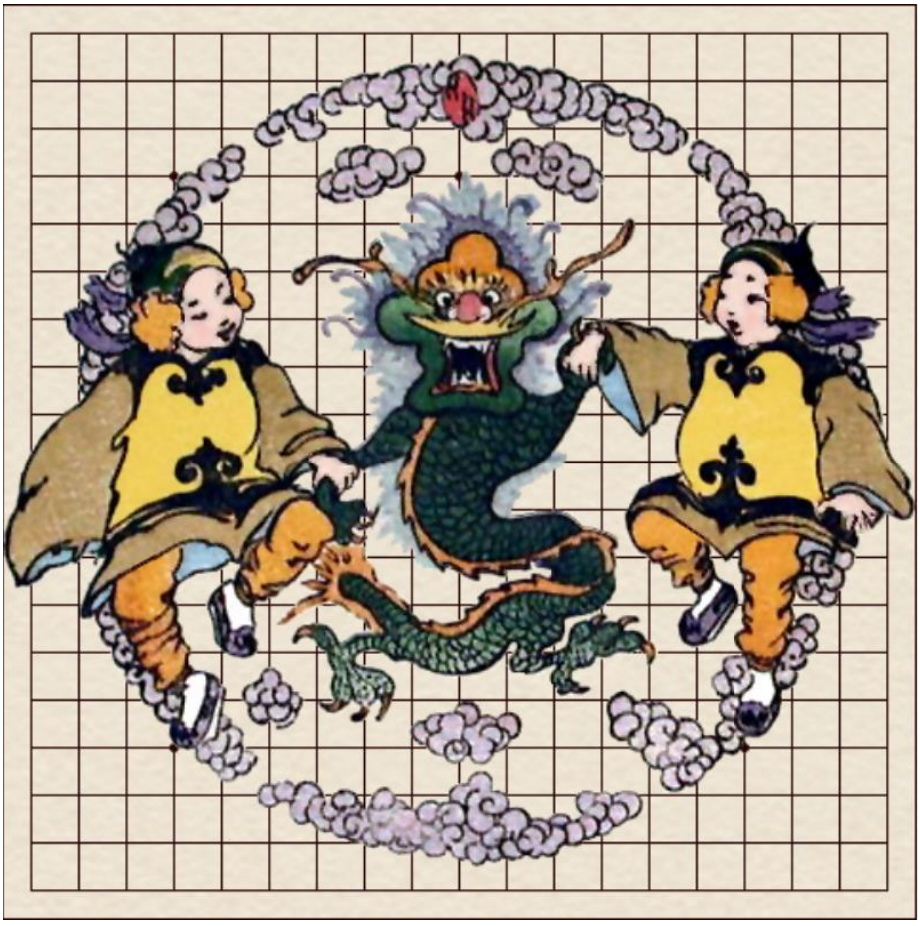
به نام حق

پروژه جستجوی خصمانه درس هوش مصنوعی

استاد درس: دکتر احد هراتی

پاییز ۹۷

Go

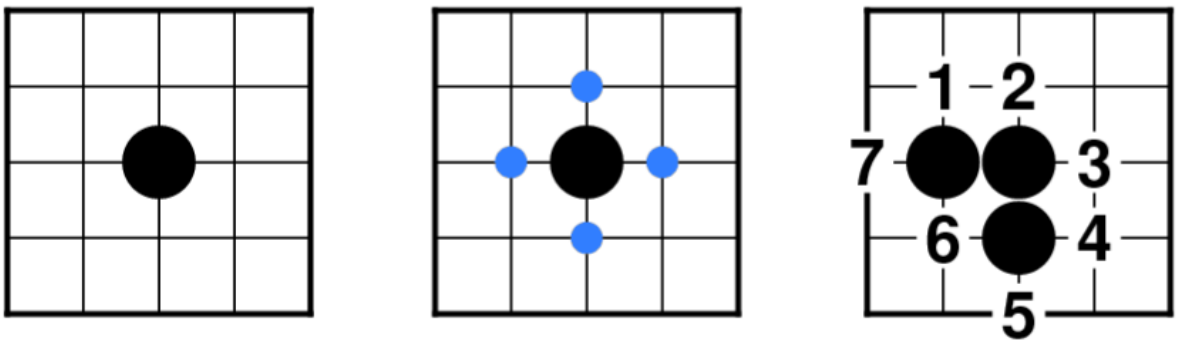


مقدمه

Go نام یک بازی تخته‌ای، مخاصمه‌ای و دونفره بسیار محبوب و قدیمی است که اولین بار بیش از ۲۵۰۰ سال پیش توسط کشور چین معرفی شد. صفحه بازی در حالت استاندارد شامل ۱۹ خط افقی و ۱۹ خط عمودی است که دو بازیکن به ترتیب مهره[1]های سیاه و سفید خود را در نقاط تقاطع[2] صفحه بازی قرار می‌دهند. تعداد کل مهره‌ها برابر با تعداد نقاط تقاطع صفحه است. حرکت دادن مهره‌های قرار داده شده در صفحه امکان‌پذیر نیست مگر اینکه توسط مهره‌های حریف به طور کامل محاصره شوند که در این‌صورت مهره‌های محاصره شده از صفحه بازی خارج می‌شوند. هدف اصلی این بازی محاصره و تصاحب سطح بیشتری از زمین بازی توسط مهره‌ها است.

قوانین بازی

بازیکن سیاه یک مهره بیشتر از بازیکن سفید دارد. ابتدا بازیکن سیاه بازی را شروع کرده و در هر نوبت هر بازیکن مهره رنگ خودش را در نقاط تقاطع صفحه بازی می‌گذارد. مهره‌هایی که از یک رنگ هستند و به صورت عمودی و یا افقی (و نه مورب) در کنار یکدیگر قرار می‌گیرند، تشکیل یک گروه[3] می‌دهند. نقاط تقاطع خالی مجاور یک مهره، نقاط آزادی[4] آن مهره تعریف می‌شود. مطابق با شکل زیر، مهره‌های یک گروه، نقاط آزادی را با هم شریک هستند. هنگامی که یک گروه از مهره ها توسط مهره‌های حریف محاصره شوند به طوری‌که تعداد نقاط آزادی آن گروه صفر شود، آن گروه از مهره‌ها از جدول بازی حذف شده و به عنوان مهره‌های زندانی حریف تلقی می‌شوند.

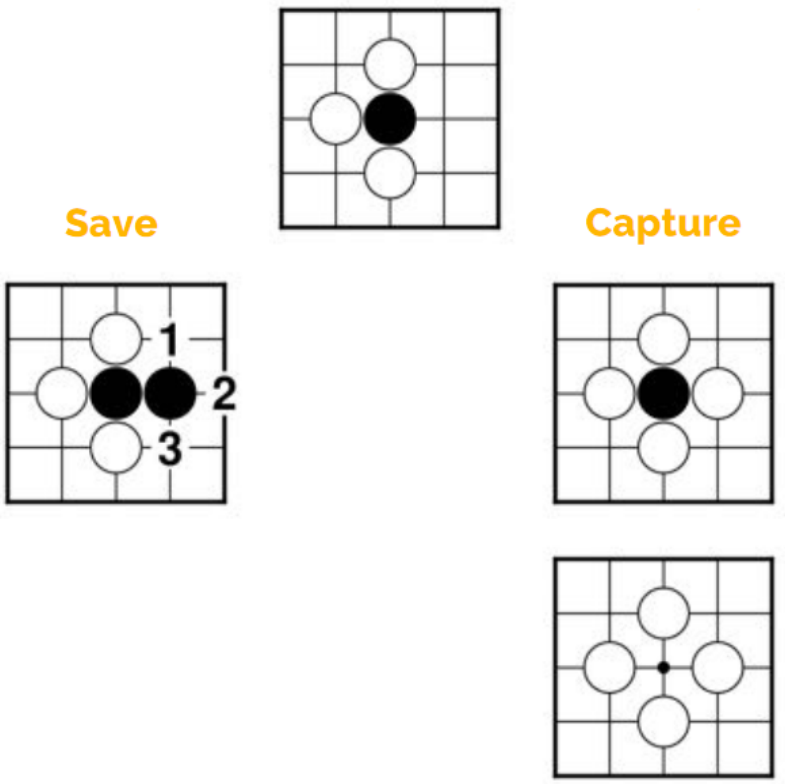


(a) (b) (c)

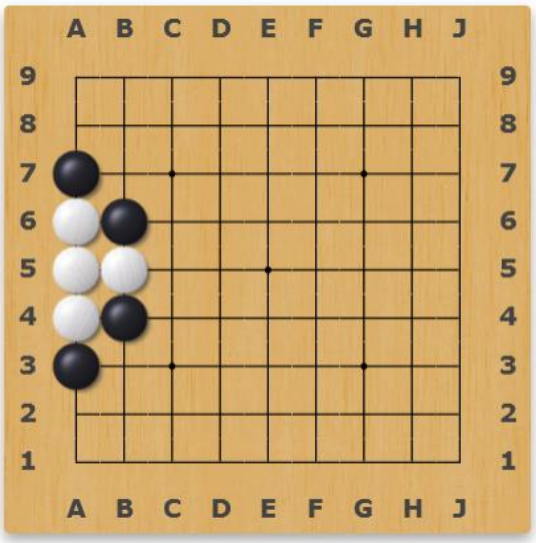
قانون Capturing : هر مهره بر روی صفحه بازی می‌بایست حداقل یک آزادی داشته باشد – و یا متصل به گروهی باشد که آن گروه دست‌کم یک آزادی دارد. به حالتی که در آن مهره یا گروهی از مهره‌ها، تنها یک آزادی دارند وضعیت Atari می‌گویند. هر مهره یا گروهی که در وضعیت آتاری قرار دارد، با ازدست دادن آخرین آزادی خود از صفحه بازی حذف می‌شوند.

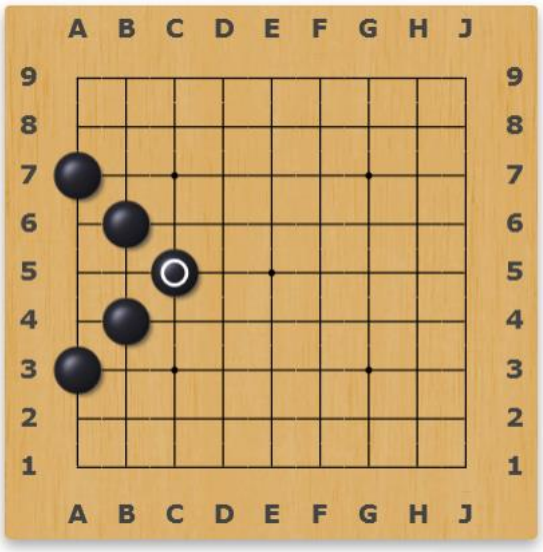
---

[1] Stone
[2] Intersection
[3] Group
[4] Liberty

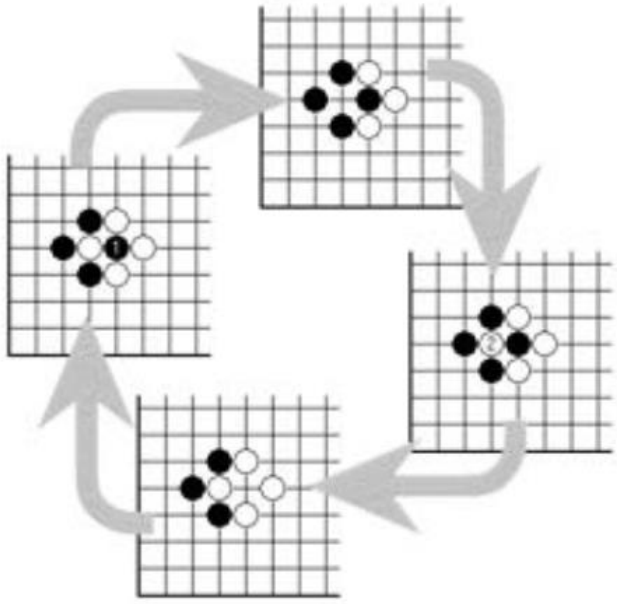


برای محاصره مهره‌هایی از حریف که در لبه صفحه بازی هستند، تنها لازم است از سمت فضای صفحه بازی محاصره شوند. به عنوان مثال، بازیکن سیاه می‌تواند گروه سفید را مطابق با شکل زیر محاصره کند.





قانون Ko : بازیکنان اجازه انجام حرکتی که بازی را به موقعیت قبلی بر گرداند را ندارند. این قانون مانع از تکرار بی‌پایان بازی می‌شود.



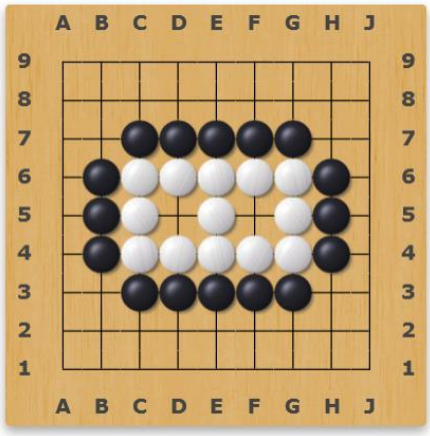
قانون Pass : ممکن است بازیکن در یک مرحله از بازی تصمیم بگیرد آن دور از بازی را بازی نکرده و مهره‌ای بر روی صفحه قرار ندهد. این حالت معمولا زمانی رخ می‌دهد که بازیکن به این نتیجه برسد که هیچ حرکت مفیدتری باقی نمانده است. اگر هر دو بازیکن به صورت متوالی Pass کنند، بازی به پایان رسیده و امتیازها شمرده می‌شود.

قانون Suicide : خودکشی به این معناست که یک بازیکن مهره‌اش را در مکانی از صفحه بگذارد که آزادی برای آن مهره صفر باشد و در همان لحظه مهره از صفحه بازی حذف شود. در این نسخه از بازی Suicide یک حرکت غیر مجاز است. اما درنظر داشته باشید که با گذاشتن یک مهره در صفحه، همیشه قبل از چک کردن وضعیت خودکشی، بایستی وضعیت محاصره شدن مهره‌های حریف ارزیابی شود.

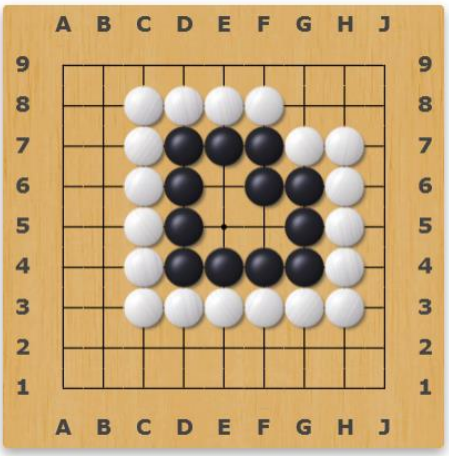
قانون Komi : از آنجایی که همیشه بازیکن سیاه بازی را آغاز می‌کند، در پایان بازی یک مقداری به امتیاز بازیکن سفید اضافه می‌شود که به این امتیاز اضافی Komi می‌گویند.

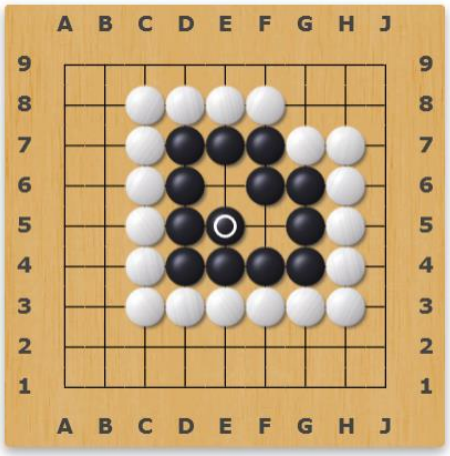
اصطلاح Eye : یک نقطه تقاطع خالی در داخل **یک** گروه که از همه طرف به مهره‌های **همان** گروه می‌رسد، چشم می‌گویند.

یکی از استراتژی‌های بازیکن‌ها در بازی Go ساخت گروه‌هایی با بیش از دو چشم است. چرا که اگر گروهی حداقل دو چشم داشته باشد، این گروه برای همیشه زنده مانده و هیچوقت محاصره نمی‌شود؛ زیرا حریف همزمان نمی‌تواند دو چشم را با مهره خود پر کند. به عنوان مثال در شکل زیر گروه سفید هیچگاه از صفحه بازی حذف نمی‌شود مگراینکه بازیکن سفید به اشتباه یکی از چشم‌های خود را با یکی از مهره‌هایش پر کند.



در شکل زیر، بهترین تصمیم برای بازیکن سیاه، ساخت دو چشم به منظور حفاظت از گروه خود می‌باشد.





نحوه امتیازدهی در پایان بازی :

بعد از اتمام مهره‌ها و یا در شرایطی که دو بازیکن پشت‌سر هم Pass کنند، بایستی تابع مربوط به امتیازدهی اجرا شود. قبل از امتیازدهی، مهره‌هایی که بر روی صفحه قرار دارند و از همه طرف به مهره‌های حریف می‌رسند و از طرفی با گسترش گروه خود نه امکان Capture کردن مهره‌های حریف را دارند و نه می‌توانند خود را به گروه‌های خودی دیگر متصل کنند، از صفحه بازی حذف می‌شوند. به این مهره‌ها اصطلاحا مهره‌های مرده[5] می‌گویند. سیستم امتیازدهی به روش Area Scoring است. بدین معنا که امتیاز هر بازیکن برابر است با تعداد مهره‌هایی که بازیکن بر روی صفحه دارد به علاوه تعداد نقاط تقاطع خالی که از هر طرف توسط این بازیکن محاصره شده است.

[5] Dead Stones

مسئله

در این پروژه تنها کافی‌است الگوریتم MiniMax و هرس آلفا و بتا را به کد Base در فایل core.cpp اضافه کرده و با استفاده از آن در هر نوبت از بازی در مورد اکشن بعدی تصمیم‌گیری کنید. توجه داشته باشید که با توجه به پیچیدگی زمانی و فضایی این بازی در الگوریتم MiniMax رسیدن به حالت‌های پایانی در درخت MiniMax ممکن نیست. بنابراین بایستی درخت جستجو را تا عمق خاصی پیمایش کنید و سپس با استفاده از یک یا چند هیوریستیک، میزان سودمندی گره‌های غیرپایانه در این عمق را تخمین بزنید. برای این‌کار می‌توان برای بازی تعدادی ویژگی را در نظر گرفته و به هرکدام یک ارزش نسبت داد. سپس با ترکیب ارزش‌های ویژگی‌ها، می‌توان سودمندی یک وضعیت خاص را تخمین زد.

نکات

- ابعاد صفحه بازی ۹ × ۹ است.
- Suicide یک اکشن غیرمجاز بوده و مقدار Komi برابر با 6.5 است.
- زمان پردازش هر برنامه ۱۵ دقیقه خواهد بود.
- توابع موردنیاز برای پیاده‌سازی قوانین بازی در پیوست آمده است.
- تنها فایل‌های درون پوشه core مجاز به تغییر هستند. درصورت نیاز به ایجاد فایل‌های جدید بایستی آنها را در همین پوشه قرار دهید.
- برای تحویل این پروژه، بایستی ابتدا از طریق این لینک Repository شخصی خودتان را در تمرین ایجاد شده در سایت GitHub Classroom ایجاد کرده و فایل های مربوطه را در این مخزن Commit و Push کنید. تنها کافی‌است یک نفر از اعضای تیم از طریق لینک فوق یک Repository بسازد و سپس می‌تواند هم‌تیمی خودش را به پروژه اضافه کند.
- زمان تحویل پروژه، متعاقبا در توسط گروه حل‌تمرین اعلام می‌شود.
- زمان برگزاری **لیگ مسابقات Go** در دانشکده مهندسی نیز متعاقبا توسط گروه حل‌تمرین اعلام می‌شود. ۱۰ تیم برتر مسابقات نمره اضافه کسب خواهند کرد.
- پیاده‌سازی پروژه به صورت **گروهی دو نفره** است و تحویل نهایی آن به صورت حضوری می‌باشد.
- کپی‌برداری و یا عدم تسلط در ارائه پروژه، نمره 0 در پی خواهد داشت.
- برای ارتباط با گروه حل‌تمرین می‌توانید از طریق یکی از ایمیل‌های Hosein.mohebbi75@gmail.com و یا Benyaminbashari@gmail.com اقدام کنید. همچنین می‌توانید سوالات مربوط به پروژه را در سایت کویرا و در بخش پرسش‌وپاسخ مطرح نمایید.

پیوست:

به‌منظور عدم نیاز به پیاده‌سازی قوانین بازی، بخشی از پروژه GNU Go در این پروژه به عنوان کتابخانه استفاده شده است که در ادامه توضیحات لازم برای استفاده از آن ارائه می‌شوند.

GTP[6]

GTP نام پروتکل متنی استفاده شده در کد Base است که کدهای کلاینت می‌توانند از طریق آن به برنامه سرور و مانیتور متصل شوند.

Base Go

کد Base پیوست شده در این پروژه، یک نسخه ساده شده از پروژه بزرگ GNU Go می‌باشد که توابع مربوط به پیاده‌سازی قوانین بازی و ارتباط برنامه با برنامه سرور و مانیتور از طریق پروتکل GTP به طور کامل پیاده‌سازی شده است.

آماده‌سازی و کامپایل کد Base Go

از آنجایی‌که کد بیس به زبان C نوشته شده است، برای کامپایل کد نیاز به CMake دارید. در صورتی که از سیستم‌عامل مبتنی بر Unix استفاده می‌کنید، برای کامپایل کد مراحل زیر را طی کنید:

ابتدا فایل مربوط به BaseGo.zip را از حالت فشرده خارج کرده و مسیر داخل پوشه BaseGo را معادل با عبارت path-to-source در نظر بگیرید و سپس دستورات زیر را در ترمینال وارد نمایید:

```
cd <path-to-source>
mkdir build
cd build
cmake ..
make
```

برای کامپایل کد در دفعات بعدی تنها کافی‌است در مسیر فعلی دستور make را دوباره اجرا نمایید.

به‌منظور آسان‌تر شدن برنامه‌نویسی و کامپایل کد و عدم نیاز به دستورات بالا حتما از یک IDE مناسب مانند Qt (برای ویندوز، لینوکس و مک) استفاده نمایید.

تنها فایل‌های مجاز برای تغییر، فایل‌های داخل پوشه core هستند. می‌توانید از طریق فایل info.h نام تیم خود را مشخص کنید. فایل‌های core.h و core.cpp شامل دو تابع اساسی زیر می‌باشد:

```
int play(int color);
```

این تابع یک عدد صحیح به عنوان color در ورودی دریافت می‌کند که بدین‌معناست که شما بایستی به عنوان بازیکنی با این رنگ بازی کنید. تمامی توابع و پیاده‌سازی‌های شما بایستی در این تابع فراخوانی شده و در پایان یک حرکت مجاز را به عنوان خروجی تابع برگردانید.

[6] Go Text Protocol

```
void reset();
```

این تابع یک بار و تنها زمانی که یک بازی جدید ایجاد می‌شود فراخوانی می‌شود. بنابراین از این تابع می‌توانید برای مقداردهی اولیه متغیرها و یا ایجاد هرآنچه که در طول برنامه نیاز دارید استفاده کنید.

برای اضافه کردن فایل جدید، حتما فایل ایجاد شده را در پوشه core قرار داده و نام این فایل را مطابق با دستور زیر در فایل CMakeLists.txt موجود در همین پوشه اضافه نمایید:

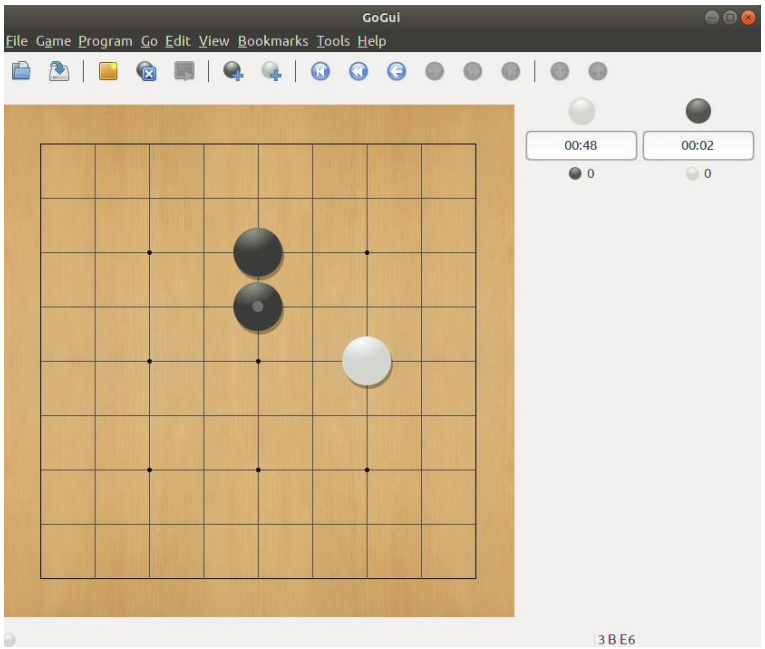
```
set(SOURCE_FILES core.cpp core.h info.h <your-new-file>)
```

فایل board.h

این فایل که در پوشه engine قرار دارد شامل توابعی مفید و از قبل آماده است که اطلاعاتی در مورد صفحه بازی، ویژگی‌های هر حرکت(اعم از مجاز و یا غیرمجاز بودن، خودکشی و ...)، مهره‌ها و مقدار آزادی (Liberty) آن‌ها به شما می‌دهد.

GoGui

GoGui یک برنامه گرافیکی برای داوری مسابقه و نمایش بازی Go می‌باشد. این برنامه نیازی به نصب ندارد و برای اجرا تنها به جاوا نسخه حداقل ۵ نیاز دارد. درصورتی که از سیستم‌عامل مبتنی بر Unix استفاده می‌کنید، فایل GoGui در پوشه bin را اجرا نمایید و در صورتی که از ویندوز استفاده می‌کنید فایل GoGui در پوشه lib را اجرا نمایید.



بعد از باز شدن برنامه شما می‌توانید با استفاده از زبانه Game یک بازی جدید ایجاد کنید. همچنین می‌توانید فایل اجرایی تیم خودتان را از طریق زبانه Program > Attach یا Program > New Program به GoGui متصل کنید. در صورتی که برای اولین بار این کار را انجام می‌دهید لازم است مسیر فایل اجرایی را نیز وارد نمایید.

موفق باشد