

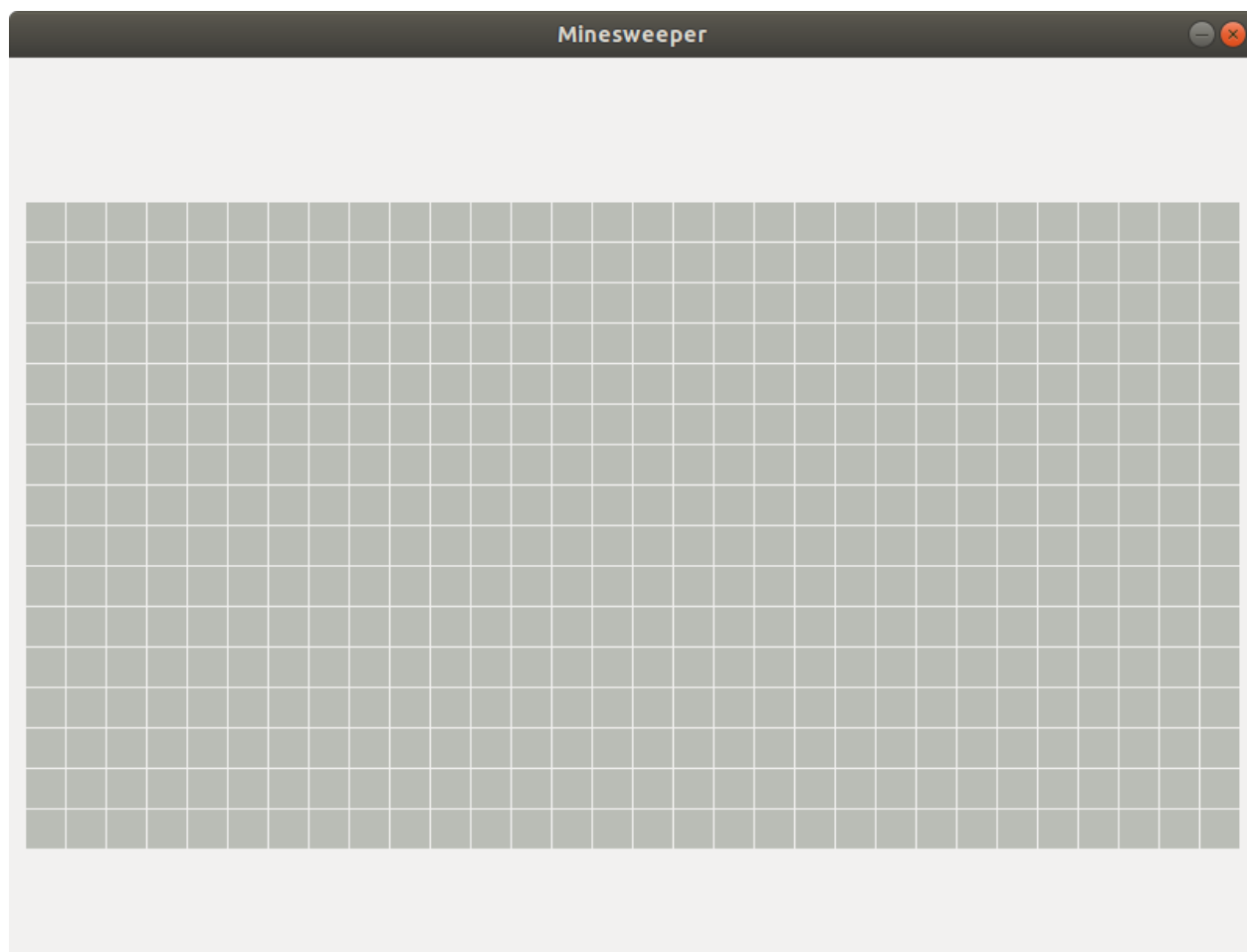
## مین روب

در این پروژه قصد داریم بازی معروف Minesweeper را پیاده سازی کنیم. زبان مَدَنظرِ این سؤال ++C است. اسکرین شات هایی که در صورت سؤال آمده اند برای فهمِ بیشترِ سؤال آمده اند، و شما کافی است بازی را در کنسول پیاده سازی کنید.

### توضیح راجع به بازی

صفحه ی بازی یک جدول  $n$  در  $m$  است. در بعضی از خانه های این جدول بمب هایی وجود دارد که کاربر نمی تواند آن ها را ببیند. هدفِ بازی این است که کاربر بتواند مکانِ بمب ها را کشف کند، بدونِ این که هیچ بمبی را منفجر کند.

در ابتدای بازی، صفحه ی بازی به این صورت است:

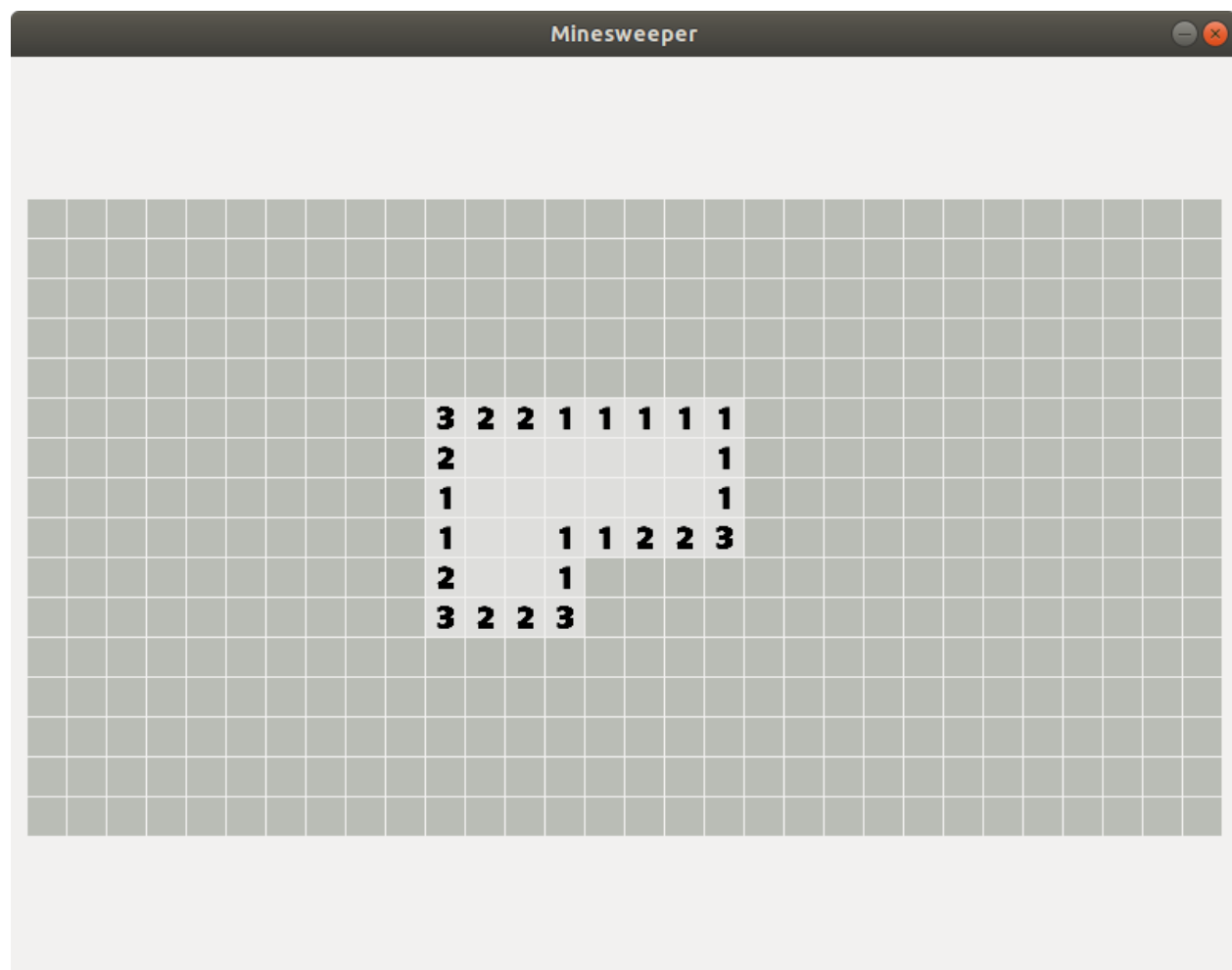


کاربر می‌تواند با کلیک بر روی هر کدام از خانه‌های جدول، بازی را شروع کند. تضمین می‌شود که در کلیک ابتدایی کاربر نمی‌بازد.

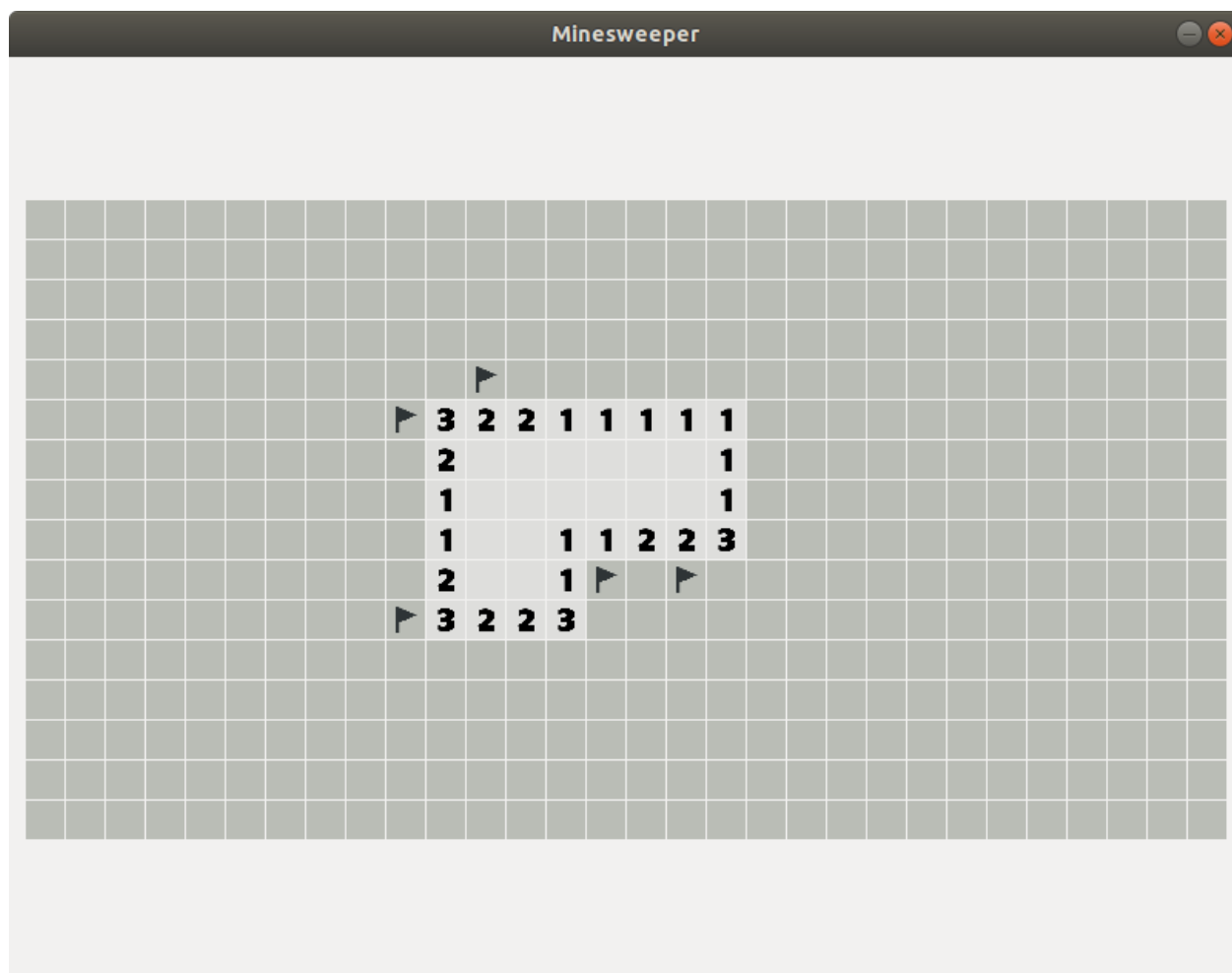
با کلیک بر روی هر خانه، آن خانه «آشکار» می‌شود. یعنی محتوای پنهان آن به کاربر نمایش داده می‌شود. این محتوا می‌تواند حالت‌های مختلفی داشته باشد:

- بمب: در صورتی که خانه‌ای که کاربر انتخاب کرده بمب داشته باشد، بازی با باخت پایان می‌یابد.
- عدد: خانه‌هایی که بمب ندارند، در داخل خود یک عدد دارند. این عدد نشان‌دهنده‌ی تعداد بمب‌هایی است که این خانه در مجاورت خود دارد.
- خالی: خانه‌هایی که نه خودشان بمب دارند نه در اطراف آن‌ها بمبی وجود دارد، خالی هستند.

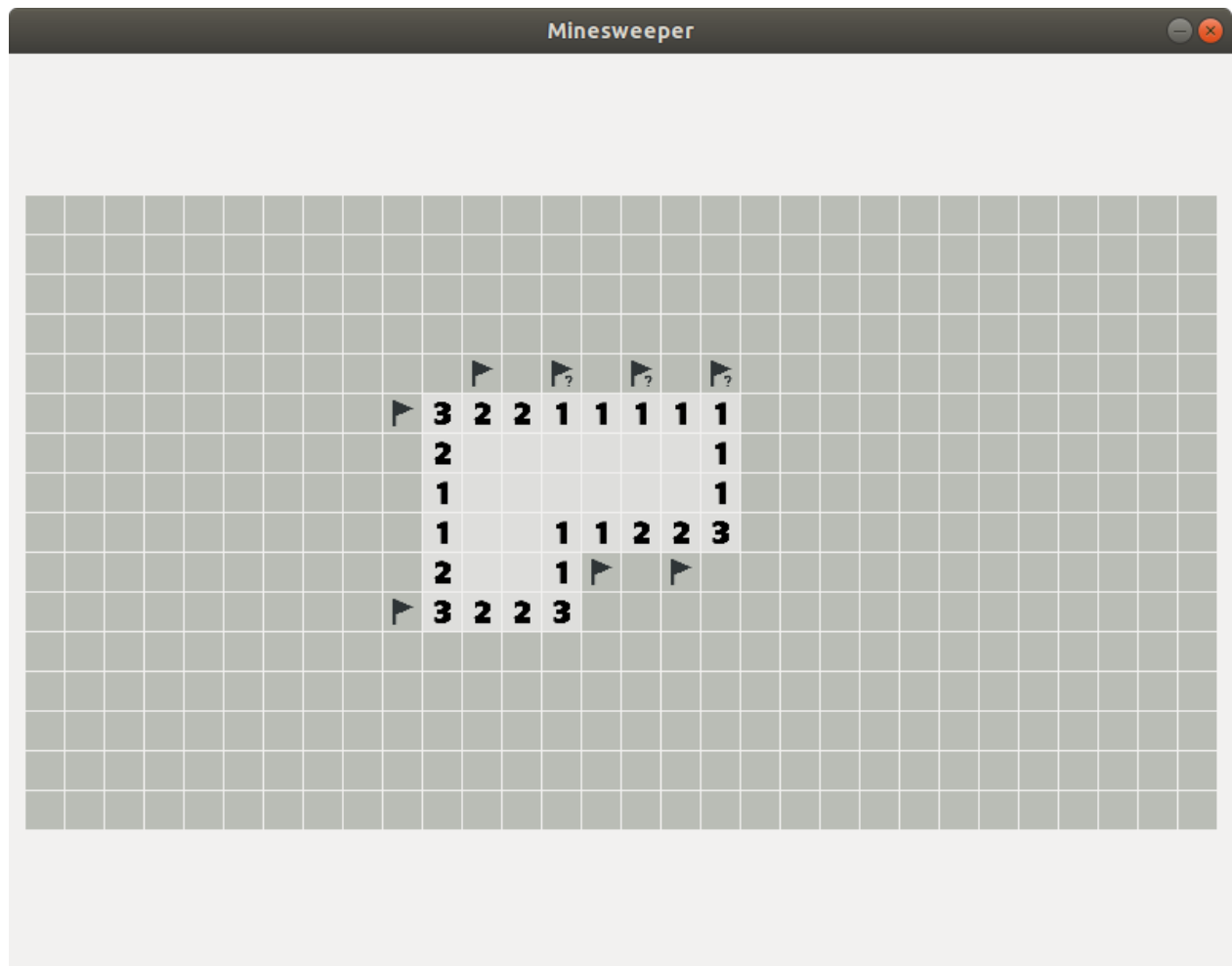
به عکس زیر توجه کنید:



با توجه به عددی که در عکس هست، می‌توان مکان بمب‌های مخفی را حدس زد. به عنوان مثال، در عکس زیر خانه‌هایی که پرچم دارند نشان‌دهنده‌ی خانه‌هایی هستند که کاربر با توجه به اطلاعاتی که دارد می‌داند در آن‌ها بمب هست. این پرچم‌ها با راست‌کلیک و به انتخاب کاربر در جدول اضافه می‌شوند. این پرچم‌ها صرفاً برای این است که کاربر اشتباهاً روی خانه‌های بمب‌دار کلیک نکند و تأثیر دیگری روی روند بازی ندارند.

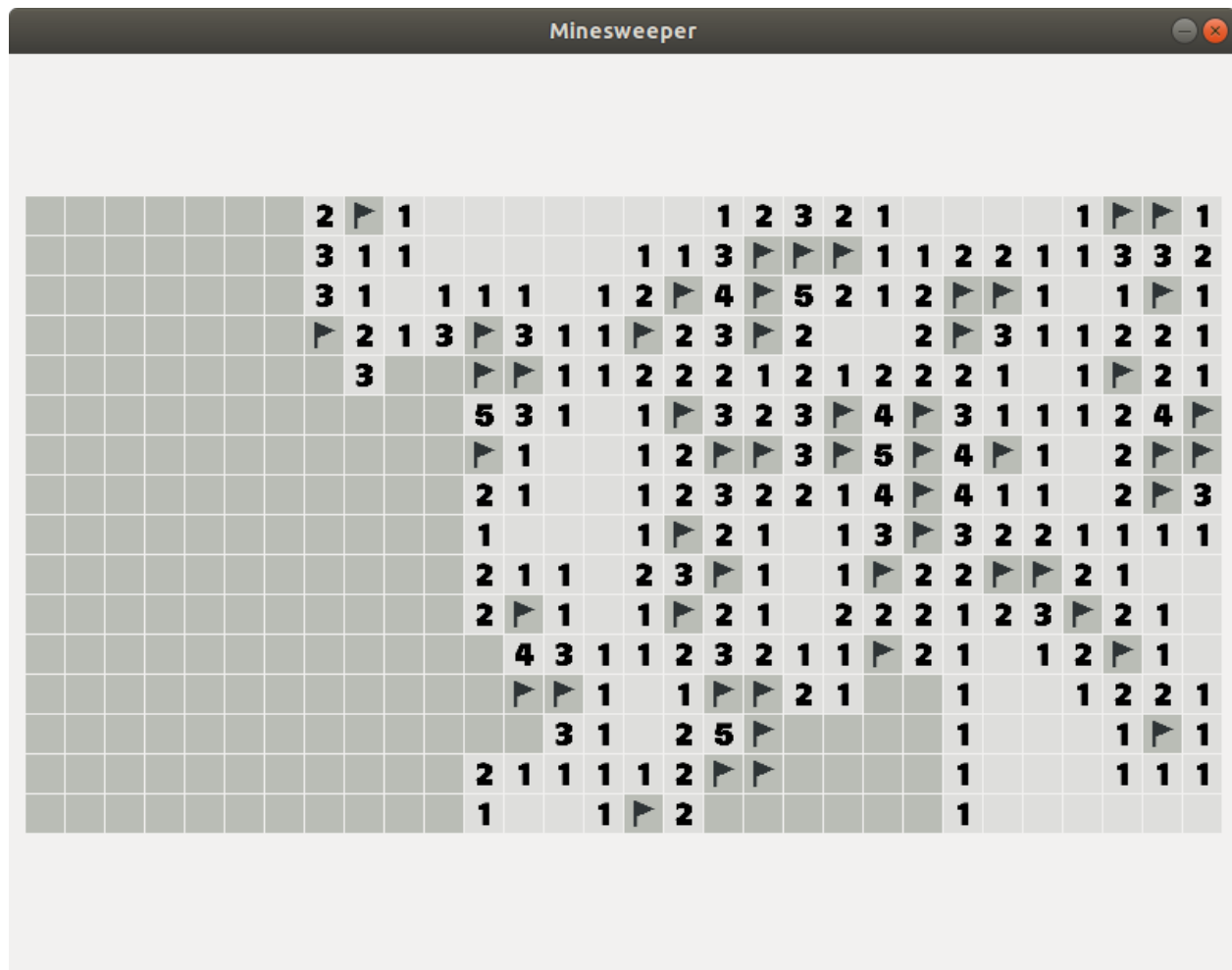


همچنین کاربر علاوه بر پرچم می‌تواند در جدول علامت سؤال نیز قرار دهد. این علامت مخصوص خانه‌هایی است که کاربر شک دارد که بمب هست یا نه. فرق این نوع خانه‌ها با خانه‌های پرچم در این است که اگر کاربر روی پرچم کلیک کند، خانه‌ی مورد نظر آشکار نمی‌شود. ولی علامت سؤال صرفاً یک علامت است و اگر کاربر روی خانه کلیک کند، آن خانه آشکار می‌شود. به در شکل زیر تعدادی علامت سؤال نیز اضافه شده است.

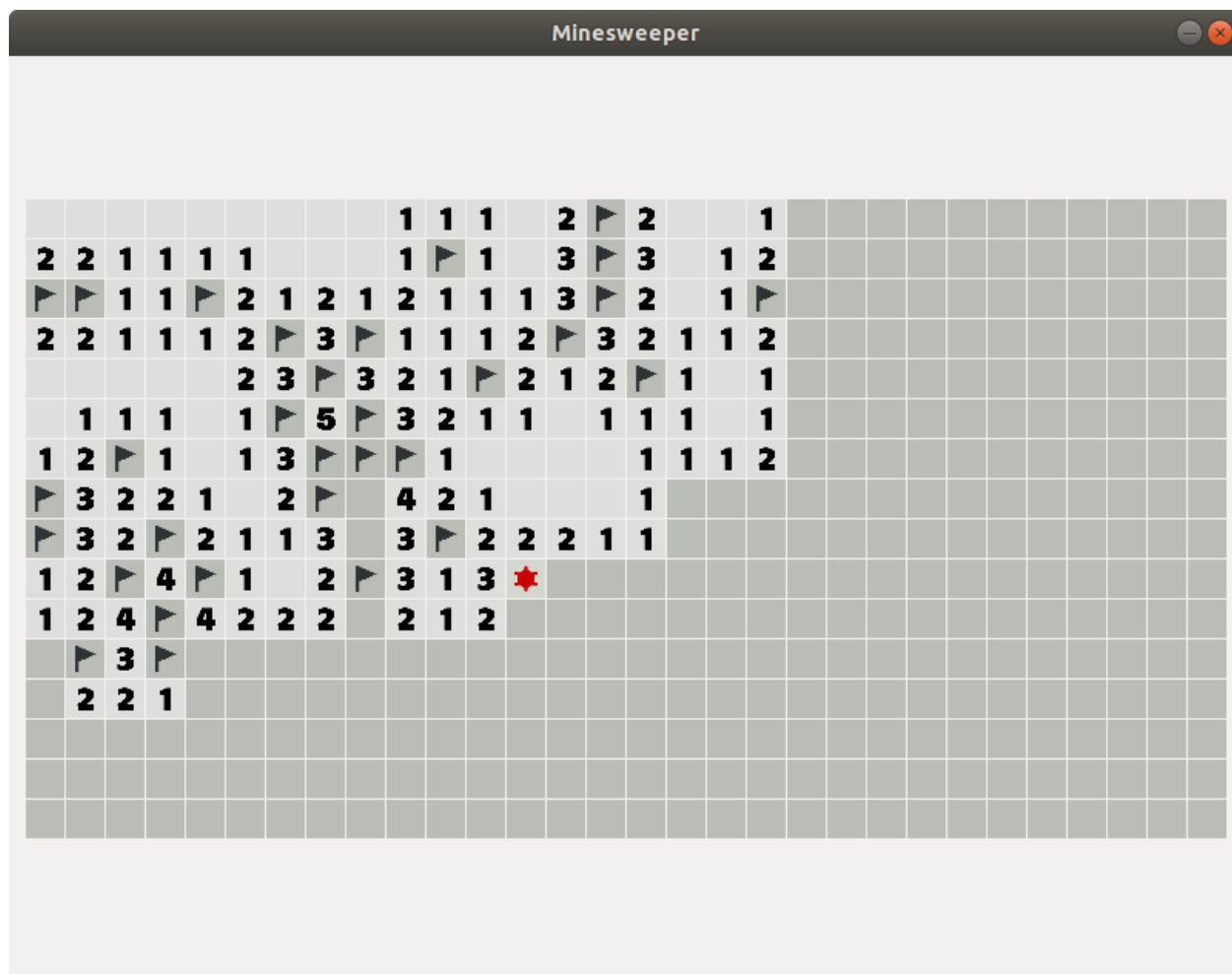


هدف بازی این است که کاربر تمامی خانه‌هایی که بمب ندارند را پیدا کند. در این صورت کاربر برنده است. اگر در این روند کاربر اشتباهاً روی یک بمب کلیک کند می‌بازد.

اسکرین‌شات‌های دیگری از بازی:



باخت:



پیشنهاد می‌کنم برای آشنایی بیشتر با این بازی، خودتان بازی را دانلود کرده و کمی با آن بازی کنید.

## موارد پیاده‌سازی در پروژه

شما در این تکلیف باید این بازی را در کنسول پیاده‌سازی کنید.

وقتی برنامه اجرا می‌شود، از کاربر طول، عرض و تعداد بمب‌ها را بگیرید. توجه داشته باشید که تعداد بمب‌ها از تعداد خانه‌ها نباید بیشتر باشد. همچنین باید حداقل برای یک خانه بدون بمب جا باشد.

بعد از گرفتن مشخصات صفحه از کاربر، صفحه‌ی بازی را به او نشان دهید. نحوه‌ی نمایش بر عهده‌ی خودتان است و در این زمینه اجباری وجود ندارد. هر بار که صفحه را به کاربر نشان می‌دهید، باید از او یک دستور دریافت کنید. این دستور یکی از دستورهای زیر می‌تواند باشد:

- پرینت: این دستور صفحه را (تا جایی که کاربر پیش رفته) چاپ می‌کند. این نمایش باید همان جزئیاتی را نشان دهد که در اسکرین‌شات‌های بالا مشاهده کردید.
  - کلیک روی یک خانه: این دستور از کاربر سطر و ستون را دریافت می‌کند، و خانه‌ی مورد نظر را برای کاربر آشکار می‌کند. اگر خانه‌ی مورد نظر پرچم باشد اتفاقی نمی‌افتد. اگر با این کار بمبی منفجر شود کاربر می‌بازد. در غیر این صورت عددِ مربوطه در آن خانه به کاربر نشان داده می‌شود.
  - گذاشتن پرچم روی یک خانه: با این دستور، بعد از گرفتن سطر و ستون از کاربر، در خانه‌ی مربوطه پرچم قرار داده می‌شود. توجه داشته باشید که پرچم فقط روی خانه‌هایی می‌تواند قرار بگیرد که آشکار نشده‌اند.
  - گذاشتن علامت سؤال روی یک خانه: با این دستور، بعد از گرفتن سطر و ستون از کاربر، در خانه‌ی مربوطه علامت سؤال قرار داده می‌شود. توجه داشته باشید که علامت سؤال فقط روی خانه‌هایی می‌تواند قرار بگیرد که آشکار نشده‌اند.
  - تعداد بمب‌های کل: تعداد بمب‌های جدول را چاپ می‌کند.
  - تعداد بمب‌های کشف‌نشده: با فرض این که کاربر روی خانه‌های درستی پرچم گذاشته باشد، این عدد نمایان‌گر تعدادِ بمب‌هایی است که کاربر هنوز جای آن‌ها را کشف نکرده است.
- همچنین در صورتِ برد یا باخت، پیامِ مربوطه را به کاربر نمایش دهید. بعد از این اتفاق دیگر لازم نیست از کاربر دستور بگیرید.

## امتیازی

- دستوری را به بازی اضافه کنید که به کاربر زمانِ گذشته را نشان دهد. این زمان باید نسبت به زمانِ شروع بازی اندازه گرفته شود.
- بهترین زمان‌هایی که کاربر در بازی‌های قبلی کسب کرده است را در یک فایل نگهداری کنید. هر بار که برنامه را باز می‌کنید این فایل را در برنامه‌ی خود لود کنید. کاربر می‌تواند با وارد کردن یک دستور، رتبه‌های برتر را (به همراه اسمِ کسی که رکورد را ثبت کرده) مشاهده کند. همچنین اگر کاربر بازی را در زمانی حل کند که بتواند در Top 10 قرار بگیرد، باید اسمش را از او دریافت کنید و در رتبه‌ها ذخیره کنید. برای این کار می‌توانید کتابخانه‌ی fstream را مطالعه کنید.

- اگر بازی اصلی را ببینید، بعضی اوقات که کاربر روی خانه‌ی با عددِ صفر کلیک می‌کند (قبل از این که بداند که خانه‌ی موردِ نظر صفر است!)، تمامِ خانه‌های مجاورِ خانه‌ی کلیک‌شده باز می‌شوند! این مسئله در اسکرین‌شات‌هایی که در صورتِ سؤال هست هم قابلِ مشاهده است. مثلاً در اسکرین‌شاتِ دوم، من فقط روی یک خانه‌ی خالی کلیک کرده‌ام ولی خانه‌های زیادی آشکار شده‌اند. راه‌نمایی: می‌توانید الگوریتمِ DFS را مطالعه کنید.

هدفِ اصلیِ این تمرین، به کارگیریِ دانشِ شیء‌گراییِ شما است. بنابراین هر چه بهتر از شیء‌گرایی استفاده کنید نمره‌ی بهتری می‌گیرید. همان‌طور که در ابتدای تمرین اشاره شد، زبانِ این سؤال ++C است.