

به نام خداوند بخشنده مهربان

مار سیب‌خوار



۱. هدف پروژه

هدف از این پروژه آشنایی با API ایست که در اختیار لایه application قرار می‌گیرد تا با بقیه هاست‌های شبکه صحبت کند. هدف ثانویه‌ی این پروژه آشنایی با نرم افزارهایی‌ست که قسمت اصلی پردازش‌ها سمت سرور انجام و نتیجه برای کاربر ارسال می‌شود. پروژه در قالب یک بازی که اکثراً با آن آشنایی دارند تعریف شده تا درک آن راحت‌تر شود.

۲. توضیح بازی مار و سیب

اگر با این بازی آشنایی دارید می‌توانید این بخش را رد کنید. هدف بازی این است که مار به سیب برسد، آن را بخورد و با این کار امتیاز بگیرد. در بازی موانعی مثل دیوارهای میانی، دیوارهای دور نقشه و خود مار وجود دارند. مار در یک بازه زمانی مشخص، یک حرکت انجام می‌دهد و اگر کاربر چیزی وارد نکرده باشد به مسیر قبلی ادامه می‌دهد. در نسخه مد نظر ما قرار است چند کاربر هم زمان در یک نقشه بازی کنند و برای بدست آوردن یک سیب به رقابت بپردازند. در ادامه توضیح مبسوط آن خواهد آمد.

۳. نیازمندی‌هایی که باید انجام شوند

قبل از توضیح لازم به ذکر است، خیلی از مواردی که به آن‌ها اشاره نشده، سلیقه‌ای و مناسب درس‌هایی چون برنامه‌سازی پیشرفته و گرافیک است. ولی هدف ما بحث شبکه‌ای این بازی، چند کاربره بودن و پردازش‌های سمت سرور است. برای همین در مواردی که به صورت مستقیم گفته نشده است، شما آزادید که به اختیار خود عمل کنید و هر سناریویی را که صلاح می‌دانید، پیاده‌سازی کنید.

بازی با یک کاربر شروع می‌شود، اما بازی متوقف می‌ماند تا کاربر دیگری وصل شود با وصل شدن کاربر دوم بازی شروع می‌شود و بازیکنان رقابت می‌کنند تا سیب را تصاحب کنند. (بقیه کاربران به صورت پویا اضافه می‌شود و منتظر آن‌ها نمی‌مانیم). نکته‌ای که باید به آن توجه کنید این است که کاربر مار را در شروع بازی همیشه در گوشه‌ی بالا و سمت چپ صفحه می‌بیند و برای کاربران مختلف باید نقشه‌ها را بچرخانید تا این منظور عملیاتی شود.

در یک زمان مشخص (در حدود چند صد میلی ثانیه) هر مار یک حرکت انجام می‌دهد و اگر کاربر با فشردن کلید مسیر مار را عوض نکند مار همان مسیر قبلی را طی خواهد کرد. موانع تنه‌ی مار، دیوارهای داخلی و دیوار دور نقشه هستند که در مواجهه با هر کدام باید از سناریوهای زیر تبعیت کنید:

- اگر ماری به دیوارهای داخلی برخورد کند، می‌سوزد.
- برای برخورد مار به دیوارهای دور نقشه می‌توان دو سناریو در نظر گرفت: حالت اول، مار از دیوار عبور کند و در طرف دیگر نقشه ظاهر شود. حالت دوم، دیوار را به عنوان مانع در نظر بگیریم. یکی از این سناریوها را پیاده‌سازی کنید.
- برای برخورد به دیگر مارها می‌توانید از آن‌ها رد شوید و به عنوان جریمه به طول مار اضافه کنید یا این که دیگر مارها را به عنوان مانع در نظر بگیرید و بر خورد به آن‌ها باعث حذف کاربر شود.
- برای برخورد به تنه‌ی مارِ خودمان هم یکی از سناریوهای قبل را انتخاب کنید.

اگر یک کاربر یک سیب بخورد، به طول بقیه‌ی کاربران اضافه می‌شود. در شرایطی که دو مار همزمان به یک سیب برسند ماری که در زمان ظاهر شدن سیب فاصله‌ی بیشتری از آن داشته^۱، صاحب سیب محسوب می‌شود. در صورتی که فاصله‌ها یکی بود، سیب به ماری می‌رسد که طولش کم‌تر بوده است. اگر شرایط طوری شد که فاصله‌ها و طول‌ها یکی بودند، بازی را ببندید، به جای خلوتی بروید و به شانس خود فکر کنید (:

^۱ فاصله منتهن یا A* (برای فاصله منتهن به این آدرس و برای A* به این آدرس مراجعه کنید)

بازی با خورده شدن پانزدهمین سیب به اتمام می‌رسد و ماری که طولش از بقیه کم‌تر باشد به عنوان برنده اعلام می‌شود (کافیست در خروجی سرور نام کاربری آن شخص نوشته شود) توجه کنید که بازی حداکثر با ۴ کاربر اجرا می‌شود و طبیعتاً هیچ‌گاه دو نقشه هم زمان در حال اجرا نخواهند بود. برای این که ببینیم کاربر فعالیت می‌کند یا نه، یک سیگنال به نام ضربان قلب از طرف کاربر در بازه‌های زمانی مشخص برای سرور فرستاده می‌شود، که سرور با دریافت این پیام می‌فهمد که کاربر هنوز هم مشغول بازی کردن است. (برای راحتی کار، این پیام‌ها در یک سوکت جداگانه فرستاده می‌شوند).

۴. توضیح فرمت نقشه و ارسال پیام‌ها

برای انجام پروژه به شما قطعه کدی داده شده است، در ادامه به توضیح نحوه اجرای آن می‌پردازیم. می‌توانید به دلخواه خود در این کد تغییر ایجاد کنید، اما حتماً باید با دستیاران در این باره مشورت کنید. در آغاز برنامه نام کاربری را از کاربر در کنسول دریافت می‌کنیم.^۲ در قدم بعد، به سرور متصل می‌شویم و اتصالی برای ارسال ضربان قلب درست می‌کنیم. (در بخش بعد نحوه ارسال آن را توضیح می‌دهیم)

۱.۴ نحوه ارسال ضربان قلب

بعد از دریافت نام کاربری یک اتصال به پورت ۱۲۳۴۵ سرور به آدرس ۱۲۷.۰.۰.۱ برقرار می‌شود و هر ۵ ثانیه^۳ یک پیام JISON با فرمت زیر برای سرور ارسال می‌شود تا به سرور اطلاع دهد که کاربر فعال است.

```
temp = {
    'username': username,
    'message': 'HeartBeat'
}
```

شکل دوم: قالب ارسال پیام ضربان قلب

۲.۴ نحوه دریافت و ارسال پیام‌های کنترلی

بعد از اتصال ضربان قلب اتصال دیگری به سرور برقرار می‌شود (با همان شماره پورت و IP) و از طریق آن حرکات کاربر ارسال و پیام‌های سرور دریافت می‌شود. اولین چیزی که خوانده می‌شود اطلاعات نقشه یعنی اندازه آن و محل قرار گیری دیوارهای میانی است.^۴ در شکل سوم قالب پیام دریافت شده از سرور را می‌بینید.

^۲ لزومی نداد یکتا بودن نام کاربری بررسی شود فرض کنید نام‌های کاربری یکتا وارد می‌شود.

^۳ می‌توانید با تغییر متغیری که در ابتدای این کد تعریف شده است این بازه را تغییر دهید.

^۴ نقشه را باید از فایلی که به شما داده شده است بخوانید.

```
board = {
    'size': 5,
    'obstacles': [[1, 0], [0, 2], [2, 3]]
}
```

شکل سوم : قالب ارسال مشخصه‌های ثابت نقشه

بعد از خواندن اطلاعات ثابت اطلاعات اولیه‌ی اجسام متحرک نقشه را از سرور می‌خوانیم. سرور در پاسخ مکان اولیه‌ی مارها و اولین سیب را باز می‌گرداند. (طول اولیه‌ی مارها سه است)

```
moving_object_of_board = {
    'snakes': [[1, 2], [1, 3], [1, 4]], [[3, 2], [3, 3], [3, 4]],
    'food': [3, 3],
}
```

شکل چهارم : قالب ارسال مکان اولیه‌ی اجسام متحرک (مکان اولیه‌ی مارها درست نیست)

بعد از گرفتن اطلاعات اولیه نقشه بازی نمایش داده می‌شود. بعد از این، تنها از سرور بخش‌هایی را می‌خوانیم که باید بروز رسانی شوند. این بخش‌ها شامل موقعیت مارها، سیب و خانه‌هایی است که باید پاک شوند. برای مثال در پیام زیر به کلاینت موقعیت‌های دو مار و سیب را می‌دهیم، همچنین می‌گوییم که چه خانه‌هایی باید پاک شوند.

```
moving_object_of_board = {
    'snakes': [[1, 2], [1, 3], [1, 4]], [[3, 2], [3, 3], [3, 4]],
    'food': [3, 3],
    'remove': [[1, 1], [3, 1]]
}
```

شکل پنجم : قالب ارسال مکان به روز رسانی‌های نقشه

تا کنون تمام پیام‌هایی که از سرور برای ما ارسال می‌شوند را دیده‌ایم. کاربر تنها هنگامی که کلیدی را فشرد، پیام برای سرور ارسال می‌شود. (فرمت این پیام در شکل ششم آمده است) key می‌تواند یکی از سه مقدار down, up, right, left را بگیرد. سرور در صورت مجاز بودن حرکت جهت مار را به آن سمت تغییر می‌دهد.

۵. بخش‌بندی پروژه

پروژه به دو بخش تقسیم شده است و در زمان تحویل هر بخش را یکی از اعضا به صورت تصادفی تحویل می‌دهد. به همین دلیل تسلط بر روی تمام بخش‌های پروژه برای هر دو نفر اجباری است. بخش اول شامل پیاده‌سازی درست ضربان قلب، عملیات‌های مربوط به آن و پردازش و تهیه نقشه برای هر نفر است. بخش دوم

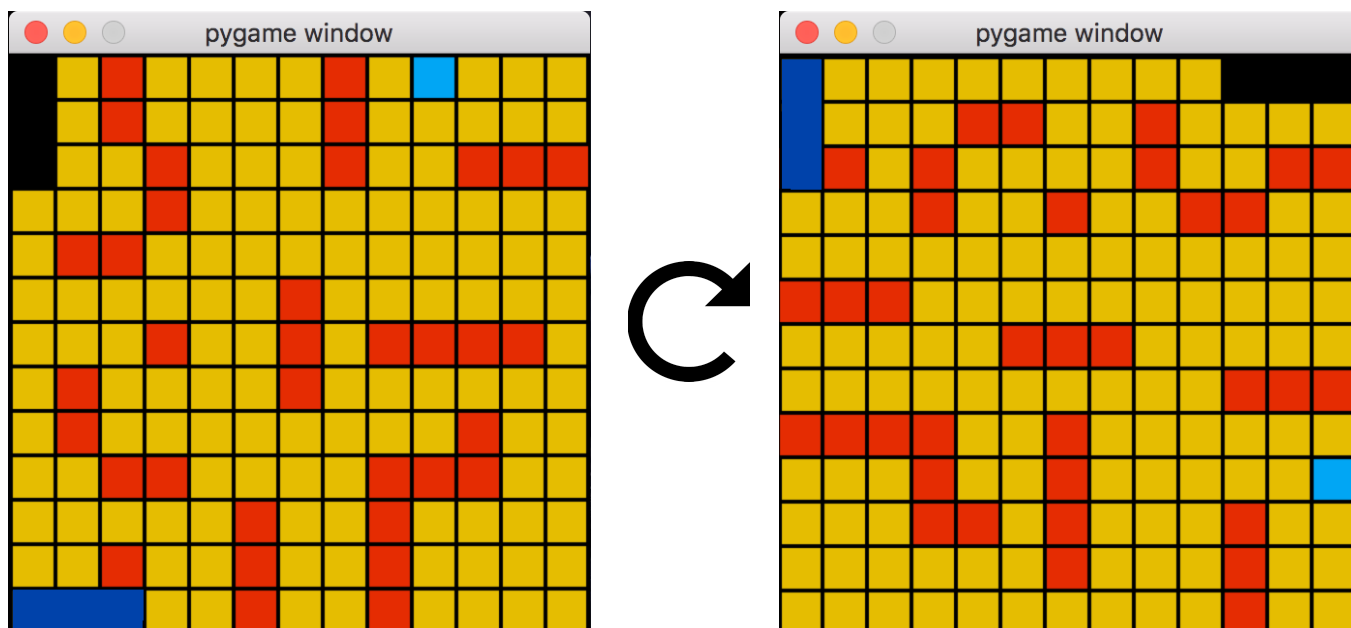
```
temp = {
    'username': username,
    'action': key
}
```

شکل ششم: قالب ارسال حرکت به سرور

شامل دومین سوکت به همراه نحوه پیاده سازی واکنش به بسته‌های دریافت شده و تولید پاسخ، استفاده از thread و non-blocking بودن ارتباط است.

۶. تبدیل نقشه

به شکل زیر توجه کنید شکل سمت چپ نقشه‌ای است که کاربر مشکی هنگام شروع بازی می‌بیند.^۵ طبق قوانینی که گفته شد هنگام شروع بازی کاربر آبی باید در نقطه‌ی بالا چپ صفحه قرار داشته باشد پس وقتی می‌خواهیم به این کاربر صفحه بازی را نمایش دهیم آن را ۹۰ درجه می‌چرخانیم. طبیعتاً حرکت رو به پایین مار مشکی در شکل سمت چپ معادل حرکت رو به چپ مار مشکی در شکل سمت راست است. انجام کارهای مربوط به این مورد در بخش اول قرار می‌گیرد. برای بقیه کاربران ۱۸۰ و ۲۷۰ درجه باید نقشه را بچرخانیم.



شکل هفتم: تبدیل نقشه‌ها

^۵ مار آبی برای توضیح استفاده شده است وگرنه همه مارها مشکی هستند.

۶. توضیحات تکمیلی

- پروژه‌ها دو نفره است و نمره‌ها لزوماً یکی نیست.
- بخش‌هایی که به صورت مشخص ذکر نشده‌اند را بین خود تقسیم کنید.
- برای اجرا کردن پروژه نیاز به نصب کتابخانه Pygame دارید.
- هنگام تحویل به صورت تصادفی مشخص می‌شود که چه کسی کدام بخش را تحویل می‌دهد برای همین اعضای گروه باید به هر دو بخش مسلط باشند.
- زمان فرستادن ضربان قلب را می‌توانید بر حسب نیاز خودتان تغییر دهید.
- حق ندارید بدون اجازه دستیاران آموزشی تغییری در کد داده شده اعمال کنید.
- پرسیدن مواردی که سلیقه‌ای محسوب می‌شود، از دستیاران ممنوع است و نمی‌توانید موقع تحویل بگویید چون دستیار آموزشی گفت، من این کار را انجام دادم. به جز مواردی که در فروم درس آمده باشد.
- تنها استفاده از زبان‌های برنامه نویسی پایتون و جاوا مجاز است. با توجه به این که ممکن است در بعضی از پروژه‌های آینده فقط مجاز به استفاده از زبان جاوا باشید توصیه می‌شود از این زبان استفاده کنید.
- استفاده از ابزار mininet برای درست کردن توپولوژی که روی آن پروژه اجرا شود نمره اضافه دارد.
- نقشه‌ی بازی یک فایل به نام map.txt است و باید از آن استفاده کنید. 1 مانع و 0 خالی بودن آن را نشان می‌دهد. بین اعداد یک فاصله وجود دارد آن را در نظر بگیرید.