



### مقدمه

هدف از این تمرین آشنایی شما با برنامه‌نویسی شیء‌گرای رویدادمحور<sup>۱</sup> و استفاده از آن در کنار کتابخانه‌های گرافیکی است. انتظار می‌رود از فنون برنامه‌نویسی که تاکنون در کلاس درس فراگرفته‌اید یا در هنگام تحویل حضوری تمرین‌ها به شما تذکر داده شده است به طور کامل در این تمرین استفاده کنید. توصیه می‌شود تمرین را در بخش‌های کوچک جلو ببرید و هر مرحله عملکرد آن را آزمایش کنید و بررسی درستی آن را برای مرحله آخر نگذارید.

برای پاسخ به این تمرین باید از کتابخانه‌ی گرافیکی SDL<sup>۲</sup> استفاده کنید. برای راحتی کار شما، در این **لینک** یک کتابخانه‌ی واسط به نام RSDL<sup>۳</sup> برای کار کردن با SDL در اختیار شما قرار داده شده است. قبل از شروع به انجام این تمرین توصیه می‌شود حتماً **ویدیوهای** مربوط به برنامه‌نویسی رویداد محور را ببینید و مستندات موجود در این **لینک** را مطالعه کنید.

همچنین این تمرین را باید به صورت چندفایلی پیاده‌سازی کنید. **ویدیوهایی** جهت آشنایی و آموزش ایجاد برنامه‌های چندفایلی با استفاده از makefile در صفحه‌ی درس قرار داده شده که می‌توانید از آن‌ها استفاده کنید.

<sup>۱</sup> Event Driven Programming

<sup>۲</sup> Simple DirectMedia Layer

<sup>۳</sup> Ramtin Simple DirectMedia Layer

# STAR WARS

"A long time ago in a galaxy far, far away...."

## روند بازی

در این بازی، شما یک سفینه را کنترل می‌کنید و سعی می‌کنید با هدف قرار دادن دشمنان که در بالای صفحه قرار دارند، همه آن‌ها را از بین ببرید تا سفینه‌ی گروگان گرفته شده را از چنگشان در بیاورید.

بازی در چند مرحله<sup>4</sup> انجام می‌شود. پس از نابود شدن همه‌ی دشمنان در یک مرحله، دشمنان مرحله‌ی بعد روی صفحه ظاهر می‌شوند. اگر توانستید تمام دشمنان در تمام مرحله‌ها را از بین ببرید شما برنده بازی هستید و اگر به دشمنان یا تیر آن‌ها برخورد کنید می‌بازید و بازی تمام می‌شود. اگر به صورت اشتباهی تیر شما به سفینه گروگان گرفته شده برخورد یا خودتان به آن برخورد کردید نیز می‌بازید و بازی تمام می‌شود.

برای آشنایی بیشتر با این سبک بازی و تجربه آن می‌توانید به این [لینک](#) مراجعه کنید.

**دقت کنید** بازی موجود در لینک به طور کامل با این تمرین یکسان نیست و حتماً جزئیات پروژه را در صورت تمرین مطالعه کنید.

## اجزای بازی

اجزای مختلف بازی و عملکرد مورد انتظار از آن‌ها در ادامه توضیح داده خواهد شد.

### سفینه

در طول بازی کنترل یک سفینه در اختیار شماست. این سفینه می‌تواند در چهار جهت بالا، پایین، چپ و راست حرکت کند. همچنین، می‌توانید به کمک این سفینه به دشمنان تیر پرتاب کنید و آن‌ها را از بین ببرید.

هر تیری که پرتاب می‌کنید با سرعت ثابتی در جهت مستقیم حرکت می‌کند و در صورتی که به جسمی برخورد نکند، از صفحه‌ی بازی خارج می‌شود. جهت سفینه‌ی شما (و همچنین جهت شلیک) همواره به سوی بالای صفحه است و حرکت

---

<sup>4</sup> round

کردن سفینه جهت آن را تغییر نمی‌دهد. همچنین، سفینه‌ی شما نمی‌تواند از صفحه‌ی بازی خارج شود و بایستی همواره درون صفحه باقی بماند.

برای کنترل این سفینه، می‌توانید از چهار کلید W, A, S, D به‌ترتیب برای حرکت در جهت راست، پایین، چپ و بالا استفاده کنید. همچنین با هر بار فشردن کلید فاصله، می‌توانید شلیک کنید.

## دشمنان

موجودیتی از بازی هستند که هدف ما در هر مرحله از بین بردن تمام آن‌ها است. تمام دشمنان با یک بار اصابت تیر از بین می‌روند. در کل دو نوع دشمن داریم:

- **دشمن عادی:** این نوع از دشمن پس از ظاهر شدن روی صفحه در جای خود ثابت باقی می‌ماند و تنها به سمت پایین صفحه تیر پرتاب می‌کند. این نوع دشمن (و سایر دشمن‌ها) بین هر دو بار پرتاب تیر، مدت زمان ثابتی صبر می‌کند و در طول این زمان تیری پرتاب نمی‌کند.

- **دشمن متحرک:** این نوع از دشمن مشابه دشمن ساده به سمت پایین صفحه تیر پرتاب می‌کند و تفاوت آن این است که پس از شروع بازی شروع به حرکت افقی می‌کند و زمانی که به انتهای صفحه برسد جهت حرکت خود را عوض می‌کند.

## سفینه‌ی گروگان گرفته شده

در روند بازی با سفینه‌هایی مواجه می‌شوید که توسط دشمنان گروگان گرفته شده‌اند. این سفینه‌ها دشمن شما نیستند، بنابراین اگر سفینه‌ی شما یا تیرهایی که شلیک می‌کنید به این سفینه‌ها برخورد کنند، بازی را می‌بازید. می‌توانید فرض کنید که دشمنان و تیرهای آن‌ها از روی این سفینه‌ها رد می‌شوند و برخوردی صورت نمی‌گیرد.

**توجه:** اگر با توجه به تکلیف‌های قبلی احساس می‌کنید انجام این تکلیف در زمان مقرر برایتان دشوار است می‌توانید بازی را بدون دشمن متحرک و سفینه گروگان گرفته شده پیاده‌سازی کنید و تا حداکثر ۸۰ درصد نمره پروژه را کسب کنید.

## قدرت‌ها<sup>5</sup> (امتیازی)

زمانی که دشمنان را مورد هدف قرار می‌دهیم، با احتمال ثابتی ممکن است در محل قبلی آن‌ها یک قدرت ایجاد شود. اگر با سفینه به این قدرتها برخورد کنیم، آن‌ها را به دست می‌آوریم. اگر پس از گذشت زمان مشخصی موفق به برخورد با این قدرتها نشویم، از صفحه‌ی بازی حذف می‌شوند؛ همچنین، پس از گرفتن آن‌ها، پس از گذشتن زمان مشخصی نیز اثر آن‌ها از بین می‌رود.

در کل دو نوع قدرت داریم:

- **افزایش سرعت تیر:** این قدرت باعث می‌شود که سرعت پرتاب تیر سفینه دو برابر شود.
- **سپر:** این قدرت باعث می‌شود که تیرهای پرتاب شده توسط دشمنان به سفینه اصابت نکند، به این معنی که تیر در صورت برخورد با سفینه بدون هیچ تاثیری از روی آن رد می‌شود.

## نقشه‌ی بازی

صفحه‌ی بازی را به صورت یک جدول مستطیلی در نظر بگیرید که اندازه‌ی آن در ورودی داده می‌شود. خانه‌ی  $(0, 0)$  در این جدول، در گوشه‌ی بالایی و چپ صفحه خواهد بود. موقعیت دشمنان با شماره‌ی سطر و ستون در این جدول مشخص می‌شوند. برای این‌که همیشه امکان نابود کردن دشمنان وجود داشته باشد، می‌توانید فرض کنید که هیچ دشمنی در پایین‌ترین سطر صفحه قرار نمی‌گیرد.

برای آشنایی بیشتر با نقشه‌ی بازی به بخش **ورودی** مراجعه کنید.

## درجه‌ی سختی بازی (امتیازی)

برای اینکه بازی روندی منطقی را طی کند و برد و باخت بازیکن به مهارت وی بستگی داشته باشد، توزیع تیرهای دشمن باید به صورتی باشد که بازیکن همواره راهی برای نجات خودش داشته باشد. برای این منظور همواره حداقل یک ستون از دشمنان موجود در نقشه نباید شلیک کنند. علاوه بر آن، فاصله بین تیرهای متوالی باید طوری تنظیم شود که حرکت از یک ستون بدون تیر به یک ستون بدون تیر دیگر مقدور باشد.

با استفاده از مفهومی که توضیح داده شد، می‌توان درجه‌ی سختی بازی را نیز تعیین کرد. سه درجه‌ی سختی آسان، متوسط و سخت را برای بازی تعریف می‌کنیم. در بازی آسان، هر بار که دشمنان شلیک می‌کنند، حداقل سه ستون از آن‌ها انتخاب

---

<sup>5</sup> power-up

می‌شوند و تیری از آن‌ها شلیک نمی‌شود؛ از بقیه‌ی ستون‌ها تعداد دلخواهی از دشمنان می‌توانند تیر شلیک کنند. در بازی متوسط به روش مشابه دو ستون و در بازی سخت نیز تنها یک ستون انتخاب می‌شود. توجه کنید که در صورتی که فاصله زمانی بین هر دو شلیک متوالی دشمنان کم باشد و دو ستون که در دو دور متوالی شلیک دشمنان انتخاب می‌شوند مجاور نباشند، بازیکن مقدور به حرکت به سمت ستون دور دوم نخواهد بود و در نتیجه می‌بازد. برای اینکه این مشکل پیش نیاید، فاصله‌ی زمانی بین هر بار تیر زدن را باید به مقدار صحیحی تنظیم کنید.

## پایان بازی

در صورتی که سفینه‌ی شما به دشمنان یا تیر آن‌ها برخورد کند، بازی را می‌بازید و روند بازی تمام می‌شود. همچنین اگر سفینه‌ی شما یا تیری که شما شلیک کرده‌اید به یکی از سفینه‌های گروگان گرفته شده برخورد کند نیز بازی را می‌بازید. در صورتی که موفق به از بین بردن آخرین سفینه در آخرین مرحله نقشه شوید، بازی تمام شده و برنده می‌شوید.

## ورودی

ورودی بازی که اطلاعات اندازه‌ی نقشه‌ی بازی، موقعیت دشمنان و درجه‌ی سختی هر مرحله را دارد، به صورت یک پرونده‌ی متنی کنار پرونده‌ی اجرایی بازی قرار می‌گیرد. تعیین ساختار این پرونده‌ی متنی به عهده شما است. این فایل باید شامل تمام اطلاعات مورد نیاز برای ساخت مراحل بازی باشد. در ادامه یک ساختار پیشنهادی برای این پرونده توضیح داده خواهد شد، دقت کنید که نیازی نیست الزاماً از این ساختار استفاده کنید:

در خط اول، دو عدد  $m, n$  قرار دارند که به ترتیب تعداد سطرها و ستون‌های جدول بازی را نشان می‌دهند.

در خط بعدی، عدد  $l$  آمده است که تعداد مراحل بازی را نشان می‌دهد.

در ادامه،  $l$  جدول  $m \times n$  آمده‌اند که هر خانه‌ی این جدول‌ها، یکی از حروف زیر خواهد بود:

- حرف E که نشان‌دهنده‌ی دشمن ساده است.
- حرف M که نشان‌دهنده‌ی دشمن متحرک است.
- حرف S که نشان‌دهنده‌ی سفینه‌ی گروگان گرفته شده است.
- حرف '!' (نقطه) که نشان‌دهنده‌ی خانه‌ی خالی در جدول است.

در انتهای هر جدول درجه سختی بازی می‌آید. Easy نشان دهنده مرحله آسان، Medium نشان دهنده مرحله متوسط و Hard نشان دهنده مرحله سخت خواهد بود.

بین هر دو جدولی که در ورودی داده می‌شود، در یک خط سه حرف '-' می‌آید.

در مرحله‌ی اول، سفینه‌ی شما در یک خانه‌ی تصادفی از پایین‌ترین سطر نقشه ظاهر می‌شود. در طول روند بازی کنترل سفینه بر عهده‌ی شماست و با شروع هر مرحله‌ی جدید، سفینه‌ی شما به یک خانه‌ی تصادفی در پایین‌ترین سطر صفحه برمی‌گردد.

یک نمونه از ورودی بازی در ادامه آمده است:

```
3 4
2
.E..
E..M
....
Easy
---
ES.E
.M..
....
Medium
```

## نکات تکمیلی

- در این تمرین تعدادی پارامتر وجود دارد (مانند اندازه اشیا در صفحه، سرعت پرتاب تیر و فاصله بین پرتاب تیرها و...). از شما انتظار می رود که طبق صلاح دید خودتان این اعداد را تنظیم کنید به صورتی که بازی قابل بازی کردن باشد و تفاوت بخش های خواسته شده مشخص باشد (خیلی روی اعداد دقیق حساس نباشید).
- در صورت تمرین کلیت انتظارات از بازی بیان شده و در تصمیم گیری شیوه پیاده سازی آن ها (مانند ساختار نقشه ورودی، فیزیک بازی، اندازه اشیا و ...) دست شما باز است. هدف نهایی این است که برنامه نهایی قابل بازی کردن باشد.
- توصیه می شود که پروژه را از ساده به پیچیده جلو ببرید و ابتدا کلیت بازی را پیاده سازی کنید. به عنوان مثال می توانید ابتدا سفینه را بنویسید. سپس در طی چند مرحله، برنامه را پیچیده تر کنید. همچنین روند ارزیابی شما به صورتی قسمت بندی شده است که حتی در صورتی که پروژه را کامل نکرده باشید، نمره مربوط به قسمت های کامل شده به شما تعلق بگیرد. به عنوان مثال، ابتدا صرفاً حرکت سفینه در صفحه را پیاده سازی کنید. بعد از آن دشمن ساده را اضافه کنید و به همین ترتیب اجزای بازی را کامل کنید.
- از آنجایی که از این پس پروژه ها باید به صورت چند فایل پیاده سازی شود، توصیه می شود که پروژه را از ابتدا در چند فایل پیاده سازی کنید و از پیاده سازی کل پروژه در یک فایل و سپس تبدیل آن به چند فایل خودداری کنید.

## نحوه تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی zip با نام **A5-<SID>.zip** در صفحه‌ی ایلرن درس بارگذاری کنید که **SID** شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۹۹۹ است، نام پرونده‌ی شما باید **A5-810199999.zip** باشد.

- دقت کنید** که پرونده‌ی zip آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از zip کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید. برای مثال، نمونه فایل مورد قبول در ادامه آمده است:

A5-810199999.zip

```
|— main.cpp
|— map.txt
|— makefile
|— ...
```

- برای ایجاد رابط کاربری گرافیکی<sup>۶</sup> و تمامی افکت‌های برنامه خود باید از کتابخانه‌های SDL2 و RSDL استفاده کنید.
- فایل بارگذاری شده توسط شما باید پوشه‌ی کامل پروژه باشد که شامل کد کامل برنامه شما به همراه کتابخانه‌ی RSDL، تصاویر و سایر موارد است.
- در این تمرین بازی شما توسط دستیاران آموزشی آزموده می‌شود و تست خودکار ندارد.
- نمره هر بخش در صورت صحت عملکرد آن در بازی شما اختصاص می‌یابد و داشتن کد یک بخش که در بازی قابل آزمون نیست نمره‌ای برای شما ندارد.
- برنامه شما باید حتماً طراحی شیء‌گرا داشته باشد.
- دقت کنید** که پروژه‌ی شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد C++11 استفاده می‌کنید.
- دقت کنید** که نام پرونده‌ی اجرایی شما باید starwars.out باشد.

---

<sup>۶</sup> GUI



- نحوه اجرای برنامه به صورت `map.txt starwars.out` است که `map.txt` نام فایل نقشه شما است.
- طراحی درست، رعایت سبک برنامه‌نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.