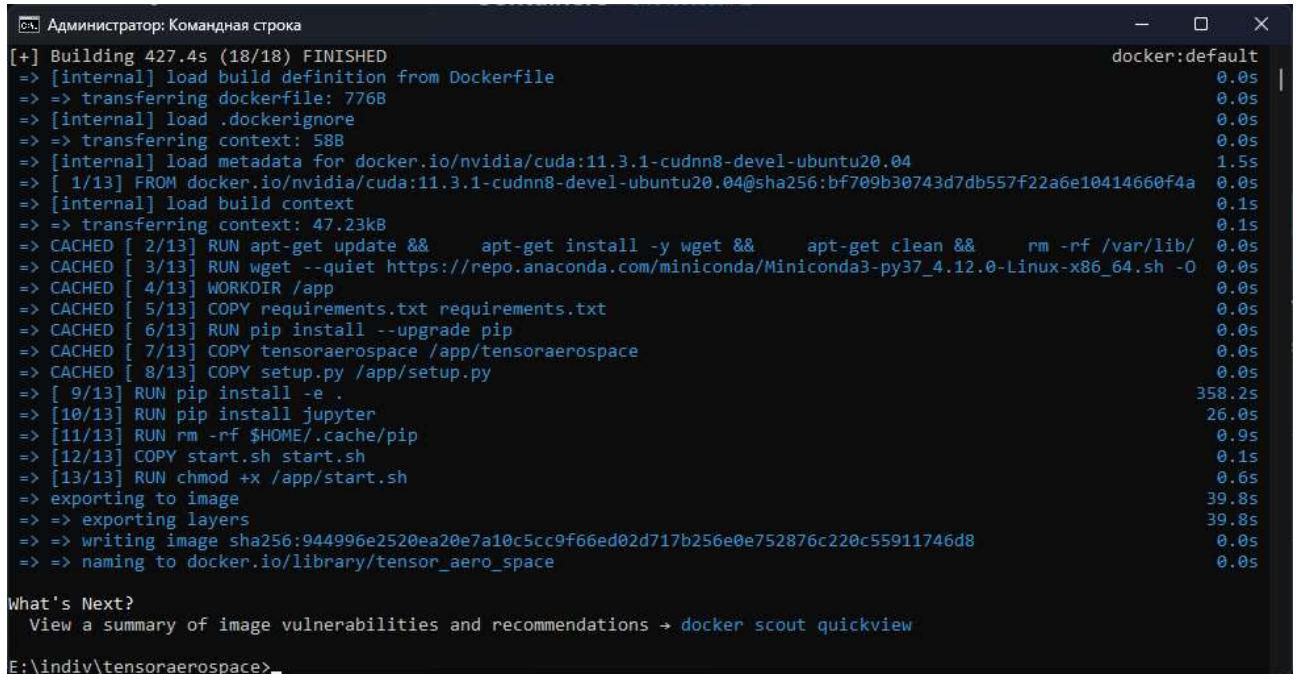


3. Установил библиотеку на свой локальный ПК, как показано на рисунке 4.

Successfully installed tensoraerospace

Рисунок 4 – Установка библиотеки на ПК

Запустил сборку проекта через *Docker*, как показано на рисунке 5.



```
Администратор: Командная строка
[+] Building 427.4s (18/18) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 776B
=> [internal] load .dockerignore
=> => transferring context: 58B
=> [internal] load metadata for docker.io/nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04
=> [ 1/13] FROM docker.io/nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04@sha256:bf709b30743d7db557f22a6e10414660f4a
=> [internal] load build context
=> => transferring context: 47.23kB
=> CACHED [ 2/13] RUN apt-get update && apt-get install -y wget && apt-get clean && rm -rf /var/lib/
=> CACHED [ 3/13] RUN wget --quiet https://repo.anaconda.com/miniconda/Miniconda3-py37_4.12.0-Linux-x86_64.sh -O
=> CACHED [ 4/13] WORKDIR /app
=> CACHED [ 5/13] COPY requirements.txt requirements.txt
=> CACHED [ 6/13] RUN pip install --upgrade pip
=> CACHED [ 7/13] COPY tensoraerospace /app/tensoraerospace
=> CACHED [ 8/13] COPY setup.py /app/setup.py
=> [ 9/13] RUN pip install -e .
=> [10/13] RUN pip install jupyter
=> [11/13] RUN rm -rf $HOME/.cache/pip
=> [12/13] COPY start.sh start.sh
=> [13/13] RUN chmod +x /app/start.sh
=> exporting to image
=> => exporting layers
=> => writing image sha256:944996e2520ea20e7a10c5cc9f66ed02d717b256e0e752876c220c55911746d8
=> => naming to docker.io/library/tensor_aero_space

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview

E:\indiv\tensoraerospace>
```

Рисунок 5 – Сборка проекта

Запускаю пример через *Docker*, как показано на рисунке 6.

```
E:\indiv>docker run -v example:/app/example -p 8888:8888 -it tensor_aero_space
exec /app/start.sh: no such file or directory
```

Рисунок 6 – Запуск примера

К сожалению, тут присутствует ошибка со стороны разработчиков. Они не добавили файл автозамены окончания строк (*CRLF* и *LF*). По итогу необходимо вручную исправлять это в *VS*. Либо можно добавить файл *.gitattributes* с автозаменой в форк проекта. После чего необходимо снова скачать библиотеку и произвести сборку проекта, как показано на рисунке 7.

```
Администратор: Командная строка - docker run -v example:/app/example -p 8888:8888 -it tensor_aero_space
=> => transferring context: 58B 0.0s
=> [internal] load build definition from Dockerfile 0.1s
=> transferring dockerfile: 746B 0.0s
=> [internal] load metadata for docker.io/nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04 1.6s
=> [ 1/13] FROM docker.io/nvidia/cuda:11.3.1-cudnn8-devel-ubuntu20.04@sha256:bf709b30743d7db557f22a6e10414660f4a 0.0s
=> [internal] load build context 0.3s
=> transferring context: 4.17MB 0.2s
=> CACHED [ 2/13] RUN apt-get update && apt-get install -y wget && apt-get clean && rm -rf /var/lib/ 0.0s
=> CACHED [ 3/13] RUN wget --quiet https://repo.anaconda.com/miniconda/Miniconda3-py37_4.12.0-Linux-x86_64.sh -O 0.0s
=> CACHED [ 4/13] WORKDIR /app 0.0s
=> CACHED [ 5/13] COPY requirements.txt requirements.txt 0.0s
=> CACHED [ 6/13] RUN pip install --upgrade pip 0.0s
=> CACHED [ 7/13] COPY tensoraerospace /app/tensoraerospace 0.0s
=> CACHED [ 8/13] COPY setup.py /app/setup.py 0.0s
=> CACHED [ 9/13] RUN pip install -e . 0.0s
=> CACHED [10/13] RUN pip install jupyter 0.0s
=> CACHED [11/13] RUN rm -rf $HOME/.cache/pip 0.0s
=> [12/13] COPY start.sh start.sh 0.1s
=> [13/13] RUN chmod +x /app/start.sh 0.3s
=> exporting to image 0.1s
=> exporting layers 0.1s
=> writing image sha256:8d2799a9bc8a294bd3c01f7115d74dcd0338fa92072146c21ab680fb1bd02f33 0.0s
=> naming to docker.io/library/tensor_aero_space 0.0s

What's Next?
1. Sign in to your Docker account → docker login
2. View a summary of image vulnerabilities and recommendations → docker scout quickview

E:\obn\TensorAeroSpace>docker run -v example:/app/example -p 8888:8888 -it tensor_aero_space
```

Рисунок 7 – Запуск примера

После выполненных действий открылся *localhost*, как показано на рисунке 8.

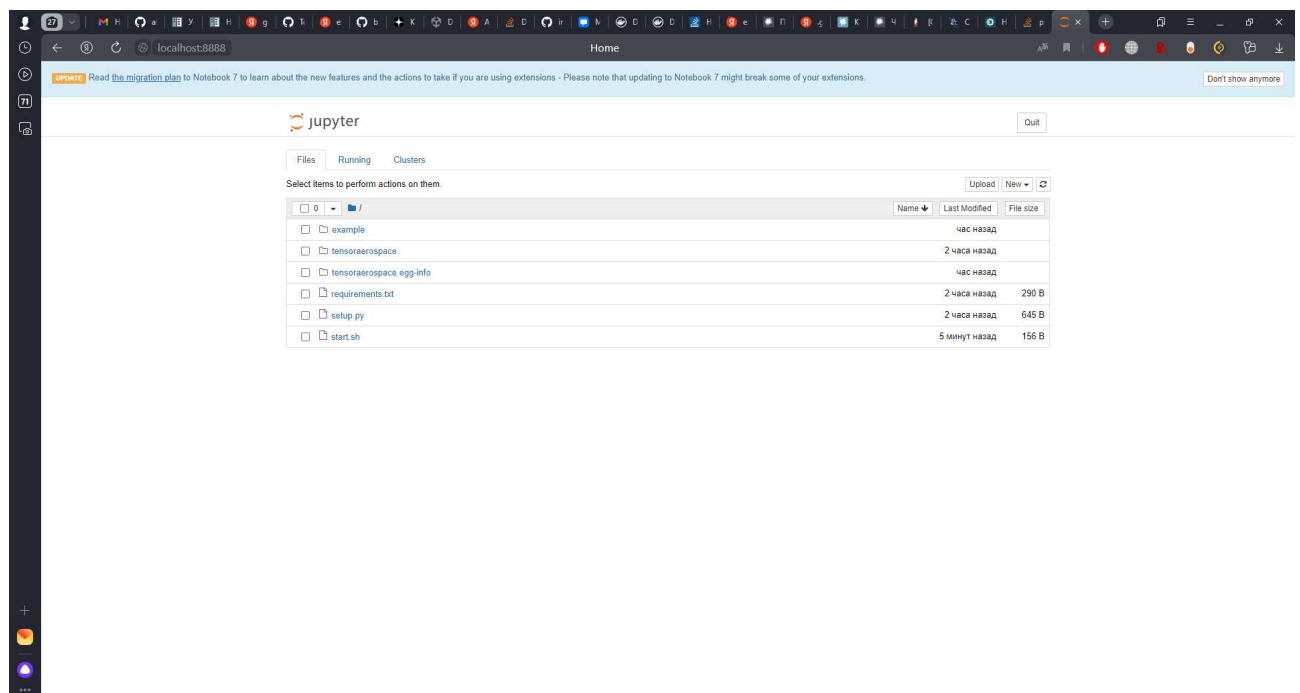


Рисунок 8 – Запуск *localhost*

Установил *UnityHub* с необходимой версией программы, как показано на рисунке 9.

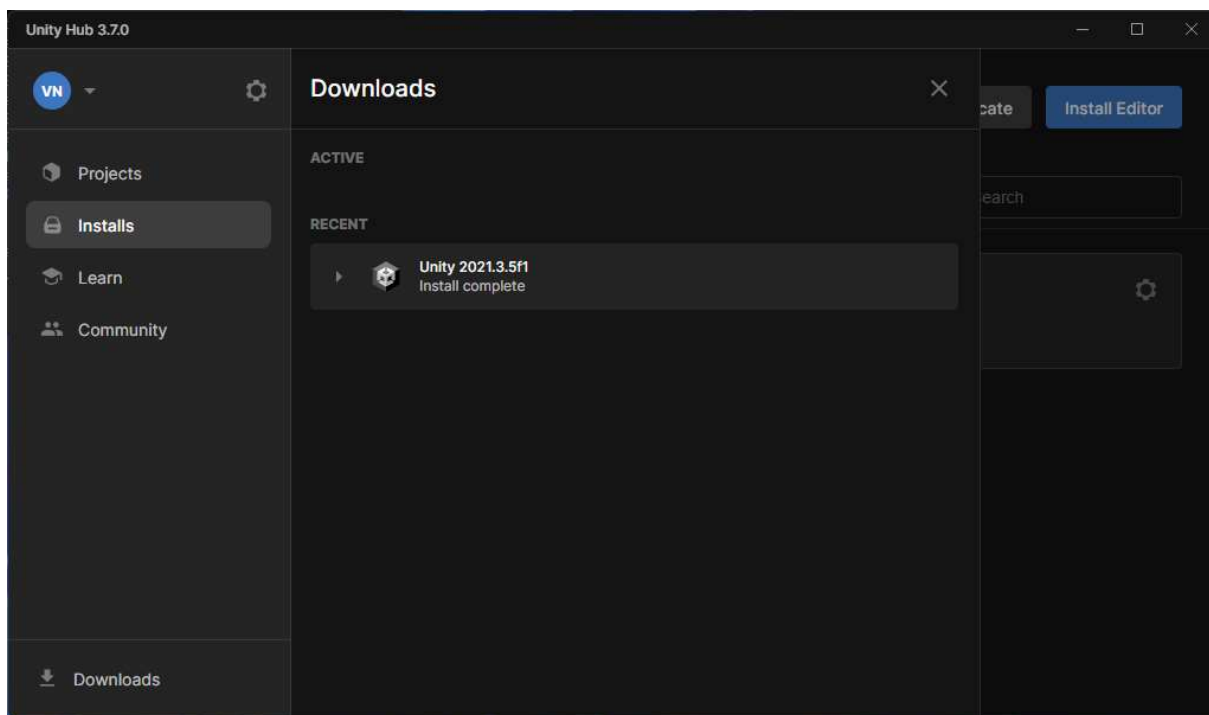


Рисунок 9 – Необходимая версия *UnityHub*

Установил *VisualStudio (VS)* с необходимыми компонентами, как показано на рисунке 10.

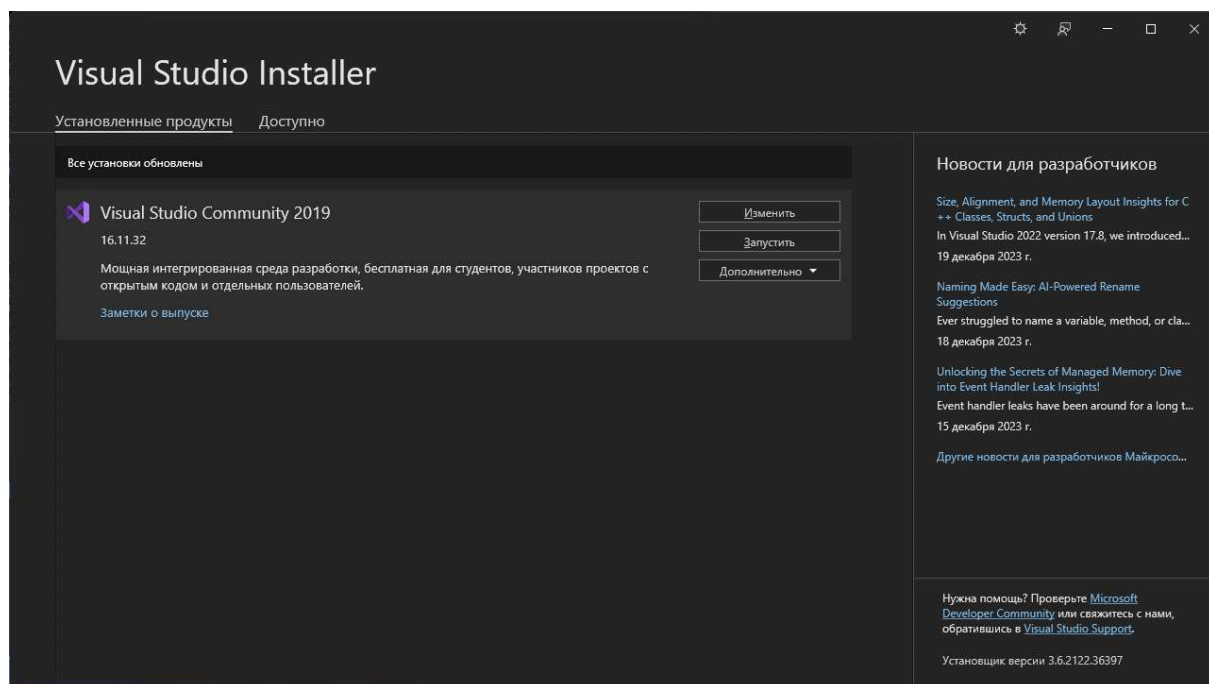


Рисунок 10 – Установленный *VS*

4. При попытке запуска проекта в *UnityHub* выскочила ошибка, как показано на рисунке 11.



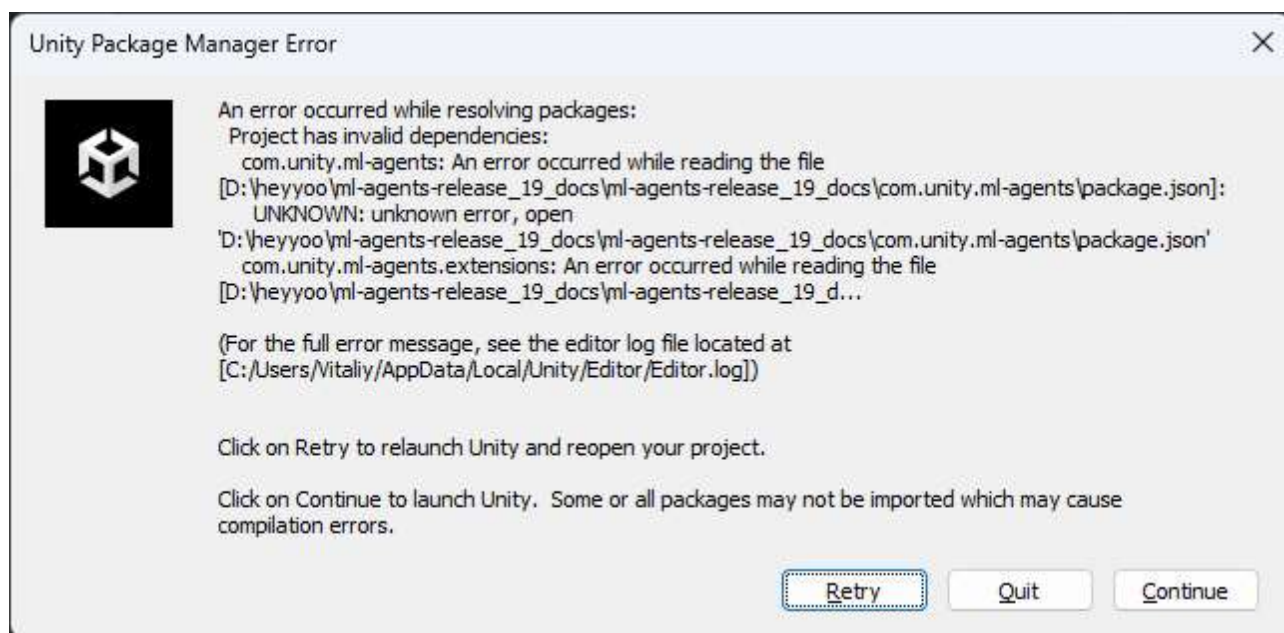


Рисунок 11 - Ошибка

По итогу смог открыть модель легкого самолета, как показано на рисунке 12.

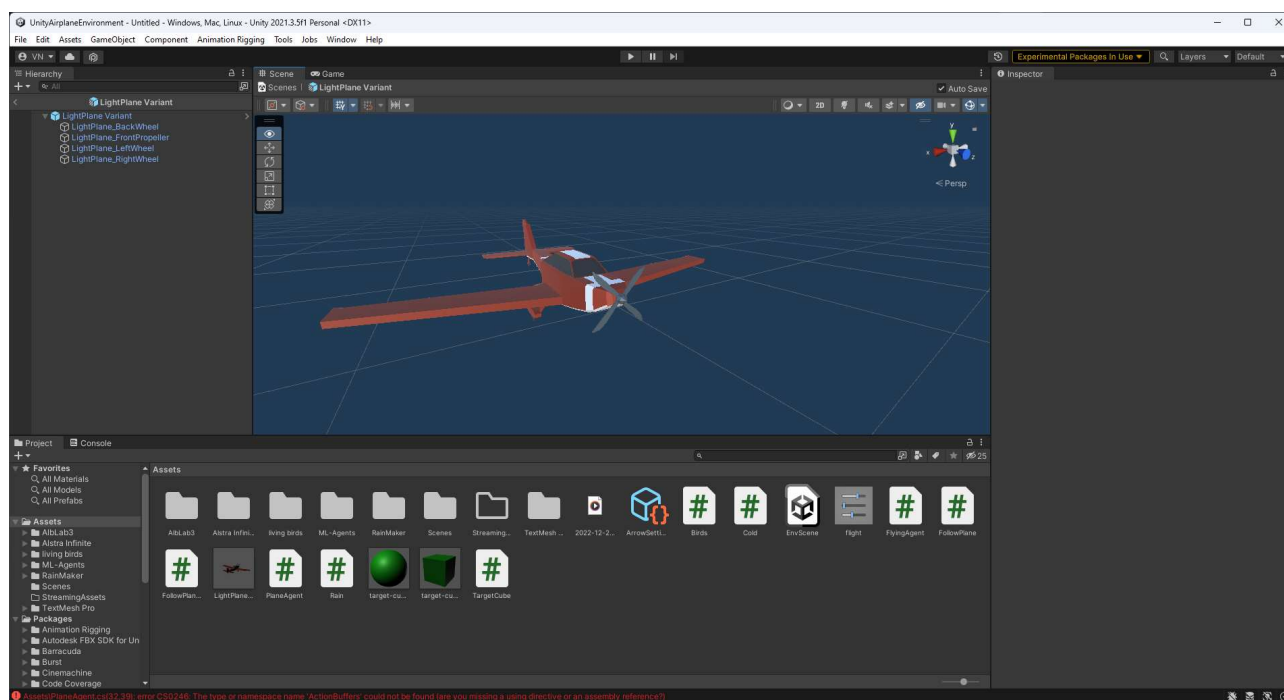


Рисунок 12 – Запущенная в *Unity* модель

Разместил второй самолет и птицу, как показано на рисунке 13.

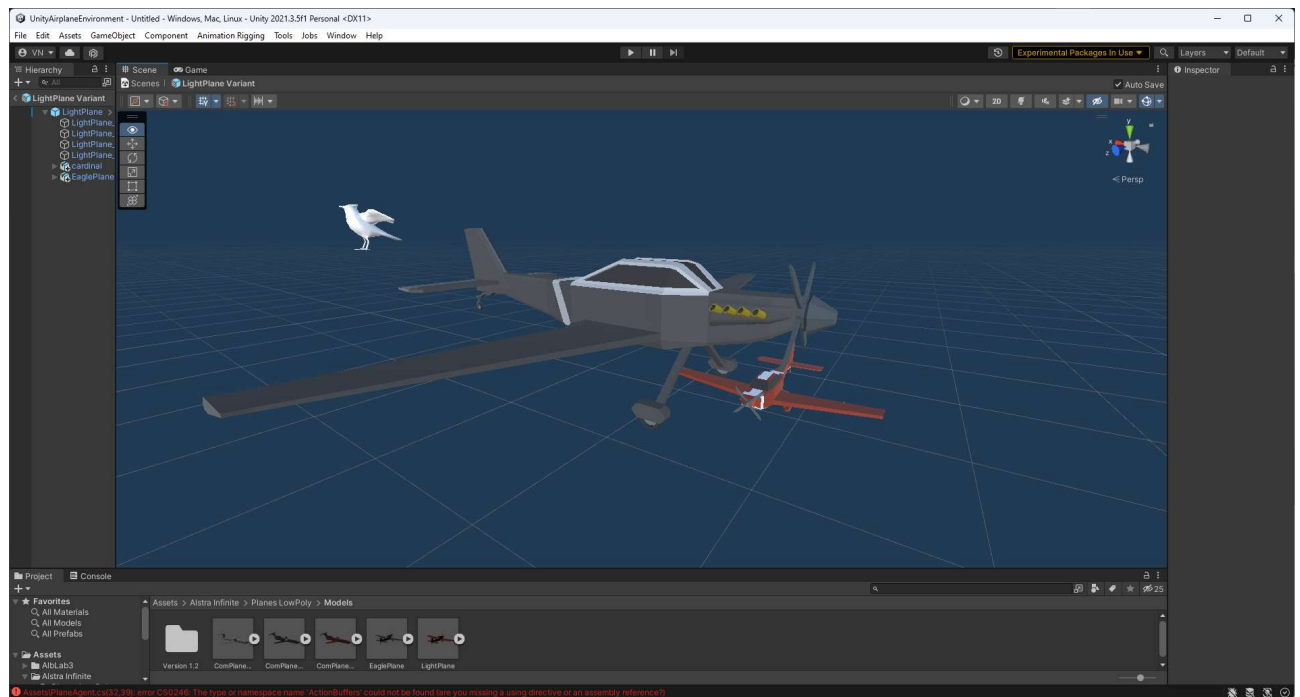


Рисунок 13 – Добавленные модели

**Вывод:** выполнили *Fork* библиотеки, изучили библиотеку *TensorAeroSpace*, установили библиотеку на свой локальный ПК, провели апробацию.