فاز پنجم و نهایی پروژه بتل شیپ – Let’s Play !

با توجه به اینکه روز چهارشنبه، زمان زیادی صرف پروژه شد و بخش های بیشتری به نسبت جلسات قبل پیاده سازی شد، و اینکه در این فاز که فاز نهایی پروژه است نیاز است که پروژه خود را تکمیل کرده و آماده تحویل کنید، تصمیم گرفتیم آخرین نسخه از پیاده سازی پروژه در کلاس را همراه با این پی دی اف برایتان در سایت آپلود کنیم و توضیحات مربوط به آن ها را نیز به طور خلاصه در این بخش مرور کنیم. سپس بخش‌های مورد نیاز برای تکمیل پروژه را توضیح می دهیم و در نهایت به سراغ بخش‌های امتیازی پروژه می رویم.

زمان تحویل پروژه متعاقبا اعلام خواهد شد.

آنچه در کلاس انجام دادیم: در جلسه گذشته، ابتدا به سراغ روشی رفتیم تا بتوانیم کنسول را پاک کنیم تا محیط اجرای برنامه کمی تمیزتر و مرتب تر شود. برای پاک کردن کنسول ویندوز، نیاز است با سیستم عامل ارتباط برقرار کنیم. پس پکیج os را ایمپورت میکنیم. سپس با تابع system میتوانیم دستورات مورد نیاز را به ویندوز بدهیم. دستور cls در ترمینال ویندوز، کنسول را پاک می‌کند. بنابراین با عبارت os.system(‘cls’) کنسول ویندوز پاک خواهد شد. اما در IDE های دیگر مانند PyCharm یا VSCode، باید روش دیگری پیدا کنیم. برای این کار، از پکیج PyAutoGUI استفاده میکنیم. تابع hotkey که در این پکیج قرار دارد، دکمه های کیبورد را به عنوان ورودی میگیرد و عملیات مربوط به فشردن این دکمه ها را انجام می‌دهد. پس اگر در IDE مورد استفاده ما، shortcut ای برای پاک کردن کنسول باشد، میتوانیم این shortcut را به hotkey بدهیم، و اگر shortcut برای پاک کردن کنسول وجود نداشت، از تنظیمات IDE، خودمان shortcutای برای آن تعریف میکنیم، و سپس آن را به hotkey میدهیم.

سپس به سراغ روشی رفتیم تا بتوانیم میان اجرای خط های برنامه، کمی مکث ایجاد کنیم و در بخش هایی از برنامه مثل بخشی که می خواهیم نقشه نتیجه را به کاربر نشان دهیم، از این مکث استفاده کنیم. برای این کار از تابع sleep که در پکیج time است، استفاده می کنیم. این تابع، مدت زمان مکث را به ثانیه می گیرد و مکث را ایجاد می کند.

سپس به سراغ متود check_if_destroyed کلاس Ship رفتیم. در این متود نیاز است به نقشه بازیکنی که صاحب این کشتی است، دسترسی پیدا کنیم. پس owner را به عنوان پارامتر، دریافت میکنیم. سپس با توجه به جهت کشتی، خانه های مربوط به کشتی را روی نقشه بازیکن پیمایش میکنیم و اگر خانه ای با وضعیت FULL یافتیم، متوجه میشویم این خانه مورد شلیک واقع نشده و کشتی هنوز تخریب نشده است. اما اگر پیمایش‌مان تمام شد و هیچ خانه FULLی نیافتیم، متوجه می شویم کشتی منهدم شده‌است.

سپس متود destroy کلاس Ship را پیاده سازی کردیم. در این متود هم نیاز است به نقشه بازیکن صاحب نقشه دسترسی پیدا کنیم هم به امتیازات حریف، پس owner و enemy را به عنوان پارامتر می گیریم. ابتدا با توجه به جهت کشتی، روی نقشه بازیکن پیمایش میکنیم و وضعیت تمام خانه های کشتی را به COMPLETED تغییر می دهیم. سپس با توجه به سایز کشتی، امتیاز مناسب را به بازیکن حریف می دهیم. برای کشتی با سایز ۵ امتیاز ۵، سایز ۳ امتیاز ۸، سایز ۲ امتیاز ۱۲، سایز ۱ امتیاز ۲۵ داده خواهد شد. در آخر نیز کشتی از لیست ships بازیکن حذف خواهد شد.

در نهایت متود display_scores کلاس Game را پیاده سازی کردیم. در این متود صرفا امتیازات دو بازیکن چاپ می شود.

در متود play کلاس Player بعد عملیات shoot، روی لیست کشتی ها پیمایش کردیم و منهدم شدن آن ها را چک کردیم و در صورت نیاز آن ها را destroy کردیم.

در متود run کلاس Game، قبل هر بار insert_ship کنسول را پاک کردیم. سپس در حلقه اصلی، کنسول را پاک کردیم و بازیکن اول نوبتش را play کرد. باز کنسول را پاک کردیم و نقشه نتیجه را نمایش دادیم و ۲ ثانیه مکث کردیم. سپس همین فرایند را برای بازیکن دوم انجام دادیم. بالای صفحه نیز همواره امتیازات دو بازیکن را نمایش میدهیم.

آنچه نیاز است برای تکمیل پروژه انجام دهید: متودهای check_if_finished و finish کلاس Game را پیاده سازی کنید. سپس در حلقه اصلی run، بعد هر نوبتی که بازی می شود و نقشه نتیجه آن نمایش داده می شود، پایان بازی را چک کنید و در صورت نیاز، بازی را تمام کنید. (راهنمایی برای check_if_finished: کشتی ها هنگام انهدام از لیست ships بازیکن حذف می شدند.)

بدنه اصلی برنامه را از نو بسازید. یک لیست به نام players بسازید. سپس حلقه مربوط به نمایش منو را که در فاز های قبلی کامنت بود، از کامنت دربیاورید. در تابع مربوط به play with a friend لیست بازیکنان موجود در لیست players را نمایش دهید و از کاربر بخواهید دو بازیکنی که می خواهند بازی کنند را انتخاب کند. سپس یک Game جدید با دو بازیکن انتخاب شده بسازید و بازی را شروع کنید. اگر تعداد بازیکن های players کمتر از ۲ بود، پیغام مناسب چاپ شود و بازی ای ساخته نشود.

در تابع play with a bot صرفا یک پیغام چاپ کنید که این بخش در دست ساخت است :)

در تابع create a new user، نام یوزر جدید را از کاربر بگیرید. سپس یک Player بسازید و آن را به لیست players اضافه کنید. اگر قبلا یوزری با این نام در لیست players وجود داشت، پیغام مناسب چاپ شود و دیگر Player جدید ساخته نشود.

در تابع scoreboard، لیست بازیکنان موجود در لیست players را همراه با امتیاز های آن ها، چاپ کنید.

آنچه به عنوان بخش امتیازی میتوانید انجام دهید:

- در تابع show_scoreboard، لیست بازیکنان برحسب امتیازات آن ها مرتب شود و سپس نمایش داده شود.
- لیست players در یک فایل متنی ذخیره شود. سپس ابتدای هر بار اجرای برنامه، این لیست از فایل خوانده شود.
- مجاور نبودن دو کشتی را هنگام insert_ship چک کنید تا میان دو کشتی مختلف، حداقل یک خانه فاصله باشد. همچنین در هنگام انهدام کشتی، وقتی خانه های مربوط به آن به C تغییر پیدا میکنند، خانه های مجاور آن نیز به W تغییر پیدا کنند. به مثال زیر که دو نوبت پیاپی بازی را نشان می دهد، توجه کنید:

```
        Taha: 4 Bot: 14
   1  2  3  4  5  6  7  8  9  10
A  -  -  -  -  -  -  -  -  -  -
B  E  E  E  E  -  -  -  -  -  -
C  -  -  -  -  -  -  -  -  -  -
D  -  -  -  -  -  -  -  -  -  -
E  -  -  -  -  -  -  -  -  -  -
F  -  -  -  -  -  -  -  -  -  -
G  -  -  -  -  -  -  -  -  -  -
H  -  -  -  -  -  -  -  -  -  -
I  -  -  -  -  -  -  -  -  -  -
J  -  -  -  -  -  -  -  -  -  -
Taha, your turn :
```

```
        Taha: 10        Bot: 14
   1  2  3  4  5  6  7  8  9  10
A  W  W  W  W  W  W  -  -  -  -
B  C  C  C  C  C  W  -  -  -  -
C  W  W  W  W  W  W  -  -  -  -
D  -  -  -  -  -  -  -  -  -  -
E  -  -  -  -  -  -  -  -  -  -
F  -  -  -  -  -  -  -  -  -  -
G  -  -  -  -  -  -  -  -  -  -
H  -  -  -  -  -  -  -  -  -  -
I  -  -  -  -  -  -  -  -  -  -
J  -  -  -  -  -  -  -  -  -  -
Taha, your turn :
```

- بازی با بات را پیاده سازی کنید. یک کلاس به نام Bot بسازید که از Player ارث بری میکند. سپس متود insert_ship و shoot آن را override کنید. با این کار دیگر نیازی به تغییر ساختار کلاس Game نیست. (اندرمزایای پلی مورفیزم 😊)

فایل کد نهایی خود را با نام **P5_familyName.py** ذخیره کرده و در قسمت فاز پنجم و نهایی پروژه ارسال کنید.

موفق باشید – فوق برنامه پایتون پیشرفته ۲ دهم دبیرستان (دوره ۲) – تابستان ۱۴۰۰