



پروژه برنامه نویسی سوکت: طراحی بازی Uno به روش کلاینت-سرور

هدف پروژه:

هدف این پروژه، تمرین برنامه نویسی سوکت در پایتون از طریق طراحی و پیاده سازی بازی Uno به صورت شبکه ای و با استفاده از سوکت های پایتون و به صورت کلاینت-سرور است. در این پروژه شما با مفاهیم ارتباطات شبکه، مدیریت ارتباط بین کلاینت ها و سرور، و تعاملات بین آنها در یک بازی چند نفره آشنا می شوید.

پیش نیازها:

1. آشنایی با زبان برنامه نویسی پایتون
2. آشنایی با مفاهیم شبکه و سوکت ها

توضیحات کلی پروژه:

- از ریبازیتوری "[bennuttall/uno](https://bennuttall.github.io/uno/)" به عنوان مبنای پروژه استفاده کنید.
- بازی باید بین چند کلاینت و یک سرور مرکزی اجرا شود (تعداد کلاینت ها باید دو نفر و یا بیشتر باشد).
- سرور وظیفه مدیریت جریان بازی، وضعیت مهره ها، و ارسال پیام های لازم به کلاینت ها را بر عهده دارد.
- کلاینت ها باید بتوانند به سرور متصل شوند، لاگین کنند، مهره ها را حرکت دهند و اطلاعات مربوط به بازی را دریافت و ارسال کنند.

ویژگی های مورد انتظار بازی:

1. ساختار سرور:

- پیاده سازی سرور مرکزی که ارتباطات سوکت ها را مدیریت کند.
- توانایی مدیریت چندین کلاینت به صورت همزمان.
- کنترل جریان بازی شامل شروع بازی، ثبت حرکات بازیکنان، و ارسال وضعیت جدید بازی به کلاینت ها.
- مدیریت لاگین کاربران و ذخیره سازی اطلاعات مربوط به سابقه برد و باخت ها در دیسک (فایل یا دیتابیس).

2. ساختار کلاینت:

- پیاده‌سازی کلاینت‌هایی که قابلیت اتصال به سرور را داشته باشند.
- نمایش رابط کاربری اولیه برای لاگین، حرکت مهره‌ها و نمایش وضعیت بازی.
- ارسال دستورات بازیکن به سرور و دریافت پاسخ‌ها از سرور.
- نمایش سابقه برد و باخت کاربر.

3. قوانین بازی:

- بازی باید قوانین استاندارد uno را رعایت کند.
- سرور باید وضعیت بازی را ذخیره کند و هرگونه حرکت ناهنجار را شناسایی کند.

توضیح عملیات کلاینت و سرور:

• سرور:

- مدیریت اتصال‌ها و احراز هویت کاربران.
- ذخیره‌سازی و به‌روزرسانی سابقه برد و باخت در دیسک.
- کنترل قوانین بازی و ارسال وضعیت به کلاینت‌ها.

• کلاینت:

- اتصال به سرور و ارسال اطلاعات لاگین.
- نمایش رابط کاربری برای ورود به بازی و نمایش وضعیت دیگر بازیکنان.
- ارسال حرکات بازیکن به سرور و دریافت پاسخ.
- نمایش سابقه برد و باخت.

الزامات فنی پروژه:

- استفاده از کتابخانه‌های استاندارد پایتون برای برنامه‌نویسی سوکت (مانند "socket"). استفاده از کتابخانه های third party برای بخش سوکت مجاز نمی باشد.
- پیاده‌سازی ارتباط TCP برای اطمینان از اتصال پایدار.
- استفاده از مفاهیم چند نخ (Threading) یا Asynchronous برای مدیریت ارتباطات همزمان.

پیشنهادهای برای دریافت نمره اضافه:

- اضافه کردن قابلیت چت بین بازیکنان.
- امکان ذخیره‌سازی و بارگذاری بازی‌های ذخیره شده.
- پیاده سازی روش هایی برای جلوگیری از تقلب توسط بازیکنان.

معیارهای ارزیابی:

- عملکرد صحیح سرور و کلاینت‌ها.
- رعایت قوانین بازی.
- مدیریت بهینه ارتباطات و پایداری سیستم.
- کیفیت مستندات و توضیحات.

نکته: هرگونه خلاقیت در نحوه پیاده سازی پروژه با نظر تیم حل تمرین میتواند به عنوان نمره اضافه در نظر گرفته شود.

موفق باشید