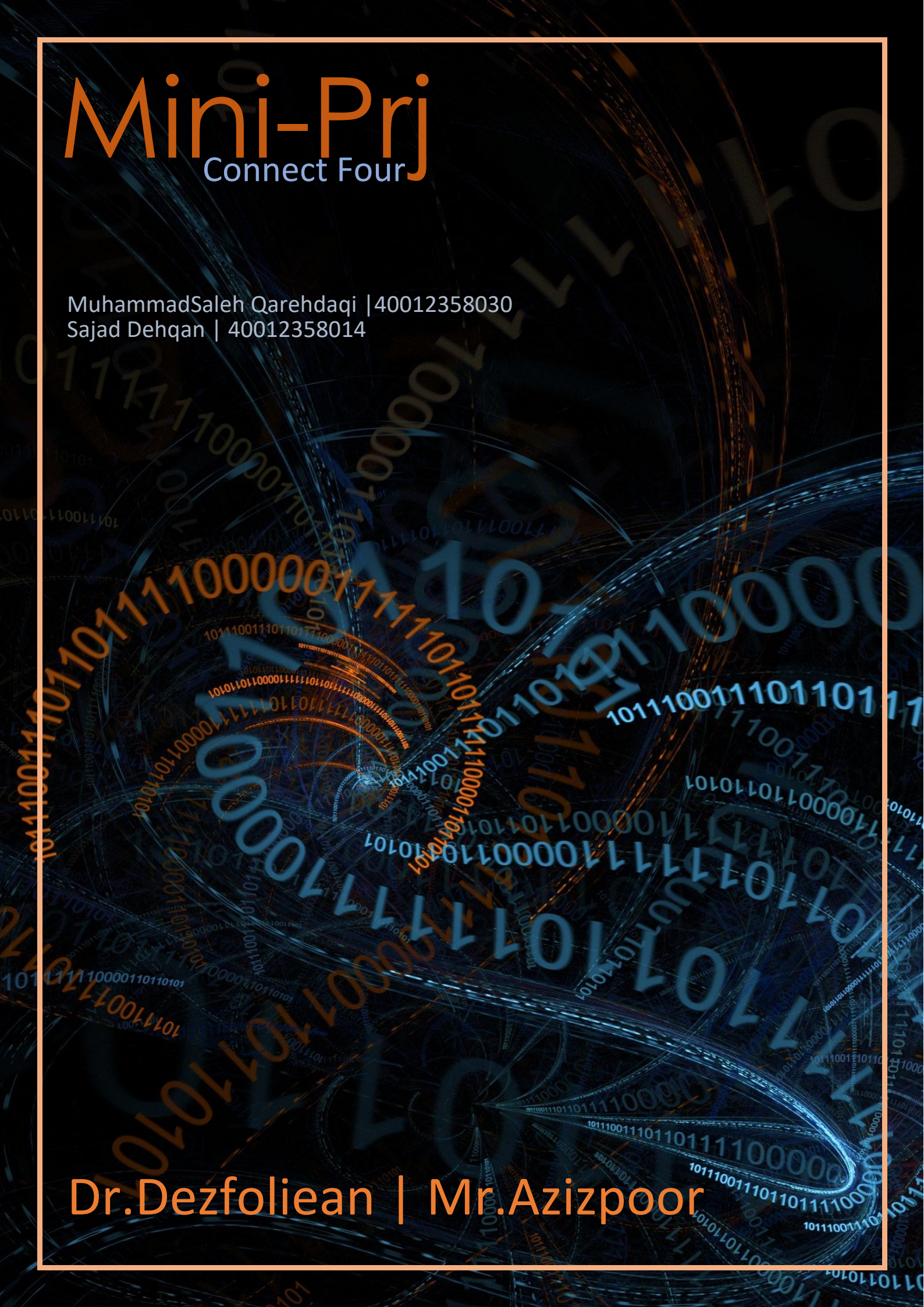
Mini-Prj

Connect Four

MuhammadSaleh Qarehdaqi |40012358030
Sajad Dehqan | 40012358014



Dr.Dezfoliean | Mr.Azizpoor

توضیحی بر عملکرد بازی

این بازی در دوحالت شروع بازی جدید و یا بارگیری از سیو قبلی در صورت وجود طراحی شده که بازیکن میتواند در هر لحظه دلخواه از بازی خارج شده و ادامه بازی را بعدا از سر گیرد.

در صورت وجود ذخیره قبلی بازی را از قسمت save ادامه میدهیم و اگر هم خواستیم بازی جدیدی را شروع کنیم وارد قسمت play می شویم.

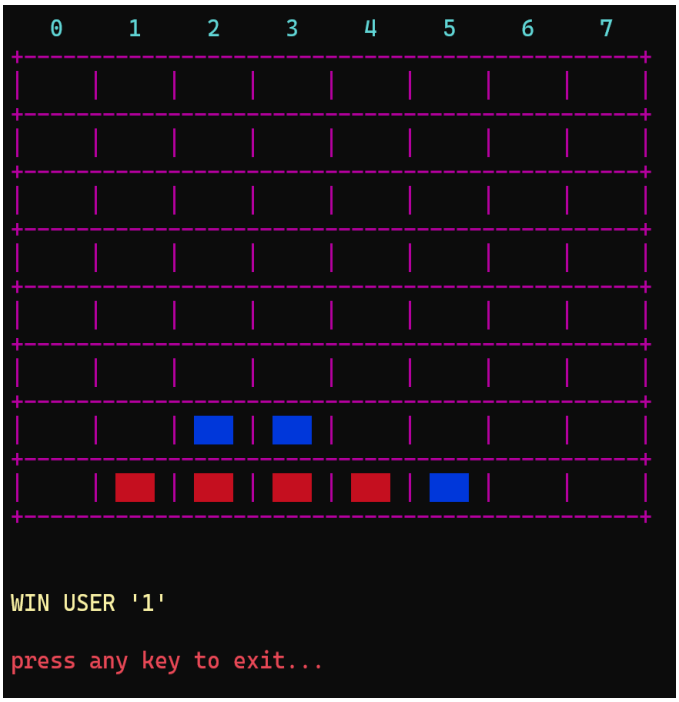
```
Welcome to  ''4 IN A ROW''  game.
we wish you have a nice time...

choise what do you want to happen?

1. (H)ELP
2. (P)lay (to Creat a new game)
3. (S)ave (to Continue from last save)
4. (F)ile (to Read a save file as '0 & 1' in terminal)
5. (E)XIT

Enter choise ----> |
```

اگر قسمت file را فراخوانی کنیم برای ما سیو بازی را که به صورت باینری میباشد و قابل مشاهده نیست به اعداد 0 و 1 تبدیل میکند و در ترمینال نشان میدهد که خط اول مجموع مرحله های رفته شده بازیکنان را نشان میدهد و خط های بعدی به ترتیب سه رقم اول از چپ کد شماره بازیکن سه رقم دوم کد ستون انتخابی و سه رقم آخر کد رنگ بازیکن را به صورت اعداد دودویی نشان میدهد.




```
----------save----------
0000111

001011000
010011011
001100000
010101011
001010000
010010011
001001000

 Enter any key to return  menu
```

عملکرد توابع موجود

```
void menu(char *choise);
```

همانطور که از نام تابع مشخص است جهت نشان دادن منوی بازی فراخوانی میشود.

```
void Help();
```

از این تابع جهت فراخوانی قوانین بازی استفاده میشود تا کاربر با بازی آشنا شود.

```
int Start(int board[][size]);
```

این تابع مسئولیت فراخوانی توابع لازم برای شروع بازی به حالت دو نفره را عهده دار است.

```
void Color(int *ColorUser1, int *ColorUser2);
```

از این تابع برای مشخص شدن رنگ هر یک از بازیکنان استفاده میشود.

```
int ChoiseColumn(int turn, int *column, int bin[]);
```

این تابع شماره ستون مد نظر بازیکن برای انداختن مهره خود را میگیرد.

```
void cycle(int turn, int column, int coloruser1, int coloruser2, int bin[]);
```

این تابع در تابع start فراخوانی میشود و سیکل چرخش بازی را تا اعلام شدن برنده بر عهده دارد و بین نوبت بازیکنان چرخش ایجاد میکند.

```
void PrintBoard(int board[][size], int column, int USER, int ColorUser1, int ColorUser2);
```

این تابع پس از مشخص شدن رنگ بازیکن و ستون انتخابی، محل مورد نظر را به رنگ بازیکن در می آورد و تخته را بر اساس آنچه خواسته شده پرینت میکند.

```
bool check_board(int board[][size]);

bool check_four_nuts(int board[][size], int *num, int *copy_i, int *copy_j, int *first);
```

پس از اینکه بازیکن ستون مورد نظر را انتخاب کرد دو تابع بولین چک برد وظیفه بررسی منطق و شروط پیروزی را عهده دار هستند که در صورت رخ دادن پیروزی مقدار true و در غیر اینصورت false را بر میگرداند که اگر پیروزی رخ دهد بازی تابع :

```
void EXIT();
```

را فرا میخواند و پایان می یابد.

اما در بخش کار با فایل هم توابعی وجود دارد که به بررسی آن می پردازیم.

```
void FWrite(int user, int column, int color);
```

هنگامی که بازیکنان انتخاب های خود را انجام می دهند این تابع پس از هر انتخاب اطلاعات بازیکن را به صورت دودویی در فایل ذخیره میکند.

```
void ConvertB_info(int dec, int bin[3]);

void ConvertB_step(int dec, int bin[7]);
```

این دوتابع اطلاعات بازیکنان شامل شماره ستون انتخابی و رنگ آن را به مبنای دسیمال دریافت می کند و جهت راحتی ما برای نمایش اطلاعت به صورت دودویی به مبنای باینری 0 و 1 میبرد تا در فایلمان نمایش به صورت دودویی داشته باشیم.

```
int ReConvertB_info(int bin[]);

int ReConvertB_step(int bin[]);
```

این دو تابع کار معکوس توابع قبلی را انجام میدهند تا ما بتوانیم اعداد دودویی ذخیره شده در فایل را به راحتی در مبنای ده بخوانیم و عملیات سیو بازی را تا جایی که کاربران بازی کرده اند را پردازش کنیم.

```
void merge(int turn, int bin[]);
```

از این تابع استفاده میکنیم که تعداد مجموع حرکات بازیکنان را در ابتدای یک فایل جدید بنویسیم و از فایل قبلی اطلاعات حرکت بازیکنان را به انتهای فایل جدید منتقل کنیم و در نهایت با استفاده از تابع:

```
void FRead(int board[][size]);
```

محتویات فایل مرج شده را بخوانیم و تمامی حرکات سیو شده قبلی بازیکنان را از قسمت save منو بازی بازخوانی کنیم.

```
void PrintFile();
```

اما در نهایت با استفاده از قسمت file در منو میتوانیم اطلاعات حرکات سیو شده را به صورت 0 و 1 در ترمینال مشاهده کنیم که خط اول حاوی مجموع تعداد حرکات بازیکنان و خطوط بعدی شامل اطلاعات شماره بازیکن ستون انتخابی و رنگ آن می باشد.