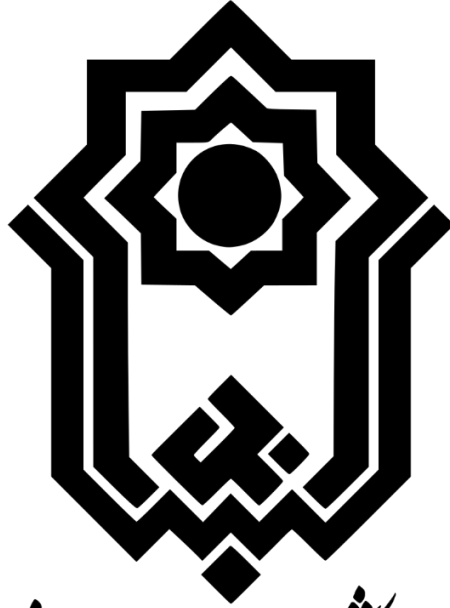




به نام آفریننده‌ی زمستان



دانشگاه بوعلی سینا

پروژه برنامه سازی پیشرفته

(فاز اول)

شطرنج پیشرفته

(Chess++)

استاد درس :

دکتر علی جاویدانی

دستیاران آموزشی :

کیانا جابری - مهدیس یعقوب انصاری - مبینا صفری - پارسا دولت آبادی - متین امیرپناه فر -
محمد متین سلیمانی - شهاب ثمری شمس - نیما مخملی

زمستان ۱۴۰۴

فهرست

1.....	مقدمه
2.....	کلیات پروژه
2.....	روند کلی بازی
۳.....	اهداف پروژه
4.....	معرفی مهره ها و ویژگی آنها
6.....	مهره های خاص
۷.....	فیچرهای منطق بازی و هوشمندی
۸.....	حالت های بازی
۹.....	بخش نمایشی
۱۰.....	بارم بندی و سیستم نمره دهی پروژه
۱۲.....	ددلاین ونحوه ی تحویل پروژه

مقدمه

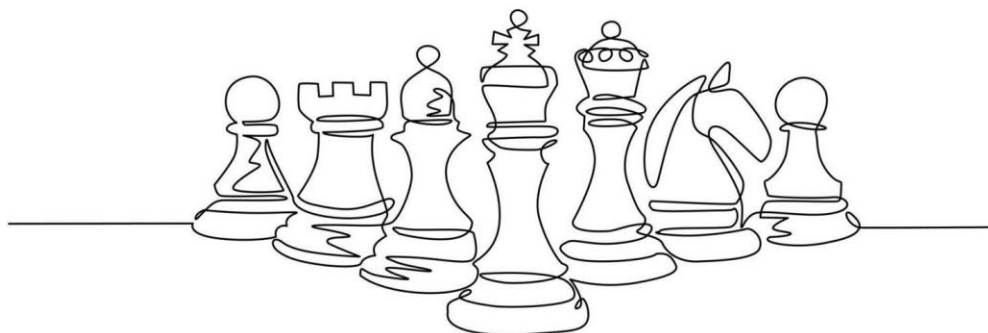
شطرنج، یکی از کهن‌ترین و پرطرفدارترین بازی‌های فکری جهان، همواره نمادی از تفکر، استراتژی و تصمیم‌گیری هوشمندانه بوده است. با وجود گذشت قرن‌ها از پیدایش این بازی، شطرنج همچنان الهام‌بخش طراحان بازی و برنامه‌نویسان برای خلق نسخه‌های نوآورانه‌تر است.

پروژه‌ی حاضر با عنوان **chess++** با هدف بازآفرینی این بازی کلاسیک در قالبی دیجیتال و مدرن طراحی شده است. در این پروژه، از دانشجویان انتظار می‌رود نسخه‌ای از بازی شطرنج را پیاده‌سازی کنند که علاوه بر قوانین سنتی، دارای ویژگی‌ها یا مهره‌هایی جدید و منحصر به فرد باشد، عناصری که بتوانند جریان بازی را تغییر دهند و استراتژی‌های تازه‌ای ایجاد کنند.

در این نسخه از شطرنج، علاوه بر مهره‌های اصلی، یک یا چند مهره‌ی ویژه با قابلیت‌های خاص معرفی می‌شوند که امکان انجام حرکات متفاوت یا اثرگذاری‌های خاص روی صفحه‌ی بازی را دارند. هدف اصلی از طراحی چنین تغییری، ایجاد تجربه‌ای متفاوت از شطرنج کلاسیک و تلفیق خلاقیت با منطق در بستر یک پروژه‌ی نرم‌افزاری است.

پروژه‌ی **chess++** نه تنها بازسازی یک بازی شناخته‌شده است، بلکه تلاشی برای افزودن خلاقیت، هیجان و چالش‌های جدید به آن محسوب می‌شود.

نسخه‌ای مدرن از شطرنج که مرز میان سنت و نوآوری را در قالب کدنویسی به تصویر می‌کشد.

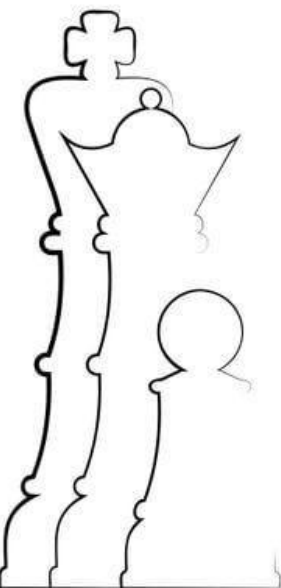


کلیات پروژه

پروژه‌ی Chess++ یک بازی شطرنج دیجیتال در محیط ترمینالی (CLI) است که از دو بازیکن پشتیبانی می‌کند. صفحه‌ی بازی شامل ۸ سطر و ۸ ستون است و مهره‌ها در موقعیت‌های استاندارد قرار می‌گیرند. هدف نهایی بازی، مات کردن شاه حریف است؛ با این تفاوت که در این نسخه، تعدادی مهره‌ی جدید با ویژگی منحصر به فرد به مجموعه‌ی مهره‌ها اضافه می‌شود، بعلاوه بعلاوه تعدادی قانون که منطق بازی را تحت تاثیر قرار خواهند داد.

روند کلی بازی

- بازی بین دو بازیکن (سفید و سیاه) آغاز می‌شود.
- بازیکنان به صورت نوبتی حرکت انجام می‌دهند.
- هر حرکت شامل انتخاب یک مهره و تعیین موقعیت مقصد است.
- قوانین اصلی شطرنج حفظ می‌شود، اما مهره‌ی‌های خاص قابلیت ویژه‌ی خود را دارد.
- در هر زمان امکان ذخیره یا بارگذاری وضعیت بازی وجود دارد.



اهداف پروژه

هدف اصلی از پروژه‌ی ++chess طراحی و پیاده‌سازی نسخه‌ای دیجیتال از بازی شطرنج است که ضمن وفاداری به ساختار اصلی بازی، با افزودن عناصر جدید و خلاقانه، تجربه‌ای متفاوت برای بازیکنان ایجاد کند. این پروژه فرصتی است تا شما توانایی خود را در طراحی نرم‌افزارهای ساخت‌یافته و مقیاس‌پذیر به نمایش بگذارید. اهداف کلیدی این پروژه عبارت‌اند از:

- **طراحی شیء‌گرا (Object-Oriented):** ساختار کلی برنامه باید بر پایه‌ی طراحی شیء‌گرا باشد تا هر بخش از سیستم (مانند مهره‌ها، صفحه‌ی بازی و بازیکنان) دارای تعریف و رفتار مستقل باشد.
- **ماژولار بودن (Modularity):** برنامه باید از بخش‌های مستقل تشکیل شود تا توسعه، نگهداری و گسترش آن در فازهای بعدی به‌سادگی انجام گیرد.
- **استفاده از الگوی MVC:** تفکیک لایه‌های داده (مدل)، منطق بازی (کنترلر) و نمایش (رابط کاربری متنی) برای افزایش خوانایی و انعطاف‌پذیری.
- **امکان ذخیره‌سازی و بارگذاری بازی:** کاربران باید بتوانند وضعیت فعلی بازی را در فایل ذخیره کرده و در زمان دلخواه مجدداً بارگذاری کنند.
- **قابلیت توسعه در آینده:** ساختار پروژه باید به‌گونه‌ای طراحی شود که افزودن قابلیت‌های جدید (مانند رابط گرافیکی، مهره‌های بیشتر یا قوانین جدید) در فازهای بعدی به‌سادگی امکان‌پذیر باشد.

معرفی مهره ها و ویژگی های آنها

• شاه (king)

هدف: اصلی ترین مهره در بازی، اگر شاه در معرض تهدید قرار گرفته باشد و هیچ حرکتی برای فرار نداشته باشد، این همان وضعیت نهایی بوده و به معنای باخت است!
الگوی حرکت : می تواند در هر جهت (عمودی، افقی یا مورب) تنها یک خانه حرکت کند.
ویژگی خاص: اگر مهره ی شاه در معرض تهدید باشد، بازیکن باید در حرکت بعدی آن را از حالت تهدید خارج کند. (کیش)
محدودیت : نمی تواند به خانه ای حرکت کند که توسط مهره ی دیگری مورد تهدید قرار گرفته است.

• وزیر (Queen)

نقش : قوی ترین مهره ی بازی
الگوی حرکت : می تواند در هر جهت (عمودی، افقی، مورب) هر تعداد خانه حرکت کند تا زمانی که مسیرش مسدود نشده باشد.
ویژگی های خاص : ترکیب توان حرکتی رخ و فیل در یک مهره؛ هم قدرت و هم انعطاف دارد.

• رخ (Rook)

نقش : مهره ی ای با حرکت مستقیم و قدرت بالا
الگوی حرکت : در جهت های عمودی و افقی، هر تعداد خانه حرکت کند تا زمانی که مسیرش مسدود نشده باشد.
ویژگی خاص : رخ همچنین با پادشاه ترکیب می شود تا تکنیکی به نام **Castling** یا قلعه رفتن (شاه قلعه) را اجرا کند **Castling** یا شاه قلعه یک حرکت ویژه است که در آن بازیکن، پادشاه خود را دو خانه روی صفحه به سمت یک رخ در همان رتبه حرکت می دهد و رخ را به سمت دیگر شاه می برد.



• فیل (Bishop)

نقش : مهره ای با حرکت مورب

الگوی حرکت : در مسیرهای مورب، هر تعداد خانه، تا زمانی که مانعی وجود نداشته باشد.
ویژگی خاص : شطرنج فقط به صورت مورب (قطری) روی صفحه حرکت می کند و می تواند هر تعداد خانه در همان مسیر حرکت کند، تا زمانی که مانعی (مهره ای) در مسیرش نباشد. هر فیل همیشه روی خانه هایی با رنگ یکسان باقی می ماند (یکی روی سفید، دیگری روی سیاه)

• اسب (Knight)

نقش : مهره ای با حرکت غیرمنطقی و تاکتیکی

الگوی حرکت : دو خانه در یک جهت (عمودی یا افقی) و سپس یک خانه عمود بر آن (L شکل)

ویژگی خاص : تنها مهره ای است که می تواند از روی مهره های دیگر بپرد.

• پیاده (Pawn)

نقش : مهره ای ابتدایی و تاکتیکی.

الگوی حرکت :

در حالت معمول یک خانه به جلو.

در اولین حرکت می تواند دو خانه به جلو حرکت کند.

در حالت حمله، فقط به صورت مورب یک خانه جلوتر مهره ای حریف را می زند.

ویژگی خاص ها:

اگر به آخرین ردیف برسد، می تواند تبدیل (Promotion) شود به وزیر، رخ، فیل یا اسب.

در شرایط خاص، می تواند حرکت آن پاسان (En Passant) انجام دهد.



مهره های خاص

• سرباز++

یکی از پیاده ها محسوب می شود که ناشناس بوده و قابلیت ویژه ای دارد، آن این است که علاوه بر دو خانه‌ی مورب توانایی تهدید خانه‌ی مقابل هم دارد.

• وزیر زره‌دار

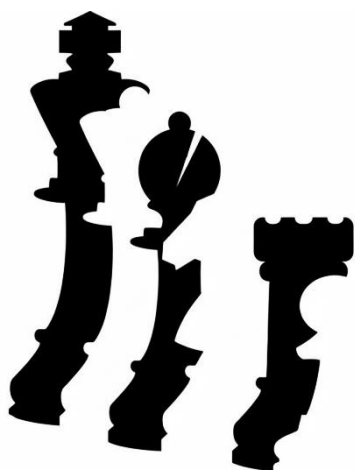
این وزیر با زره‌ای که دارد می تواند در مقابل حمله‌ی اول از خود محافظت کند اما در حمله‌ی دوم زره‌ی وزیر از بین رفته و با حمله‌ی بعدی کشته خواهد شد.

• جاسوس بی پدر

این مهره در واقع متعلق به حریف است، اما در ابتدا برای ما قابل تشخیص نیست. رنگ یا هویت اصلی آن پس از چند حرکت مشخص می‌شود. در هر زمانی که حریف بخواهد، می‌تواند از مهره‌ی جاسوس استفاده کرده و با آن حمله کند یا هویت واقعی آن را آشکار سازد.

• جوکر:

سرباز جوکر یک سرباز استاندارد است که دارای توانایی ویژه‌ی تقلید است. جوکر در طول بازی، می‌تواند دوبار در نوبت‌های دلخواه خود بازیکن رفتار و الگوی حرکتی هر مهره‌ی دیگر را تقلید کند. هر تقلید تنها برای همان نوبت معتبر است و پس از پایان نوبت، سرباز جوکر به حالت عادی خود باز می‌گردد.





فیچرهای منطق بازی و هوشمندی

- سیستم راهنمایی : دکمه‌ای برای پیشنهاد تمام حرکت های ممکن
- اعتبارسنجی حرکات : بررسی و توضیح حرکات غیرقانونی با پیام آموزشی (مثلاً:اسب نمی تواند این طور حرکت کند).
- سیستم مرور بازی : امکان برگشت چند حرکت و مرور کل بازی.
- بمب زماندار : اگر مهره‌ای بیش از n (به صلاح دید شما) حرکت ثابت بماند، منفجر شده و مهره‌های اطراف (به شعاع یک خانه) را نابود می کند.
- قتل عام : در صورت کشته شدن وزیرحریف، وزیر بازیکن برای آن نوبت می تواند دوبار حرکت کند یا از این قابلیت استفاده نکند.
- فاجعه : در هر چند نوبت، رویداد های فصلی فعال می شود

بهار:

در اثربارش زیاد باران زمین خیس و نرم شده است و حرکت سربازان سخت تر شده ،سربازان فقط میتوانند هر دو نوبت یکبار حرکت کنند.

تابستان:

افتاب شدید تابستان روی دید فیل اثر گذاشته و بیشتر از سه خانه در مسیر مجاز توانایی پیشروی ندارد.

پاییز:

به دلیل کمبود اذوقه ، اسب ها گرسنه مانده اند و به همین علت در هر دو نوبت یکبار میتوانند حرکت کنند.

زمستون:

برف و یخ زدگی مسیرهمیشگی رخ را کند کرده است. اکنون رخ فقط چهارخانه میتواند حرکت کند.

حالت های بازی

۱. مود استاندارد

۲. مود بازی بهینه (Energy Mode)

در این حالت، هر حرکت دارای هزینه انرژی است.

بازیکن با انجام حرکات مختلف، از انرژی خود مصرف می کند و در صورت اتمام انرژی، قادر به انجام حرکت نخواهد بود.

با زدن مهره های حریف، مقداری انرژی به بازیکن بازگردانده می شود.

هدف این مود، بازی بهینه و اقتصادی است؛ بازیکن باید تصمیم بگیرد که آیا ارزش دارد برای گرفتن یک مهره خاص، انرژی مصرف کند یا خیر.

می توان برای هر نوع حرکت (مثلاً حرکت وزیر یا رخ) هزینه های متفاوتی در نظر گرفت.

۳. مود مأموریت (Mission Mode)

در این حالت، بازی دارای اهداف و محدودیت های مشخص است و بازیکن باید در

تعداد حرکات محدود به آن ها دست یابد.

برخی از نمونه مأموریت ها می تواند شامل موارد زیر باشد:

در ۱۰ حرکت وزیر حریف را بزن.

شاه خود را تا خانه ی خاصی برسان.

ظرف تعداد محدودی نوبت، از وضعیت کیش خارج شو.

با موفقیت در انجام مأموریت، بازیکن برنده اعلام می شود؛ در غیر این صورت، بازی پایان

می یابد و شکست محسوب می شود.

```
=====  
Chess++  
=====
```

1. Standard Mode
2. Mission Mode
3. Optimization Mode
4. Load Previous Game
5. Exit

```
=====  
Please enter your choice: 
```

بخش نمایشی (Interface)

رابط کاربری این پروژه باید در محیط ترمینال (CLI) پیاده‌سازی شود. نمایش صفحه‌ی بازی باید تا حد ممکن خوانا و قابل درک باشد. رعایت موارد زیر توصیه می‌شود: نمایش صفحه‌ی ۸×۸ به صورت ماتریسی از خانه‌ها (به عنوان مثال با حروف و نماد هر مهره) استفاده از رنگ‌بندی متنی برای تفکیک بازیکنان (مثلاً سفید و سیاه یا دو رنگ متضاد دیگر) نمایش پیام‌های راهنما و خطا در طول بازی (مثلاً «حرکت غیرقانونی!» یا «کیش!») استفاده از نمادهای ساده برای مهره‌ها (مثلاً K برای King، Q برای Queen و...) در صورت تمایل، دانشجویان می‌توانند از کتابخانه‌هایی مانند [ncurses](#) یا [escape code](#) های ANSI برای رنگ‌بندی و افکت‌های متنی استفاده کنند (اختیاری).

Chess++ Board

```
      a  b  c  d  e  f  g  h
+---+---+---+---+---+---+---+---+
8 | BRO | BKN | BBI | BQA | BKI | BBI | BKN | BRO | 8
+---+---+---+---+---+---+---+---+
7 | BPA | BPA | BPA | BPA | BPA | BPA | BPA | BPA | 7
+---+---+---+---+---+---+---+---+
6 |     |     |     |     |     |     |     |     | 6
+---+---+---+---+---+---+---+---+
5 |     |     |     |     |     |     |     |     | 5
+---+---+---+---+---+---+---+---+
4 |     |     |     |     |     |     |     |     | 4
+---+---+---+---+---+---+---+---+
3 |     |     |     |     |     |     |     |     | 3
+---+---+---+---+---+---+---+---+
2 | WJK | WPP | WPA | WPA | WPA | WPA | WPA | WPA | 2
+---+---+---+---+---+---+---+---+
1 | WRO | WKN | WBI | WQA | WKI | WBI | WKN | WRO | 1
+---+---+---+---+---+---+---+---+
      a  b  c  d  e  f  g  h
```

Season: Spring

Enter your move (e.g., e2 e4):

بارمبندی و سیستم نمره‌دهی پروژه

- طراحی شیء‌گرا: (Object-Oriented Design) ۲۰ نمره
شامل تعریف صحیح کلاس‌ها، استفاده از وراثت، چندریختی و کپسوله‌سازی. رعایت اصول طراحی شیء‌گرا و تفکیک مسئولیت‌ها در کلاس‌ها مورد توجه است.
- منطق اصلی بازی و قوانین شطرنج استاندارد: ۱۵ نمره
پیاده‌سازی حرکات مهره‌ها، تشخیص وضعیت‌های کیش و مات، و جلوگیری از حرکات غیرقانونی.
- مهره‌ها و ویژگی‌های خاص: (Chess++) ۱۵ نمره
پیاده‌سازی صحیح مهره‌های جدید شامل سرباز++، وزیر زره‌دار، جاسوس بی‌پدر و جوکر، مطابق توضیحات پروژه.
- فیچرهای هوشمندی و منطق افزوده: ۱۵ نمره
شامل قابلیت‌هایی نظیر سیستم راهنما، مرور حرکات، فاجعه‌های فصلی، بمب زمان‌دار و سایر ویژگی‌های خلاقانه‌ی درون بازی.
- حالت‌های مختلف بازی: (Modes) ۱۵ نمره
پیاده‌سازی حالت‌های مختلف بازی به صورت کامل و قابل اجرا.
- رابط کاربری متنی: (CLI Interface) ۵ نمره
نمایش خوانا و واضح صفحه‌ی ۸×۸، نمایش نماد هر مهره، استفاده از رنگ یا تفکیک متنی بازیکنان و پیام‌های راهنما و خطا.
- قابلیت ذخیره و بارگذاری بازی: ۵ نمره
امکان ذخیره‌ی وضعیت فعلی بازی در فایل و بارگذاری مجدد در زمان دلخواه.

● ساختار ماژولار، کدنویسی تمیز و مستندسازی: ۱۰ نمره

طراحی پروژه به صورت ماژولار و قابل گسترش، به گونه‌ای که افزودن بخش‌های جدید (از جمله رابط گرافیکی در فاز بعدی) به سادگی انجام شود. رعایت اصول کدنویسی تمیز، استفاده از نام‌گذاری استاندارد، توضیح کافی در کدها و حفظ نظم در ساختار فایل‌ها نیز در این بخش مورد ارزیابی قرار می‌گیرد.

ددلاین و نحوه‌ی تحویل پروژه

تمام مراحل توسعه پروژه باید در یک مخزن **GitHub** انجام شود و تاریخچه **commit** ها در ارزیابی پروژه مورد بررسی قرار می‌گیرد.

هر تیم موظف است حداکثر تا تاریخ ۱ بهمن ماه، پروژه فاز اول را به‌صورت کامل در مخزن **GitHub** خود نهایی کند. در زمان ددلاین، تیم‌ها باید روی نسخه نهایی پروژه در **GitHub Tag** مناسب (مثلاً `phase1` یا `v1.0`) ایجاد کنند.

پس از ایجاد **Tag**، لینک مخزن **GitHub** به‌همراه اطلاعات زیر باید از طریق ایمیل ارسال شود:

- نام پروژه
- نام و نام خانوادگی تمام اعضای تیم
- شماره دانشجویی تمام اعضا
- لینک URL مخزن (**GitHub Repository**)

apbasu4041@gmail.com

آدرس ایمیل درس :

ایمیل باید با مشخصات زیر ارسال شود:

Subject: Project_Phase1

ارسال پس از ددلاین یا بدون **Tag** معتبر، به‌منزله‌ی تحویل ناقص تلقی خواهد شد. **تأکید مهم:** پروژه‌هایی که بدون استفاده از **GitHub** یا فاقد تاریخچه توسعه معتبر باشند، مورد ارزیابی قرار نخواهند گرفت.