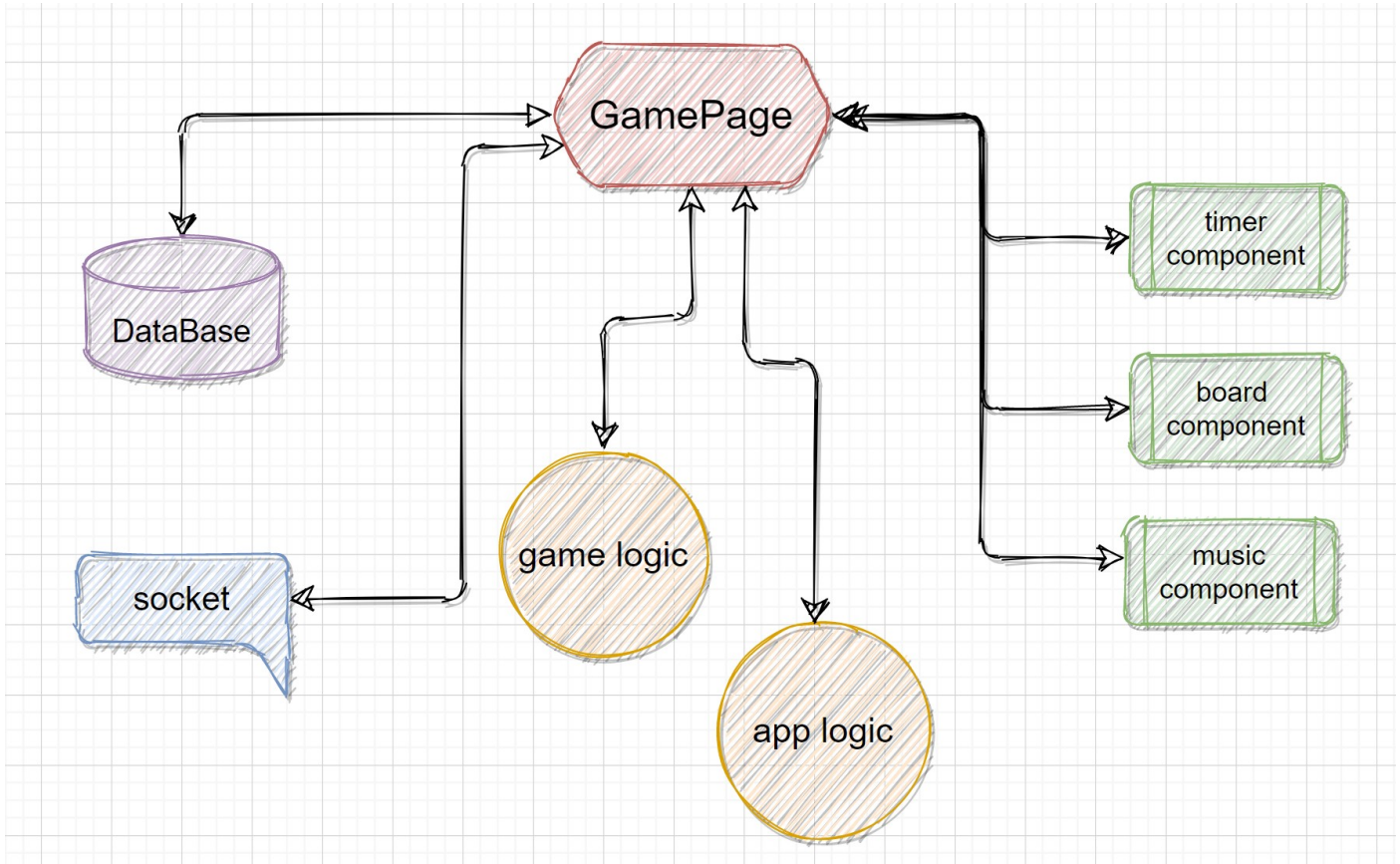


# گزارش پروژه پایان ترم برنامه نویسی پیشرفته

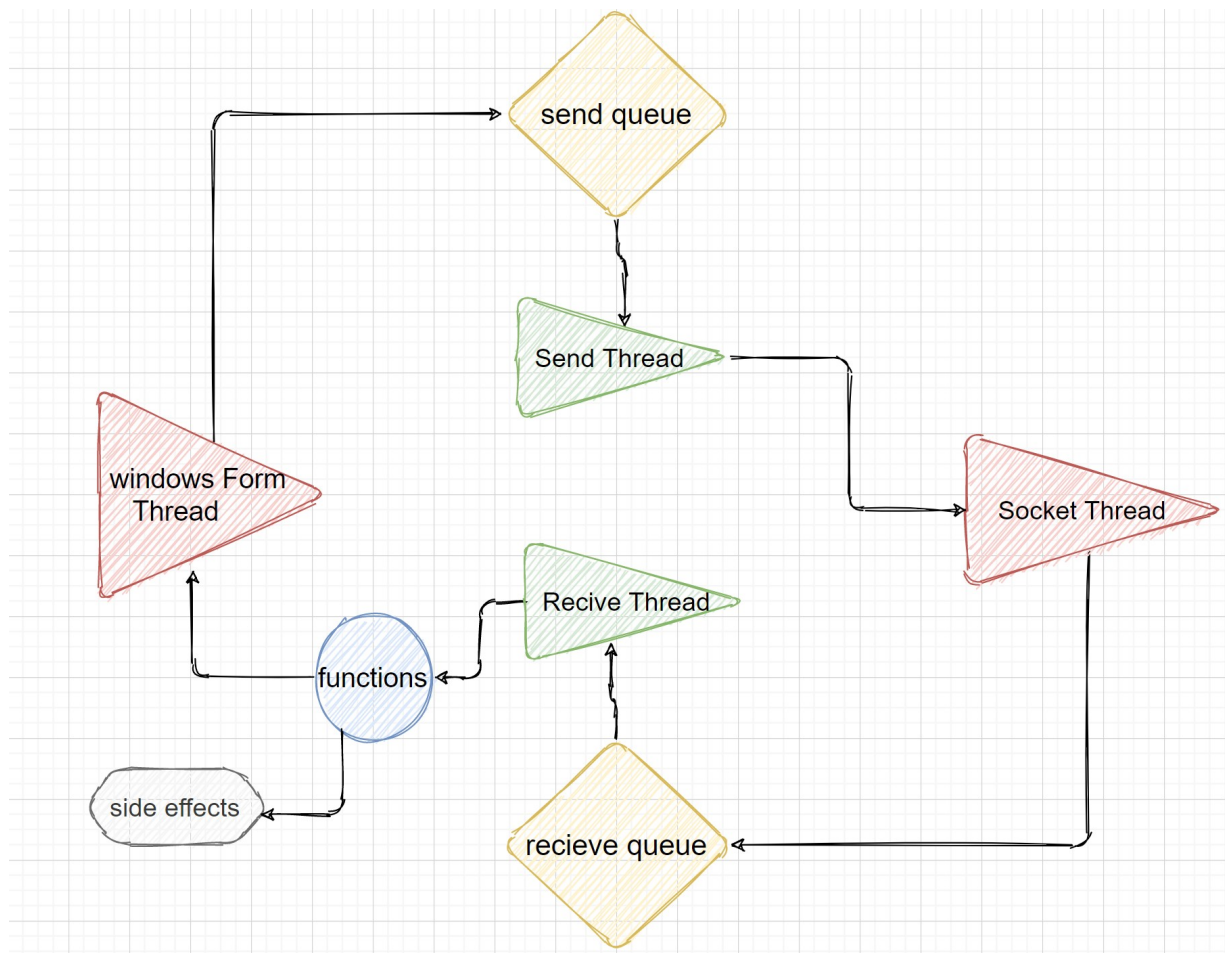
اعضای گروه: حمیدرضا بلوری کاشانی - مهدیه افتخاریان

موضوع: شطرنج

توضیحات و روند کلی اجرای برنامه:



روند کلی اجرای برنامه به این صورت است که صفحه اصلی بازی یک رابط برای اجزای بصری صفحه ( تایمر - صفحه شطرنج - پخش کننده موزیک ) و اجزای منطقی ( تابع های دیگر برنامه ) و اجزای پشت پرده ( دیتابیس و سوکت ) میباشد. لازم به ذکر اس که پروژه کاملا به صورت ماژولار نوشته شده است.



مدل اجرای در در اکثر صفحات به صورت 4 ترد می باشد

1. **ترد گرافیکی ( windows form ):** برای پردازش محیط گرافیکی که توسط کتابخانه windows form مدیریت می شود
2. **ترد سوکت ( socket ):** برای ارسال و دریافت اطلاعات از طریق سوکت:  
این ترد کارهای عمومی ارتباطی شامل ارسال و دریافت اطلاعات، وصل شدن به سرور و ایجاد سرور را بر مبنای پروتوکل socket انجام میدهد
3. **ترد ارسال ( send ):** برای ساماندهی و مدیریت ارسال ها از طریق سوکت:  
این ترد با دریافت اطلاعاتی که ترد سوکت و ترد های دیگر در متغیر «صف ارسالی ها» ( send queue ) در آن ذخیره کرده اند، در پس زمینه آن ها را مدیریت و ارسال میکند.
4. **ترد دریافت ( receive ):** برای ساماندهی و مدیریت دریافت ها از طریق سوکت:  
1. این ترد با دریافت اطلاعاتی که ترد سوکت در متغیر «صف دریافتی ها» ( receive queue ) همگام دریافت اطلاعات ذخیره کرده، تابع هایی که برای کار با آن اطلاعات به آن داده شده را اجرا میکند که آن تابع ها میتوانند روی محیط گرافیکی اثر بگذارند یا تغییرات دیگری در روند اجرای برنامه انجام دهند.

```

1  {
2    "event": "setting",
3    "board": [
4      [ -2, -3, -4, -5, -6, -4, -3, -2 ],
5      [ -1, -1, -1, -1, -1, -1, -1, -1 ],
6      [  0,  0,  0,  0,  0,  0,  0,  0 ],
7      [  0,  0,  0,  0,  0,  0,  0,  0 ],
8      [  0,  0,  0,  0,  0,  0,  0,  0 ],
9      [  0,  0,  0,  0,  0,  0,  0,  0 ],
10     [  1,  1,  1,  1,  1,  1,  1,  1 ],
11     [  2,  3,  4,  5,  6,  4,  3,  2 ]
12   ],
13
14   "restTime": 1793,
15   "showTimer": true,
16
17   "IsMusicPlaying": false,
18   "selectedMusic": "",
19
20   "showMovePreview": true,
21   "theme-bg": 0,
22   "theme-color": 2,
23   "theme-style": 1
24 }

```

پس از هر تغییر در وضعیت برنامه، داده‌ها به فرمت JSON (مانند تصویر بالا) ارسال میشوند.

1. **Event** : نوع داده ارسالی
2. **board** : صفحه شطرنج
3. **restTime** : زمان باقی‌مانده
4. **showTimer** : آیا آلمان زمان در صفحه نشان داده شود یا خیر
5. **IsMusicPlaying** : آیا موزیک در حال پخش است؟
6. **selectedMusic** : نام موزیک انتخاب شده
7. **theme-bg** : شماره «رنگ صفحه» - بین 0 تا 2
8. **theme-color** : شماره «رنگ مهره‌ها» - بین 0 تا 2
9. **theme-style** : شماره «حالت مهره‌ها» - بین 0 تا 2

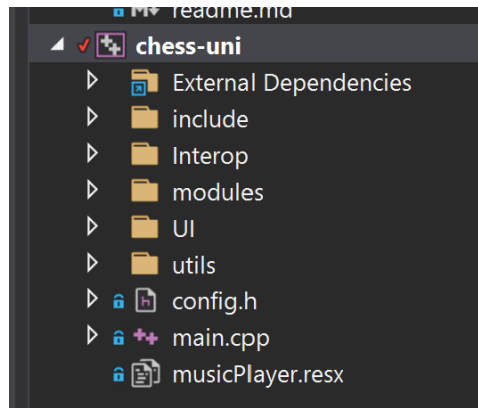
```
1 {  
2   "event": "file",  
3   "name": "audio.mp3",  
4   "type": "music",  
5   "content": "..."  
6 }
```

مدل داده‌ای ارسال اطلاعات برای انتقال فایل – محتوای فایل ( content ) با الگوریتم base64 همگام ارسال رمزنگاری می‌شود

```
1 {  
2   "event": "move",  
3  
4   "from": [0, 0],  
5   "to":   [1, 1]  
6 }
```

مدل ارسال اطلاعات برای حرکت مهره

## توضیح ساختار فایل و پوشه بندی پروژه:



- **پوشه include و Interop:**

شامل کتابخانه های اضافه شده به پروژه است

1. base64 : برای رمزنگاری فایل های باینری ( و ارسال از طریق سوکت )
2. json : برای فرمت کردن اطلاعات ( برای ارسال از طریق سوکت )
3. Interop.WMPLib و AxInterop.WMPLib : برای پخش کننده موزیک

- **پوشه UI**

شامل اجزاء گرافیکی برنامه:

1. ChessBoard.h : شامل محیط گرافیکی صفحه شطرنج و کلاس های ارتباطی صفحه شطرنج میباشد
2. GamePage.h : کلاس و داده های مربوط به صفحه بازی
3. introPage.h : صفحه اول ورود به برنامه ( انتخاب سرور یا کلاینت )
4. waitroom.h : صفحه انتظار برای وصل شدن طرف دیگر
5. musicPlayer.h : جزء بخش کننده موزیک
6. selectOption.h : جزء انتخاب کننده ( استفاده شده در بخش تنظیمات )
7. timer.h : جزء تایمر

- **پوشه Modules**

شامل کدهای سوکت و دیتابیس میباشد

- **پوشه utils**

شامل دیگر تابع های مفید که در سرار پروژه مورد استفاده قرار میگیرند

1. fileio.h : تابع های مربوط به کار کردن با فایل ها
2. lambda.h : شامل کلاسی برای پیاده سازی نگهدارنده ای شامل تابع و مقداری که قرار است در زمان فراخوانی به آن تابع داده شود
3. gameLogic.h : شامل تابع ها و داده های اولیه برای ساختار صفحه شطرنج و مهره ها میباشد
4. ui.h : شامل اطلاعاتی درباره تم ها و رنگ های به کار رفته در برنامه میباشد
5. packet.hpp : شامل کلاسی برای بسته بندی کردن داده ها قبل از ارسال از طریق سوکت میباشد
6. utils.h : شامل تابع های عمومی که در سرار برنامه مورد استفاده قرار میگیرد

- **فایل config.h**

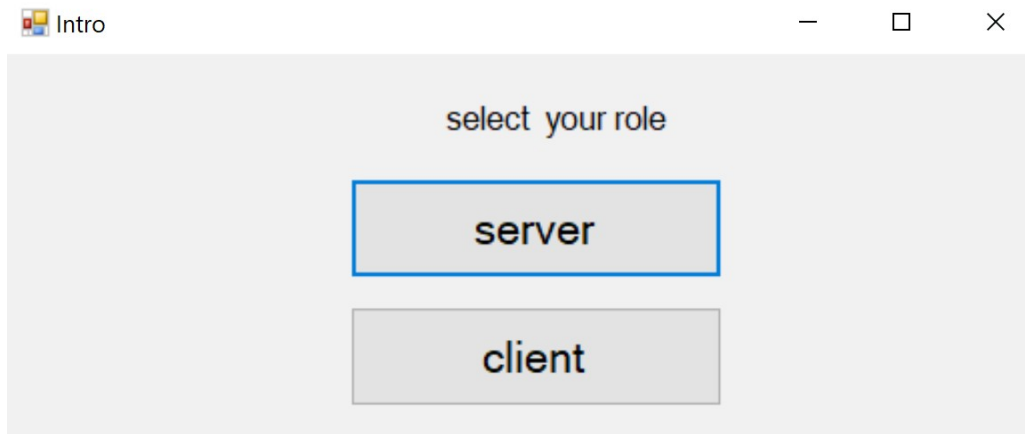
فایل تنظیمات عمومی که در سراسر برنامه مورد استفاده قرار میگیرد

- **فایل Main.cpp**

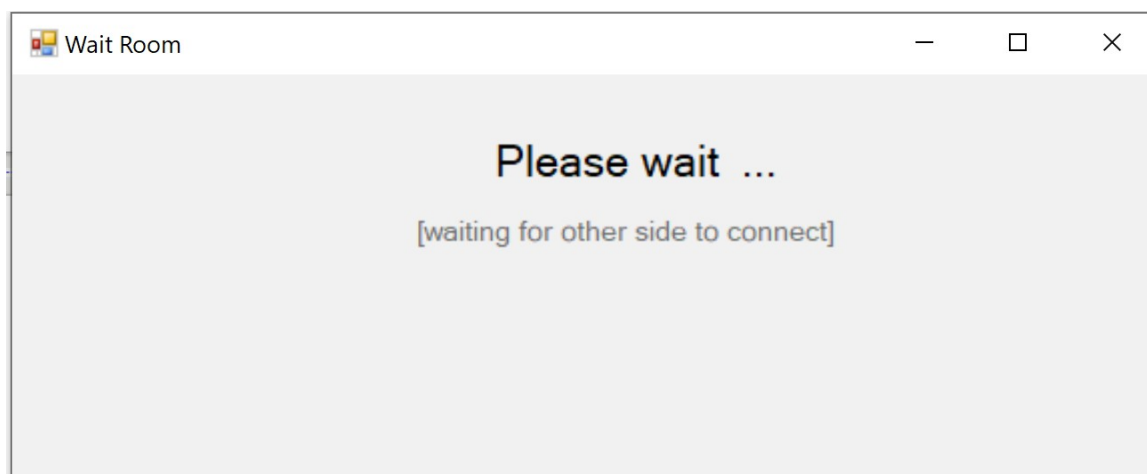
فایل ورودی برنامه که وظیفه ترتیب اجرای اجزای مختلف برنامه را عهده دار میباشد

## راهنمای استفاده از برنامه:

1. ابتدا برنامه را سرور و کلاینت را باز میکنیم و نقش هر کدام را انتخاب میکنیم:



2. منتظر میمانیم تا بهم وصل شوند



3. بعد از اینکه بهم وصل شدند صفحه اصلی بازی برای هر دو ظاهر میشود



1. در این بخش نقش هر کاربر نوشته می شود
2. زمان باقی مانده بازی
3. آیدی بازی ( برای ذخیره سازی بازی مورد استفاده قرار میگیرد ) ( این بخش فقط برای نقش server نمایش داده می شود )
4. موزیک پلیر
5. بخش تنظیمات ظاهری بازی ( فقط برای server
6. بخش بارگزاری بازی ناتمام قبلی ( فقط برای server ) - اطلاعات سمت سرور در پوشه databases به مدت هر 5 ثانیه یکبار با همان نام آیدی بازی ذخیره میشود

## کد های نوشته شده توسط اعضا:

غیر از بخش های بخش کننده موزیک ( UI/musicPlayer.h ) و بخش بررسی حرکتهای مجاز مهره های شطرنج در فایل ( UI/chessBoard.h و کلاس boardClass ) که توسط «خانم مهدیه افتخاریان» نوشته شده بقیه کد ها توسط «حمیدرضا بلوری کاشانی» نوشته شده است.