### **Проектирование и разработка фрагмента электронного образовательного ресурса в среде дистанционного обучения для проведения занятия**

**Тема занятия**: "Введение в использование нейросетей в разработке игр"

### **Форма отчетности: Фрагмент электронного образовательного ресурса**

#### **Практическое задание (Jupyter Notebook)**

**Описание**: Создайте и обучите простую нейросеть для классификации игровых объектов. Используйте библиотеку TensorFlow/Keras для реализации нейросети. Следуйте пошаговым инструкциям, представленным в Jupyter Notebook.

**Код**:

| import tensorflow as tf from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense from tensorflow.keras.utils import to\_categorical import numpy as np  # Генерация данных num\_samples = 1000 num\_features = 10 num\_classes = 3  X = np.random.rand(num\_samples, num\_features) y = np.random.randint(num\_classes, size=num\_samples) y = to\_categorical(y, num\_classes)  # Разделение данных на тренировочные и тестовые train\_size = int(0.8 \* num\_samples) X\_train, X\_test = X[:train\_size], X[train\_size:] y\_train, y\_test = y[:train\_size], y[train\_size:]  # Создание модели model = Sequential() model.add(Dense(64, input\_dim=num\_features, activation='relu')) model.add(Dense(32, activation='relu')) model.add(Dense(num\_classes, activation='softmax'))  # Компиляция модели model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])  # Обучение модели model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)  # Оценка модели loss, accuracy = model.evaluate(X\_test, y\_test) print(f'Test accuracy: {accuracy}') |
| --- |