Open Laboratory of Interest

2021 项目考核题

By OpenAdmin

2021 年 10 月 31 日

项目考核说明

注意　答题前请认真阅读此说明，并按照说明规定的方式答题和提交

1. 除第一题规定提交格式外，请在以下 4 道题中**任选一道**完成。我们也鼓励有能力的同学完成其中的多道。如果完成多道题目，计分时按照每道题目的得分从高到低排序，总分为「**最高分题目得分**+50%×**其它题目得分**」。

2. 项目考核时间自 2021 年 10 月 31 日起，至 2021 年 11 月 14 日止。最终的展示形式为现场演示和简易答辩，根据完成度与展示效果综合评分。即使某道题目未完成，也欢迎你跟我们聊聊解决问题过程中的心路历程，我们也会将你的认真程度计入评分。

3. 问题排序与难度无关，每道题侧重的是不同方面，你可以任选自己感兴趣的方面回答。如果感觉某道题目太难，可以尝试搜索题目后面附带的**关键词**，或是查看题目附带的教程。

4. 如果关于题目内容有任何问题，欢迎在「项目考核群」里提问（公平起见，请尽量不要单独联系出题人）。

祝你考核顺利，玩的开心！

1 Git 与 GitHub

出题人：Criheacy

与笔试题中的第一题类似，如果你能按本题的要求提交其余题目的代码，就可以获得额外的加分。

1.1 题目要求

Git 是一个非常常用的版本管理工具 (VCS, Version Control System)，它可以帮助你备份防止代码丢失，可以让你“撤销”到这份程序的上一个版本，还可以在多人协同开发中协调进度；提前学习这项工具的使用可以极大提升开发效率。

而 GitHub 是一个代码的在线仓库。它有点像是百度网盘之类的工具，不过它是专门为存储代码设计的。结合 Git 工具的使用，你可以在网上备份代码文件，也可以随时浏览项目的历史版本。

本题中，你需要了解并掌握它们的基本用法。详细计分规则如下：

+10%	建立一个 GitHub 账号，创建若干个**公开仓库** (Public Repository) 用来储存每道题目的代码文件。
+20%	在上述基础上，添加 `README.md` 文件用于简介、`.gitignore` 文件用于忽略一些不应该被提交的文件（比如相关环境、大量的数据集等，一般来说仓库大小应该在 `10MB` 以下）；有良好的文件目录结构（比如代码放在 `/src` 文件夹下、资源放在 `/res` 文件夹下等）。
+30%	在上述基础上，代码文件是使用 `Git` 工具提交的，合理管理 `Git` 的提交节点和分支结构，能看到代码的历史版本。

2 网页制作

出题人：QER

2.1 题目要求

对于所有同学的总目标 实现一个可以在浏览器输入 IP 地址来访问的网页。也就是说，其他同学都可以通过打开浏览器输入对应的 IP 地址，看到你制作的内容。

对于2021级同学

1. 用浏览器通过 IP 地址访问一张放在服务器上的网页，网页上需要放一张图片（图片内容自定）。
2. 用 html 的表格，制作一个座次表放在网页上。

对于2020级（及2019级）同学

1. 使用 flex 或者 grid 等 CSS 布局，制作一个座次表放在网页上。
2. 添加2个自定义的交互功能，例如点击座次表变色表示这个位置有人了、点击座位上的名字可以编辑等等。

2.2 验收标准与限制

1. 需要现场访问你提供的 IP 地址来验证是否正确
2. 对于 2021 级同学，你使用 html 默认的表格元素制作即可。
3. 对于 2020 级同学，你需要使用 flex 或者 grid 等 CSS 布局来实现座次表样式，不可以使用 html 原生表格。

2.3 加分项

1. 符合审美标准的网页
2. 使用 json 格式存储数据
3. 使用轻量级前后端框架
4. 使用表驱动的编程模式

2.4 提示

你需要做的事情包括但不限于：

1. 购买一个服务器。可以去腾讯云、华为云、阿里云等等找一个卖服务器的平台，花大约 10￥就可以买一个入门版的学生服务器。买了之后会给你一个服务器 IP 地址。买服务器会让你装系统，自己查阅资料可以装哪一个。
2. 在服务器上配置一个 web 服务环境
3. 写一个网页
4. 将网页放到服务器上并且能正确访问
5. ……

2.5 关键词

web 服务器 linux apache nginx html css xftp xshell 防火墙 端口

3 手写数字识别

出题人：Criheacy

3.1 题目要求

给定一张手写数字图像，请你设计一个程序根据图片上的内容识别写的数字是几。比如给定下面这张图片作为输入：

你的程序应该能判断是数字 5。你可以根据像素的分布和像素值的深浅，结合各个数字笔画的特征来设法判断，并尽可能提升判断的准确率。

3.2 验收标准与限制

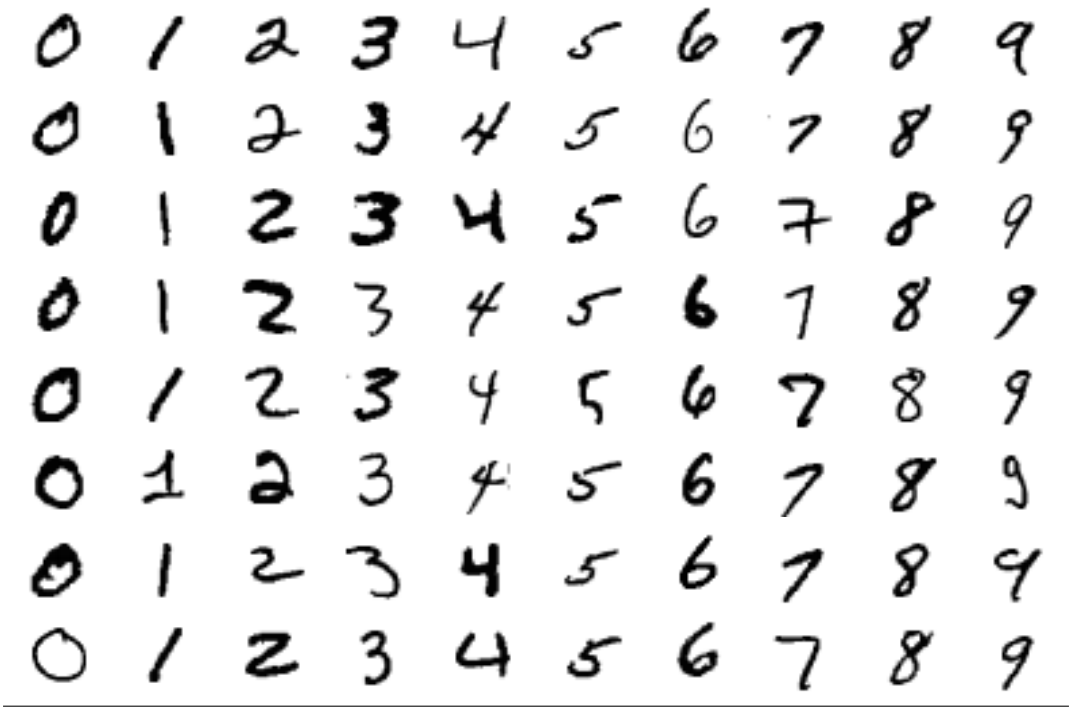
1. 能够现场输入一张程序从未见过的图片，输出识别结果。
2. 数字是通过**图像内容**识别出来的，而不是借助其它手段（比如输出一部分文件名）。
3. 自己实现算法和使用机器学习框架均可，不过不能使用现成的模型数据。你的程序应该只接受了你给定的数据集图片作为训练样本。

3.3 加分项

识别准确率越高越好。

3.4 提示

为了方便找到规律、设计算法，这里还提供了 60,000 张数字图片以及它们的标签（代表写的是数字几）。图片以「编号-数字.png」命名，下面列举了一部分图片数据：



0 1 2 3 4 5 6 7 8 9

这些数据也可以在 The MNIST database of handwritten digits 找到。

3.5 关键词

图像识别 OpenCV Pillow(Python)

机器学习 SVM kNN Neural Nets CNN

4 Unreal Engine 小游戏

出题人：*Ag2S*

Ag2S 很喜欢游戏，~~梦想着做3A大作~~；听说 UE (Unreal Engine) 效果不错，眼馋 UE5 的效果~~但是电脑带不起来~~，于是想用 UE4 制作游戏，但他现在还挣扎与 OPENGL 这些图形 API 的实验上，你能帮帮他吗？

4.1 题目要求

使用Unreal Engine 4 (虚幻 4 引擎) 制作一个小游戏，推荐使用蓝图。可以参考教程，**但不能直接用别人的项目修改**。不要求有多复杂。能够运行，可以交互即可。

4.2 验收标准与限制

1. 可以实机运行，能够现场演示。
2. 不限系统平台，Windows、Linux、Mac、ios、Android都可以。
3. **建议3D游戏**，游戏具体类型（如FPS,赛车游戏）可以参考UE提供的模板。
4. 可以使用UE自带的模板，**但不能只有模板中内容**。

4.3 加分项

1. 引用公开市场的模型、材质资源，让你的游戏拥有更漂亮的画面效果
2. 创建你自己的材质，让它实现法线贴图等效果
3. 不使用自带的场景（level），让游戏跑在在自己创建的场景
4. 在自己的场景里使用地形编辑器创建地形
5. 为你的地形创建材质、大气效果、刷上植被
6. 任何你想要的创意!

4.4 提示

1. 强烈建议使用蓝图而不是 C++ 进行开发，对于初学者而言，UE C++ 过于庞大复杂，且需要额外配置。C++ 项目也会占用更多的磁盘存储空间。
2. 关于资源，可以在 Unreal Engine 的 Markplace 中寻找免费资源下载，这里推荐 Unreal Learning Kit 中的材质

3. 在注册 EPIC 账号的时候**建议不要绑定国内邮箱**可能会收不到邮件

4. 学习入口：

   - Unreal Engine的在线课程，点击立即使用后，建议使用**游戏开发入门**课程，推荐完成第一模块**初识虚幻引擎4**
   - UE4官方文档

5 二十四点

出题人：Criheacy

5.1 题目要求

你玩过二十四点游戏吗？从一套扑克牌（除去大小王牌）中任意抽四张牌，你要想办法将这四张牌的点数通过加减乘除运算得到 24 这个数，这四个数必须全部出现在算式中，并且每个数只能出现一次。比如抽到的牌是4、7、8、8，我可以通过 $(7-(8\div 8))\times 4=24$ 得到 24 这个数。其中 J 点数为 11，Q 点数为 12，K 点数为 13。

现在需要你来协助我完成一个 24 点**联网**小游戏。在 `10.102.32.57` 服务器的 `2333` 端口上已经部署了一个 24 点游戏裁判。这个裁判来负责**出题**、**检查答案的对错**，以及**开始一场计分的 24 点答题比赛**等等。

你可以尝试在 windows 命令行或 linux 控制台输入以下指令：

```
curl http://10.102.32.57:2333/get_question -X POST
```

`curl` 命令是系统自带的向指定端地址发送 HTTP 请求的命令，上述命令中端地址被指定为服务器的 `/get_question` 接口。顾名思义，访问 `/get_question` 这个接口可以得到一道 24 点题目，如果一切顺利，你会在得到如下回复：

```
{
  "message": "Success.",
  "data": {
    "id": 12,
    "cards": ["4","2","8","3"]
  }
}
```

这其中的 id 字段为题目编号，cards 字段为需要你计算的四张牌的点数（你得到的题目编号和点数可能不同，不过格式应该是类似的）。经过你的计算，你发现只需要将这四张卡牌中的 2 与 3 相乘得到 6，将 8 与 4 相减得到 4，最后将这两次计算得到的 4 和 6 相乘即可得到 24（可能有多种解法，这里以其中一种举例）。

要想向裁判提交这道题的答案，相对来说就复杂一些，你需要先创建一个 answer.json 文件，内容如下：

```
{
  "question_id": 12,
  "answer": [
    {
      "left": 2,
      "operator": "mul",
      "right": 3
    },
    {
      "left": 8,
      "operator": "sub",
      "right": 4
    },
    {
      "left": 4,
      "operator": "mul",
      "right": 6
    }
  ]
}
```

然后在同一目录下打开命令行，并输入：

```
curl http://10.102.32.57:2333/answer -X POST -H "Content-Type:
    application/json" -d @answer.json
```

如果你能收到如下结果，就说明你已经成功提交了这道题，并且裁判判定你的结果正确。

```
{
  "message": "Success."
```

```
}
```

仔细观察上面的提交过程，思考一下我们是怎么向裁判说明我们要回答哪道题、又是怎么叙述我们的计算步骤的，或许聪明如你已经从中找出了一些规律。

可是这个游戏的玩家可不只有你啊！如果你要让玩家每次都要把答案按特定格式写进一个文件里，再在命令行中输入一大串命令，没过多长时间就没人想玩了。**你要做的就是为玩家创建一个玩家看得懂、方便操作的交互页面，然后自动帮玩家完成与裁判之间的通信过程。**这个页面可以是用鼠标拖动数字卡片完成运算的 web 应用，也可以是展示数字并让玩家输入算式完成运算的填空题，只要能准确地完成信息地传达过程，并且你觉得足够方便就行。

5.2 提示

「这篇文档」（也称为接口文档）里有裁判通信格式的详细信息。

5.3 加分项

1. 交互页面运行得稳定、准确、及时。
2. 能现场演示一道题目或一局游戏的接收回答过程。用户交互界面只需本地运行即可。
3. 用户不需要任何预备知识就能看懂，明白每一处应该如何操作；对于不易理解得部分可以提供提示或教程。

5.4 关键词

fetch ajax jQuery JSON