

رویداد اشاره گرهای هوشمند

باسمه تعالی

فایلی که برای پیاده سازی هوش به عنوان کلاینت در اختیار شما قرار داده می شود شامل قسمت های زیر است:

- فایل CONST.py
- فایل data_manager.py
- فایل io_manager.py
- فایل client.py

فایل CONST.py شامل تنظیمات بازی و برخی ثابت ها است، خط ۲ و ۳ این فایل برای تعیین آی پی و پورت ارتباطی سرور و کلاینت است که باید در سرور و کلاینت یکسان باشند، دیگر قسمت های کد نیازی به تغییر ندارد و در صورت تغییر ممکن است کلاینت به درستی کار نکند. فایل های data_manager.py، io_manager.py برای مدیریت پیام هایی است که بین سرور و کلاینت تبادل می شود، این فایل ها نباید تغییر داده شوند.

فایل client.py برای پیاده سازی هوش بازی و همچنین ارسال حرکت انجام شده به سرور است، کلاس AI وظیفه اجرای تابع do_turn را که در هر بار نوبت بازی اجرا می شود را داراست، شما می توانید به هر طریق که نیاز دارید کلاس AI را تکمیل کنید و یا از کلاس های دیگری برای پیاده سازی هوش استفاده کنید. برای دریافت تغییرات و یا تمام برد می توانید از دستور self.dm.get_wm() در تابع do_turn استفاده کنید که wm دریافتی هم شامل برد و هم شامل آخرین تغییرات اعمال شده است. در انتها و پس از تصمیم به حرکت می توانید با صدا زدن تابع self.io_manager.move و دادن ورودی مناسب حرکت انتخابی خود را به سرور ارسال کنید

قسمت انتهایی فایل client.py که تابع main و اجرای کد ها را بر عهده دارد شامل دو while است که اولی تا هنگام شروع بازی اجرا می شود و دومی از تابع do_turn استفاده کرده و برای حالت های مختلف بازی تصمیم لازم را میگیرد، شما می توانید در داخل while های این قسمت حالت های احتمالی خطای ارسال حرکت اشتباه (error_number) را که آی دی های ۳ و ۴ هستند (این آی دی ها در حالتی ارسال می شوند که سرور حرکت فرستاده شده را اشتباه بداند، توضیحات این آی دی ها در فایل CONST.py قرار داده شده است) مدیریت کنید و یا هر تغییری را که به اجرای بهتر کد کمک می کند اعمال کنید.

بنابراین به طور کلی ۲ قسمت وجود دارند که لازم است برای ایجاد یک کلاینت کامل هوشمند تکمیل شود

۱. کد های مرتبط با هوش در فایل کلاس AI

۲. قسمت انتهایی فایل client.py برای برون رفت از ارسال حرکت اشتباه به سرور مدیریت خطاها و استفاده از زمان های مشغول بودن حریف

و سرور