

# 字节跳动企业选题

邮箱: [szsse@ustc.edu.cn](mailto:szsse@ustc.edu.cn)

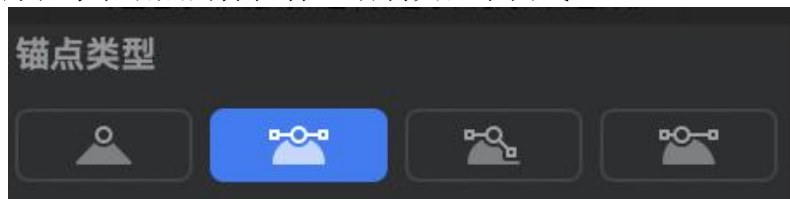
发送简历命名: 企业名+题目编号+姓名+班级, 每人限报 1 项

## 题目 1: SVG 编辑工具

类型: 前端

描述: 实现一个 SVG 编辑工具, 可以进行路径绘制及导出, 具体功能细节可以参考 sketch 的矢量工具或者 photoshop 的钢笔工具:

1. 建立一张画布, 画布可以修改宽度、高度;
2. 鼠标作为矢量工具的操控装置, 矢量工具可以在画布上进行锚点绘制路径, 路径可以为闭合路径及非闭合路径;
3. 路径可以及进行描边及填充, 可以修改边线宽度、颜色、端点类型及节点类型, 填充需要修改填充颜色及不透明度;
4. 鼠标可以选中路径, 并进行路径在画布上的移动、缩放、旋转操作, 路径可以复制、删除及调整顺序, 鼠标双击路径可以进行锚点编辑;
5. 锚点类型有如下四种, 分别为笔直、对称、断开连接及不对称, 可以通过鼠标去调节每个锚点的操控杆, 绘制贝塞尔曲线。



通过该编辑器最终导出 svg 的代码, 并可以在浏览器中正常展示。项目代码放在 github 上, 如果有能力可以自己部署到公网上最好。项目代码中包含每个组件的实现思路及组件使用方法说明。代码逻辑清晰, 注释合理

## 题目 2: 实现一个全景图 (720 度) 浏览器

类型: 前端

描述: 基于 three.js 实现一个全景图 (720 度) 浏览器, 可以切换漫游; [demo](#) (推荐移动模式打开)。

建议技术栈: HTML、JavaScript、WebGL、Three.js

功能要求:

- 实现可交互全景场景
- 实现漫游切换全景的功能

最终项目放到 github 上, 编写完整的设计文档, 代码逻辑清晰, 注释完备

## 题目 3: 实现一个 3D 魔方

类型: 前端

描述: 魔方是大家都喜爱的一个竞技玩具, 通过本练习可以实现一个基于 three 可交互的 3d 魔方, 了解图形学渲染流程、常见的矩阵变换, 为以后高性能渲染复杂场景打下坚实的基础。

建议技术栈：HTML、JavaScript、WebGL、Three.js

功能要求：

- 实现一个用 6 张贴图的 3d 魔方模型
- 滑动或点击可以旋转魔方

项目放到 github 上，编写完整的设计文档，代码逻辑清晰，注释完备。

#### 题目 4：实现一个轻量的 web3D 渲染引擎

类型：前端

描述：Three 是前端图形渲染最常用的渲染引擎，通过本练习可以实现一个轻量的 three 渲染引擎，了解图形学底层知识，渲染管线流程，为以后高性能渲染复杂场景打下坚实的基础。

建议技术栈：HTML、JavaScript、WebGL

功能要求：

- 基于 WebGL 实现轻量的 three.js 渲染引擎
- 有场景、相机、材质、常用的几何体 api、渲染器等抽象

最终项目放到 github 上，编写完整的设计文档，代码逻辑清晰，注释完备

#### 题目 5：制作一个 web 视频播放器

类型：前端

描述：过去的互联网长视频领域，一直在坚持着源端一体的结构（即绝大多数的处理都放到了中台服务上，播放器端除了基本的播放功能及一些简单的场景功能外，很少做特殊设置），旨在让市场更容易适应互联网长视频。近期，互联网长视频已基本占据了主导地位，代替了传统的广电市场。在这种基础上，市场对于长视频领域提出了新的期待，例如如何保护内容版权，如何保证广告的有效播出，如何增加用户在广告方向的互动等。这些期待赋予了播放器端新的使命。

我们在这里实现一个 web 播放剪辑器，以达到对于视音频基本的编解码，视音频的处理及视音频相关的 web 协议了解等目的。

基本要求：

- 实现一个基本的 web 播放器
- 支持播放在线 MP4 文件
- 支持输入特定的时间段对视频进行剪辑的功能

进阶要求：

- 不直接使用 video 标签
- 支持 MKV/flv 等多种视频格式
- 支持弹幕功能
- 支持 DASH/HLS 两种流媒体协议
- 支持对视频进行加密处理，只在这款 web 播放器中可以播出，且网络正常的情况下，延时不能超过 100ms
- 支持加密的基础上，支持特定的时间段不能快进或快退的处理

最终项目放到 github 上，编写完整的设计文档，编写对于 HLS/DASH/CMAF 等相关协议的调研文档，代码逻辑清晰，注释完备。

## 题目 6：制作一个网页版的纸牌游戏

类型：前端

描述：基本要求为只有两位玩家，每人 10 滴生命值，第一轮抽三张牌，而后每轮抽一张牌，牌分为进攻牌和防守牌，每轮出一张牌，直到一人生命值小于 0 游戏结束。

进阶要求：

- 提供注册登陆的功能
- 提供最多四人的游戏，只有一名胜利者，杀死一人游戏结束
- 提供游戏内聊天的功能

最终项目放到 github 上，编写完整的设计文档，如有能力最好部署到工网上，

代码逻辑清晰，注释完备。

## 题目 7：实现一个 Web 可视化组件布局系统

类型：前端

描述：目前前端可视化搭建是前端较热门的方向，已经较为成熟，但是对于用户体验而言，一个优雅易用的画布布局系统最为重要，这次的题目希望同学可以先调研下市面上已有的搭建、布局系统，在设计上可以有更多的参考对比。

基本要求：有组件候选区域，可以使用开源的组件库，有支持自由拖拽的画布区域，组件可以从候选区域拖拽到画布区域，并可以渲染到画布上，支持自由定位，画布支持标尺系统，组件在画布中拖拽时可以实时显示定位位置，渲染到画布中的组件关系是一个结构化的数据结构。

进阶要求：

- 组件之间支持自动吸附功能，当两个组件靠近时，可以自动对齐
- 支持导入、导出画布中的组件结构化数据
- 实现辅助线功能，组件在画布中拖拽时，显示组件之间的间距、和画布边距显示，并可以在辅助线的基础上对组件间的自动吸附功能做优化
- 实现组件树区域，可以显示画布中的组件列表，并支持换位、删除、隐藏、锁定功能
- 更多高级特性可以自由探索发挥，比如智能搭建、主题系统、国际化功能等

使用 React 技术栈，最终项目代码放到 github 上，规范清晰的设计文档，工程结构组织规范、代码注释清晰，最好可以有示例站点，书写测试用例，测试覆盖率最好可以达到 90%。

## 题目 8：实现一个 Web 即时通信系统

类型：前端+服务端

基本要求：支持登录、登出，支持单聊，支持文字消息，支持图片消息。

进阶要求：支持群聊，支持查看历史消息，支持已读、未读。

完整的系统设计文档，方便与导师沟通，需要有开发日志，及时记录开发时遇到的问题和工作总结，代码逻辑清晰，注释完备，代码托管到 github 上，readme 要包含项目介绍、工程目录结构介绍、调试方式介绍服务端建议使用 Node.js。

## 题目 9：使用 js 生成 low-poly 风格图像

类型：前端

描述：目前前端团队正在实践智能化相关的事情，其中会牵扯到图片识别、形状检测、图片归类等相关的处理，识别类的功能是基于图片相关的算法来实现的，图片归类方向的能力则是基于机器学习来实现的，二者相结合来一起辅助完成图片相关的能力建设，进而进一步完成上层跟前端相关的 UI 层面的能力建设。

今天，我们从另一个角度来窥探下图片识别相关的算法实现，我们会跟大家一起，从零到一基于 JavaScript 实现给定任意一张图片，生成 low-poly 风格的新图像；借此我们不仅能了解到图像识别相关的基础理论，也能借助于算法来实现对图片的进一步处理。

基本要求：了解图片处理基础理论，了解并学习相关算法理论，相关算法从零到一的实现，实现前端页面，支持图片上传完成 low-poly 图像生成。

工程要求：了解算法实现后完成算法实现模型图，基于 create-react-app 的本地开发环境搭建。基于 github 的 CI/CD 项目研发过程实践。

## 题目 10：设计一个抽奖系统

类型：服务端

基本要求：

- 抽奖活动有开始和截止日期，不在这个日期范围内不能抽奖
- 给定多个奖品以及每个奖品的数量，中奖概率，实现用户抽奖接口，对于部分奖品，活动期间每个用户最多抽中一次
- 每个用户的抽奖机会：
  - 每天只有 n 次抽奖机会，过期作废
  - 中奖的奖品有一个种类：再来一次，如果抽中再来一次可以继续抽奖一次，可累加，不过期
  - 可以给用户赠送抽奖机会，不过期，可累加

- 实现中奖榜单接口，展示过去一个小时的中奖用户，只展示十个，按照中奖的价值由大到小排列
- 需要一个简单的用户管理系统

进阶:

- 支持高并发
  - 异步发奖
  - 奖品超发
  - 异常情况考虑
- 支持在活动中动态调整奖品的数量和中奖概率
- 进行系统的 benchmark

其他说明:

- 语言不限，鼓励 golang
- 如果有余力可以自己实现前端页面
- 鼓励写单元测试

## 题目 11: 设计一个类似 12306 售票系统

类型: 服务端

基本要求:

- 参考 12306 站点进行售票系统建模设计，尽可能接近覆盖真实线上系统
- 实现的功能有但不限于:
  - 用户基本信息注册
  - 查询余票: 根据时间，车次，站点区间，座次(一等座，二等座，硬卧，硬座...)查询余票
  - 售票: 支持一次购买同一车次的多张车票(多人)，支持订单 30 分钟内锁定，超时释放。支付接口可以 mock
  - 退票: 支持一个用户账户下的批量退票
  - 改签: 同一用户一张车票只能改签一次
- 所有读写接口延迟要求  $\leq 500\text{ms}$
- 单机支持到 500qps 的并发

进阶要求:

- 实现完整的前端页面(或者 app)
- 分布式部署，负载均衡的基础上实现系统的水平扩展能力

其他说明: 语言不限，鼓励 golang

数据参考: 车次数据: [https://kyfw.12306.cn/otn/resources/js/query/train\\_list.js](https://kyfw.12306.cn/otn/resources/js/query/train_list.js)

站点数据: [https://kyfw.12306.cn/otn/resources/js/framework/station\\_name.js](https://kyfw.12306.cn/otn/resources/js/framework/station_name.js)

## 题目 12: 设计一个类似知乎问答系统

类型: 服务端

基本要求:

- 支持问题、回答的发布和修改 (文字)
- 问题拉链, 支持按热度和时间序排序
- 回答点踩
- 热门问题列表

进阶要求:

- 支持高并发
- 支持评论
- 问题、回答支持图片

其他说明:

- 语言不限, 鼓励 golang
- 如果有余力可以自己实现前端页面

## 题目 13: 设计一个类似蚂蚁森林系统

类型: 服务端

基本要求:

- 自己实现用户管理系统和好友系统
- 实现获取某个用户所有好友列表, 列表可以按照总能量总排行榜和周排行榜两种, 可以根据输入参数区分, 列表需要返回此好友未来 24 小时待收取的能量, 如果能量在 1 个小时内, 则展示能量收割倒计时
- 实现给某个用户添加能量的接口: 包含能量值和这个能量可以在什么时间点收取
- 实现“最新动态”: 可以查看历史所有你获取别人的能量和别人获取的你的能量, 按照天纬度分隔
- 实现倒计时功能: 获取好友列表时, 如果这个好友
- 某用户 A 的能量值, B 在 24 小时之内只能偷取一次, 并且总共偷取最多不能超过能量值的一半。

进阶:

- 实现任意好友之间某段时间的历史动态和能量收取 pk, 动态按照天分隔
- 进行系统的 benchmark

其他说明:

- 语言不限，鼓励 golang
- 如果有余力可以自己实现前端页面
- 鼓励写单元测试

#### 题目 14：设计一个类似蚂蚁森林 APP(与后端同学协同)

类型：客户端

基本要求：

登录和好友系统：

- 使用账号密码登录
- 好友系统，加好友，邀请好友，好友列表，好友 profile 页

首页：

- “最新动态”-好友收取信息展示，可回赠
- “排行榜”列表，包括周榜和总榜，可跳“他人页”
- “加好友收能量功能”

他人页：

- 可收取他人能量
- 能量比拼
- 好友最新动态

进阶：

- 使用跨平台语言开发
- 使用指纹/面容识别登录

其他说明：

- 和服务端蚂蚁森林项目共建，团队合作能力。
- 代码提交到 github 上
- 编写技术文档，介绍代码设计思路
- 语言不限 iOS(Swift/OC) Android(java/kotlin)

#### 题目 15：设计一个类似 12306 购票 APP(与后端同学协同)

类型：客户端

基本要求： 参考 12306 APP 功能设计开发

- 实现的功能有但不限于：
  - 登录功能
  - 乘车人列表
  - 查询余票：根据时间，车次，站点区间，座次(一等座，二等座，硬卧，硬座...)查询余票
  - 购票，退票，改签操作

- 订单查询，已支付订单，历史订单，未完成订单

进阶：

- 底部高级筛选功能

其他说明：

- 编写技术文档，介绍代码设计思路

语言不限： iOS(Swift/OC) Android(java/kotlin)

## 题目 16：设计一个类似知乎问答系统（与后端同学协同）

类型：客户端

基本要求：

- 支持问题、回答的发布和修改（文字）
- 问题拉链，支持按热度和时间序排序
- 回答点踩
- 热门问题列表

进阶要求：

- 支持评论
- 问题、回答支持图片

其他说明：

- 语言不限 iOS(Swift/OC) Android(java/kotlin)
- 编写技术文档，介绍代码设计思路