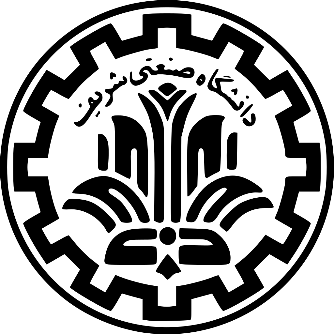
**بسمه تعالی**



**سیستم‌های نهفته**

**گزارش پروژه**

**بازی تانک دو نفره**

استاد

**دکتر انصاری**

نویسنده

**علیرضا هنرور (۹۸۱۰۲۵۵۱)**

**سیدعلی نجیبی (۹۸۱۰۶۱۲۳)**

**دانشگاه صنعتی شریف**

**نیم‌سال دوم ۱۴۰۲ – ۱۴۰۱**

# مقدمه

در این پروژه قصد داریم بازی تانک دو نفره را با استفاده از Raspberry Pi و سنسورهای حساس به صدا و فاصله، پیاده‌سازی کنیم. سنسور حساس به صدا، مربوط به تعیین زاویه‌ی پرتاب، و سنسور حساس به فاصله، مربوط به تعیین شدت پرتاب می‌باشد. بازی با استفاده از زبان برنامه‌نویسی Python نوشته شده، و لوازم استفاده‌شده در پروژه به صورت زیر می‌باشد:

* Raspberry Pi
* Breadboard
* MF Cables
* MM Cables
* Ultrasonic Sensor
* Sound Sensor
* Right and Left Keys (Bonus)

# هدف پروژه

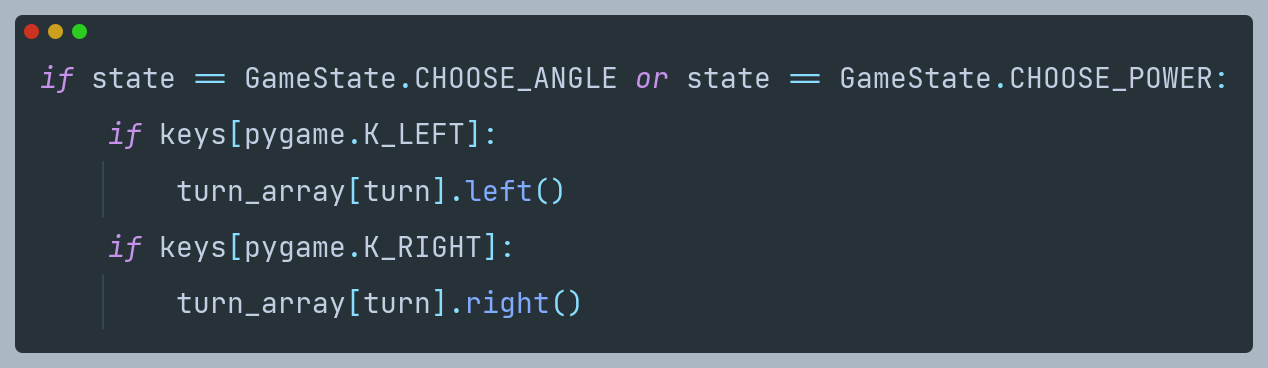
هدف در این پروژه، پیاده‌سازی بازی تانک دو نفره می‌باشد. این بازی نوبتی می‌باشد، و نوبت هر شخص که باشد، با دکمه چپ و راست اجازه حرکت دارد (اختیاری و بدون نیاز)، و ابتدا برای تعیین زاویه‌ی پرتاب بازیکن، یک خط‌چین در حال تغییر زاویه می‌باشد و با تشخیص صدا توسط سنسور، زاویه خود را ثابت می‌کند، و سپس به سراغ تعیین شدت پرتاب می‌شود. برای این کار، سنسور تشخیص فاصله، فاصله ابتدایی و فاصله انتهایی را می‌سنجد و تقسیم بر زمان کل می‌کند تا سرعت پرتاب بدست آید. به عنوان مثال، بازیکن ۳ ثانیه فرصت برای تعیین قدرت پرتاب دارد. در نتیجه تفاضل فاصله انتهایی و ابتدایی دست از سنسور تشخیص فاصله، تقسیم بر ۳ (ثانیه) می‌شود. اگر تیر به بازیکن مقابل برخورد کند، جان آن کاهش پیدا می‌کند تا اینکه یک بازیکن تماماً جانش تمام شود و ببازد.

# توضیح کد و روند انجام‌شده

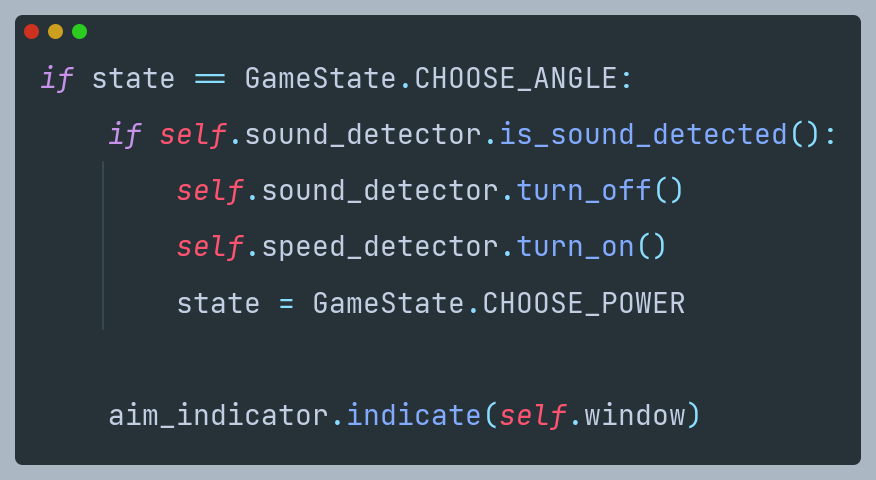


در کد بازی، ابتدا تصاویر و مکان بازیکن‌ها لود می‌شود. سپس، در یک حلقه‌ی while می‌رویم، و بازی شروع می‌شود. درون بازی، یک ساعت داریم که زمان و FPS بازی را در نظر می‌گیرد. سپس عامل سیاه بودن آسمان و سبز بودن زمین بازی اعمال می‌شود. توجه کنید که دلیل وجود این قطعه در حلقه while این است که هنگام نمایش نماد تعیین زاویه و گلوله و ...، باید نمایش‌های قبلی از بین بروند و تنها یک گلوله در هر ثانیه قابل مشاهده باشد. در نتیجه هر لحظه آسمان را سیاه می‌کنیم، تا تنها گلوله رسم‌شده در آن لحظه مشاهده شود. همچنین، در ابتدا، تشخیص صدا برای بازیکن اول فعال می‌شود.

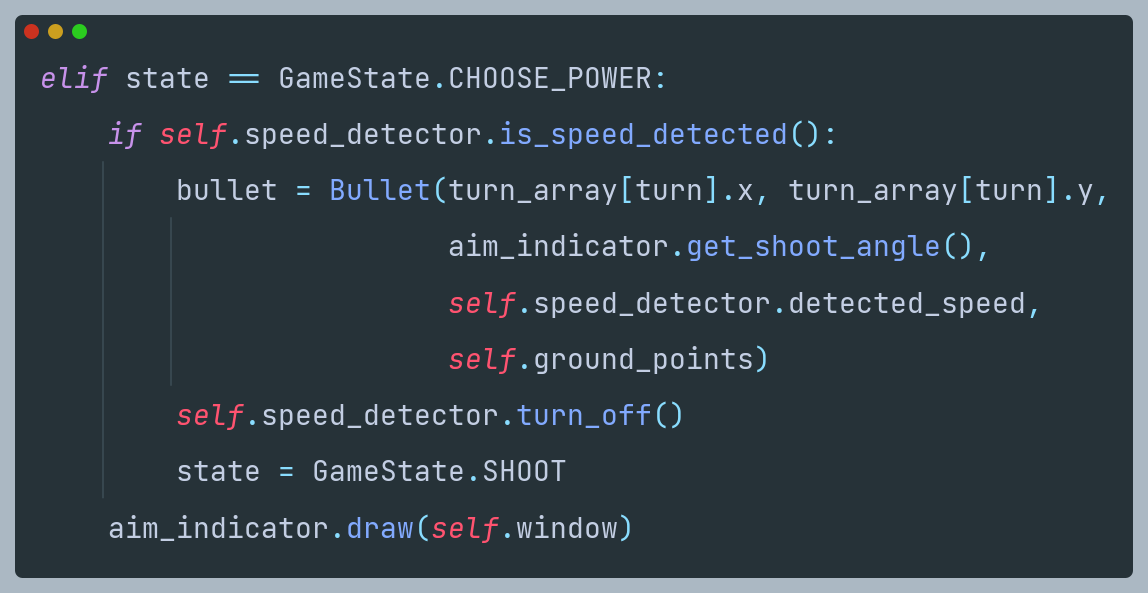
حال، به بخشی از کد می‌رسیم که چندین if و else if وجود دارد و وضعیت یا state را چک می‌کنیم و بر اساس آن یک کار انجام می‌دهیم. وضعیت‌ها به شکل زیر می‌باشند:



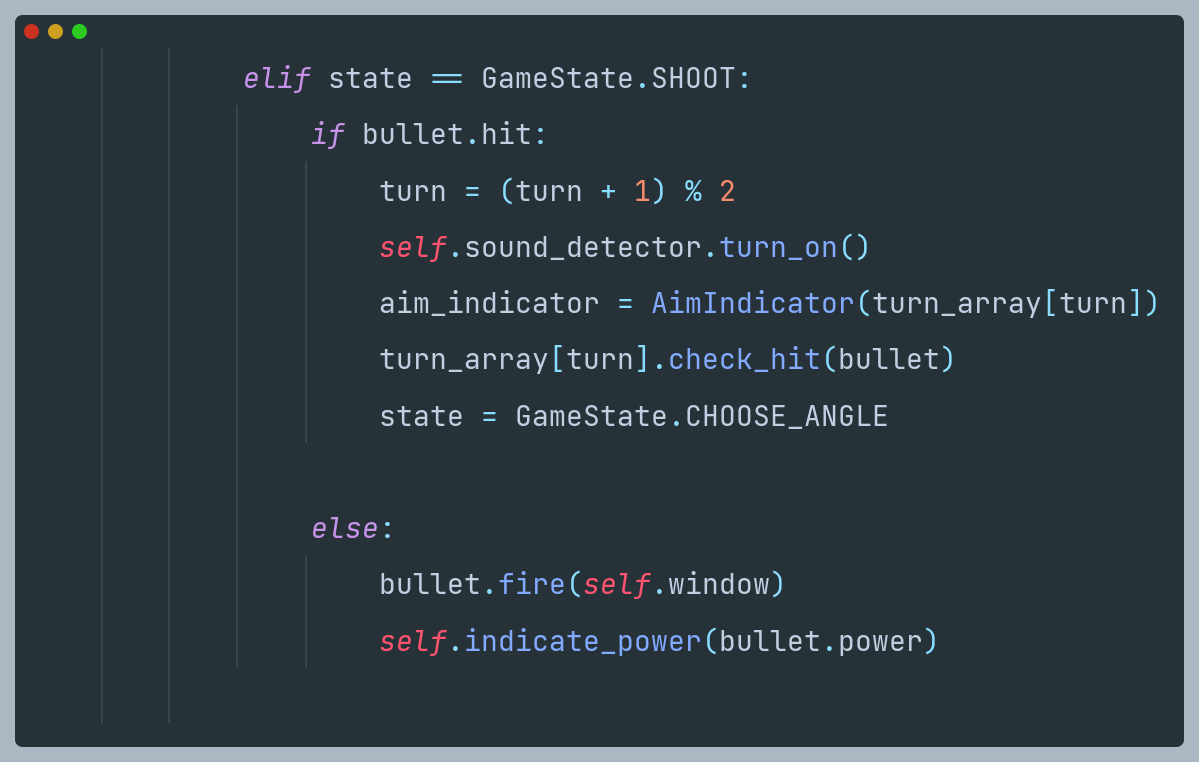
این وضعیت، صرفا جهت حرکت روی زمین می‌باشد و لزومی ندارد اجرا شود.



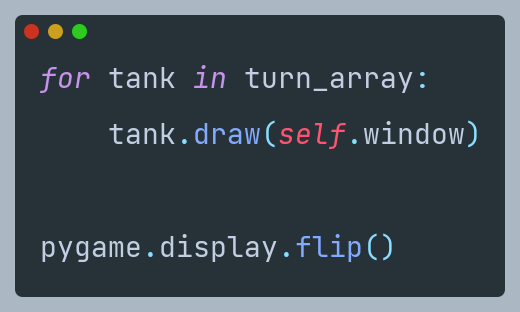
این وضعیت، تشخیص زاویه را انجام می‌دهد و از Sound Sensor کمک می‌گیرد.



این وضعیت، تشخیص قدرت را انجام می‌دهد و از Ultrasonic Sensor کمک می‌گیرد.



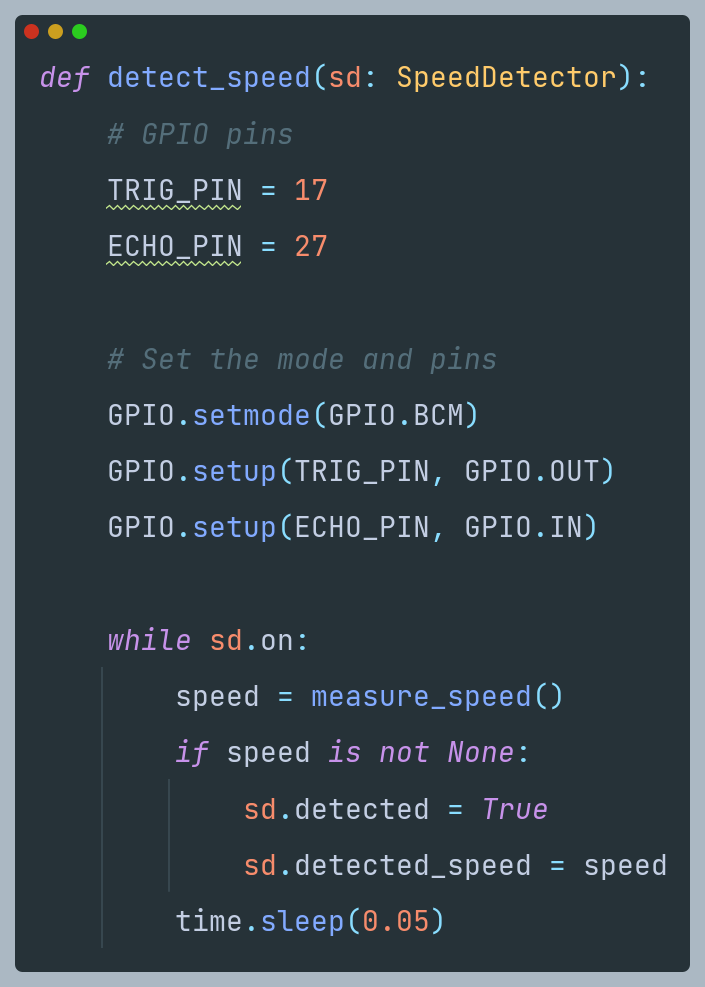
این وضعیت، جهت انجام انیمیشن پرتاب تیر می‌باشد و عملیات منطق حرکت تیر به سمت حریف را نیز دربر می‌گیرد.

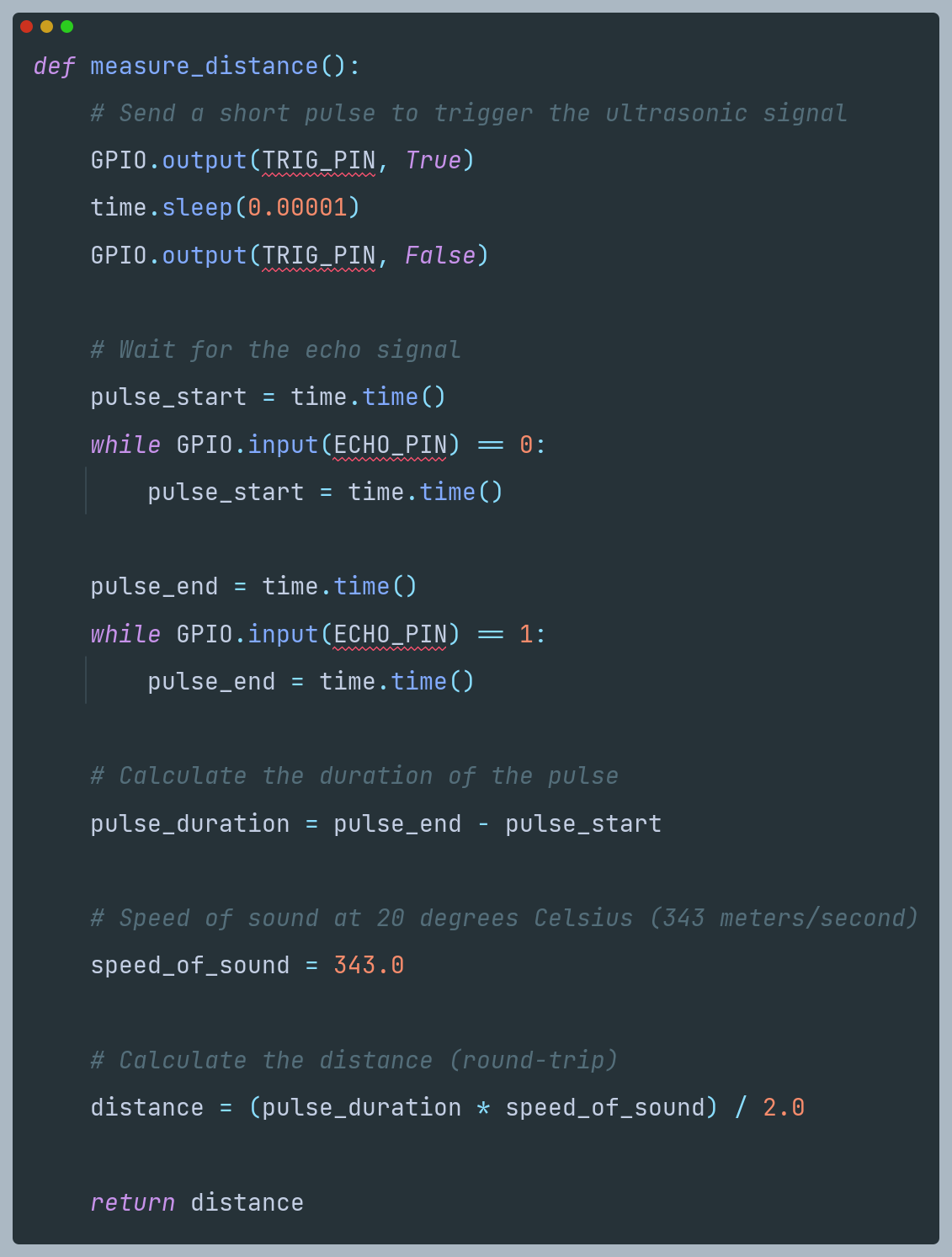
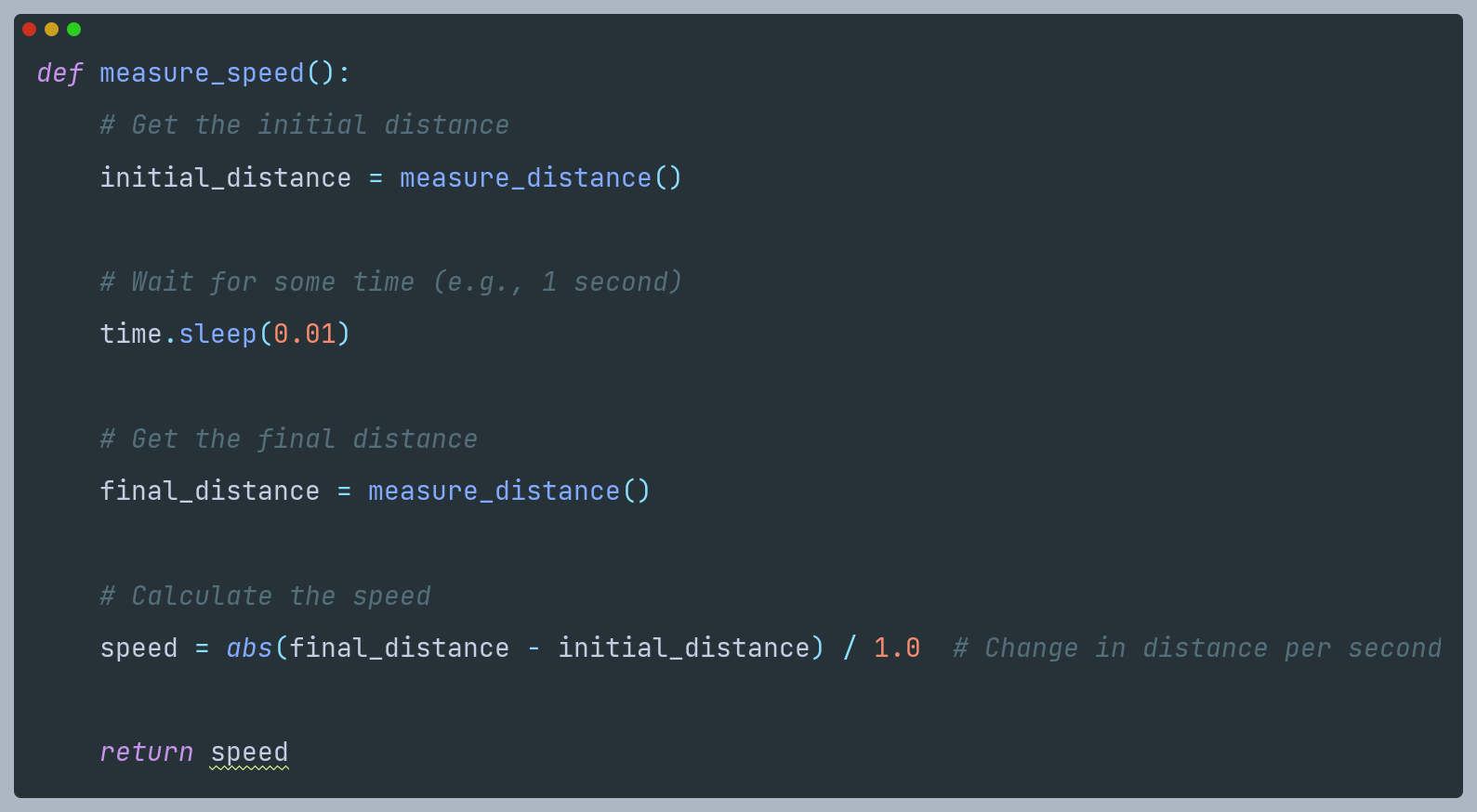


در نهایت، باید وضعبت فعلی تانک‌ها در این لحظه را رسم کنیم چرا که ممکن است حرکت کرده باشند و یا جان‌شان تغییر کرده باشد.

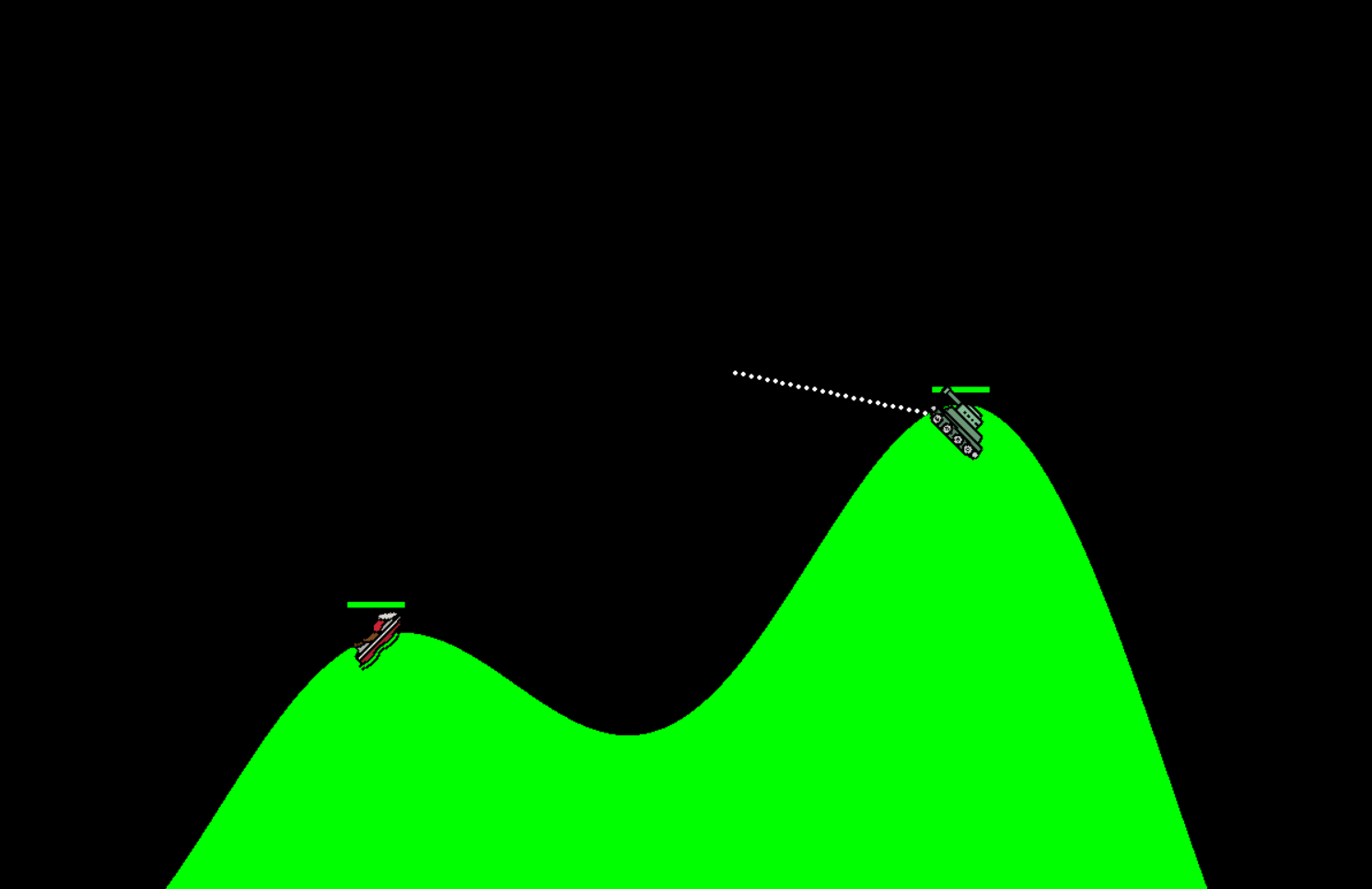
حال سراغ بررسی Sound Detection و Distance Detection می‌رویم. این دو قابلیت، به وسیله قابلیت Thread پیاده‌سازی می‌شوند:





# نتایج بدست آمده



در این بازی، همانطور که مشاهده می‌شود، یک زمین تولید می‌شود. بازیکن‌ها قابلیت حرکت دارند (امتیازی). جهت و شدت پرتاب مشخص می‌شود. و همچنین در نهایت بازیکنی که جانش تمام می‌شود، می‌بازد.