



پروژه پایانی

Final Fight

مبانی کامپیوتر و برنامه‌نویسی | زمستان ۱۴۰۲

استاد: دکتر حسن بشیری
طرح و تهیه: تیم طراحی پروژه
مهلت تحویل: ۵ اسفند ساعت ۲۳:۵۹

فهرست

- ساختار پروژه ۴
- قوانین ۵
- سفینه خودی ۵
- سفینه دشمن ۵
- شلیک گلوله ۶
- امتیاز ۶
- هیل سفینه خودی ۷
- مپ ۷
- پایان بازی ۷
- نکات ۸
- موارد مربوط به پیاده سازی ۹
- بخش امتیازی ۱۱
- گیت ۱۲
- مصادیق تقلب ۱۲

جنگ ستارگان به پایان میرسه؟

همونطور که می‌دونین تو مینی‌پروژه یه ساختار کلی داشتیم که بر اساس اون پروژه‌هامون رو زدیم. حالا تو بخش پروژه پایانی قراره از دونسته‌های مینی‌پروژه‌مون استفاده کنیم و بازی‌مون رو گسترش بدیم. چطوری قراره اینکارو بکنیم؟ در ادامه بیشتر آشنا می‌شیم...

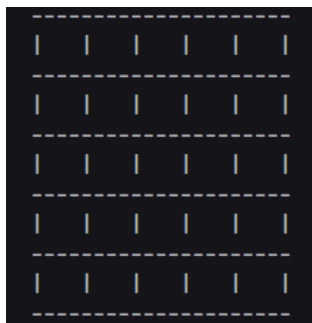
توی مینی‌پروژه سفینه‌های دشمن ثابت بودن و صرفاً سفینه خودی میتونست به 4 جهت حرکت کنه و به 2 جهت هم شلیک کنه. با تموم شدن heal سفینه یا از بین رفتن تمامی دشمنان بازی به پایان می‌رسید. اما تو پروژه پایانی تفاوت‌های زیادی وجود داره.

ساختار کلی پروژه پایانی چیه؟

ممکنه بازی‌هایی که یک سفینه پایین مپ قرار داره و فقط به چپ و راست حرکت می‌کنه و تو هر حرکت شلیک می‌کنه رو بازی کرده باشین. سفینه‌های دشمن از بالا وارد می‌شن و به سمت پایین حرکت می‌کنن و گلوله‌ها بهشون برخورد می‌کنن و از بین می‌رن. اگه بازیکن از بازی خارج شد و مجدد وارد شد باید ادامه بازی قبلی نمایش داده بشه.

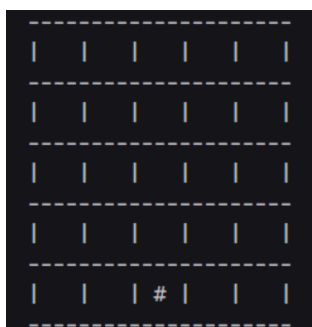
ساختار پروژه

- خب تو این بازی یه مپ داریم که مدام قراره آپدیت بشه. این مپ یه نمای فرضیه و سائز مپ اصلی تون نباید 5*5 باشه.

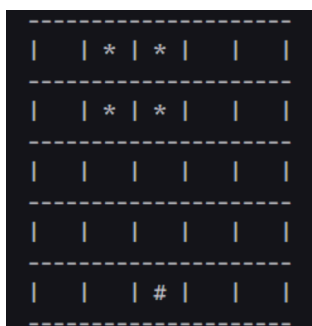


- سفینه خودی در سطر انتهایی مپ قرار داره:

مثل این شکل که سفینه در سطر آخر و وسط مپ قرار داره.



- سفینه‌های دشمن هم به صورت رندوم با سائزهای متفاوت (سائزها و مشخصات دقیق همه‌ی دشمنان فضایی، از قبل مشخص شده‌ان و شما فقط باید با تابع رندوم مشخص کنید که دشمن جدید، از کدوم نوع تعریف شده ست!) از بالا وارد مپ می‌شن: (در این شکل یه سفینه دشمن 2*2 وارد مپ شده. سفینه‌ها سائزهای مختلفی دارن که هر سفینه heal خاص خودشو داره. اگه heal یه سفینه برابر 3 باشه، یعنی باید 3 تا تیر بخوره تا ازبین بره).



- سفینه خودی در هر حرکت به سمت بالا شلیک می‌کنه. گلوله در هر مرحله باید نمایان باشه که در صورت برخورد با دشمن، دشمن از بین می‌ره.



اگه بخوایم به صورت یه جمع‌بندی نهایی از روند و پردازش حرکت‌های داخل بازی چیزی بهتون بگیم اینه که کل دنیای بازی به رفتار و عمل بازیکن بستگی داره. به این صورت که به ازای هر حرکت (به سمت چپ یا راست) یک گلوله‌ی جدید شلیک می‌شه و همه‌ی گلوله‌های قبلی که توسط بازیکن شلیک شده بودن، یه خونه از مپ بالا می‌رن تا جایی که به یه دشمن برخورد کنن و دشمن دمیج بده یا این که اگه دشمنی سر راه شون نیست، از مپ خارج بشن.

قوانین

■ سفینه خودی

این سفینه در سطر انتهایی مپ قرار می‌گیره (وسط مپ). سفینه میتونه به دو جهت چپ و راست حرکت کنه که حین هر حرکت، سفینه باید به صورت اتوماتیک شلیک کنه. میتونید برای پیاده سازی این نکته که اگه نخواستیم سفینه رو حرکت بدیم و سرجای خودش باشه جهت پایین رو به صورت پیشفرض براش تعریف کنید که اگه کلید پایین فشرده شد سرجای خودش قرار بگیره و شلیک به صورت اتومات انجام بگیره.

■ سفینه دشمن

چند مدل سفینه دشمن وجود داره که هیل هرکدوم متفاوته. هرکدوم یه اسم متفاوت دارن. سفینه های دشمن تو هر مرحله یک واحد به سمت پایین میان. ممکنه بگین چطور پیاده سازی کنیم؟ خب زمانی که سفینه خودی یه حرکت انجام داد، شلیک به صورت اتومات انجام میشه و بعد شلیک یه واحد سفینه دشمن رو به سمت پایین حرکت بدین. بهتره که برای هر سفینه یه رنگ جداگانه در نظر بگیرید.

لینک های کمکی برای رنگی کردن نوشته ها : [Link1](#) و [Link2](#)

- 1- سفینه **Dart** : این سفینه هیلش از همه کمتره و برابر 1 . سایز سفینه هم 1*1
- 2- سفینه **Striker** : هیل این سفینه برابر 2 و سایزش 2*2
- 3- سفینه **Wraith** : هیل این سفینه برابر 4 و سایزش 3*3
- 4- سفینه **Banshee** : هیل این سفینه برابر 6 و سایزش 4*4

Size	Heal	نوع سفینه
1*1	1	Dart
2*2	2	Striker
3*3	4	Wraith
4*4	6	Banshee

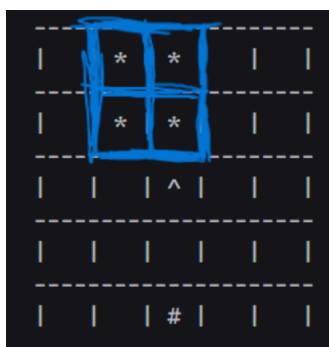
تو این شکل یه نمای کلی از سفینه های دشمن رو میبینی که heal و size های مختلفی دارن. سایز (size): نشان دهنده اینه که سفینه دشمن چه مقداری از مپ رو اشغال میکنه (اگه striker رو درنظر بگیریم توی مپ به اندازه یه ماتریس 2*2 فضا اشغال میکنه). هیل (heal): تعداد شلیک های مورد نیاز برای از بین رفتن سفینه.

■ شلیک گلوله

هرگلوله میتونه تنها یک heal از heal های دشمن کم کنه. میتونی کرکتر شلیک رو هم با کرکتر ' ^ ' نشانه گذاری کنی. گلوله ها صرفاً به سمت بالا حرکت میکنن و اگه به سفینه دشمن خوردن یا از مپ خارج شدن از بین میرن. همچنین درنظر داشته باشی که گلوله ها باید قابل مشاهده باشن.

■ امتیاز

هر سفینه امتیاز خاص خودشو داره. هر بلاک (Block) از سفینه ها 2 امتیاز داره. حالا چطوری امتیاز رو محاسبه کنیم؟ اصلاً Block چیه؟
خب یه ماتریس 2*2 در نظر بگیرید، که 4 تا block داره. حالا اگه بخوام مثال بهتر بزنم شکل زیر رو ببین:



یه دشمن از نوع **striker** داریم که سایزش 2*2 و هیلش برابر 2. از 4 تا خونه مجاور هم تشکیل شده که هرخونه اون رو block در نظر میگیریم. پس striker دارای 4 تا block.

محاسبه اش چطوره؟ خب بالاتر دیدیم که هر بلاک 2 امتیاز داشت پس:

point = count(Block) * 2 ; // Base Formula

point = 4 * 2 = 8 // Example With Striker (Enemy Type)

■ هیل سفینه خودی:

سفینه خودی یه هیل (heal) مشخص داره که اگه 0 شد بازی تموم میشه یا به اصطلاح game over میشه .

هیل سفینه خودی برابر 3. حالا چطوری هیل سفینه کم میشه؟ اگه سفینه های دشمن بعد از پایین اومدن های متعدد به سفینه ما برس، از هیل سفینه ما یکی کم میشه.

■ مپ:

بازی باید پویا باشه و بشه مپ رو با سائز های مختلف ایجاد کرد . درعین حال چون سفینه دشمن (Banshee) به 8 تا گلوله برای نابودی نیاز داره پس باید حداقل سائز مپ 15 باشه که بتونیم از بین ببریمش (یعنی یه ماتریس 15*15) . اگه عدد ورودی به عنوان سائز مپ کوچیک تر از 15 بود باید پیغام مناسبی به عنوان خطا به کاربر نمایش داده بشه و دوباره سائز مپ رو از کاربر بگیره یا این که کاربر بتونه گزینه‌ی exit رو از منوی نمایش داده شده انتخاب کنه. (طراحی و ریزه‌کاری‌های منو با شماست! اما باید گزینه‌های مختلف موردنیاز برای یک بازی معمولی رو ساپورت کنه. پیشنهاد می‌شه حداقل یه بازی ساده رو چک کنید و ببینید اولیه‌های یه منوی بازی چیه...!)

■ پایان بازی:

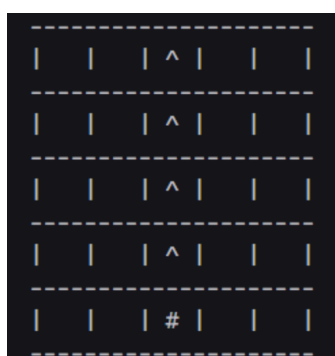
بازی در 2 صورت به پایان میرسه:

1- برنده (Win) : کاربر یه حد نصاب برای امتیاز درنظر میگیره (یعنی حد نصاب امتیاز برای برد بازی باید در ابتدای اجرای برنامه و داخل منو از کاربر به صورت ورودی دریافت بشه) و اگه مجموع امتیاز ها به حدنصاب رسید بازی تموم بشه. هنگامی که کاربر به حد نصاب امتیاز برسه، بازی باید پیغامی تحت عنوان win به کاربر نمایش بده و ازش بپرسه که می‌خواد همچنان بازی رو ادامه بده یا خیر. فرض کنید به حدنصاب بازی رسیدیم و تمایل داریم که بدون محدودیت (حدنصاب) بازی رو ادامه بدیم. پس باید کاربر قابلیت انجام ادامه بازی رو داشته باشه. صفحه (ترمینال) پاک بشه و سپس در صورتی که کاربر گزینه‌ی ادامه‌ی بازی رو از منو انتخاب کرد، بتونه به بازی که از قبل ذخیره شده ادامه بده. دقت کنید که باید دکمه/دستوری رو برای راحتی کاربر تهیه کنید که هرجایی خواست، بتونه بازی رو ناتمام بذاره و از بازی بیاد بیرون.

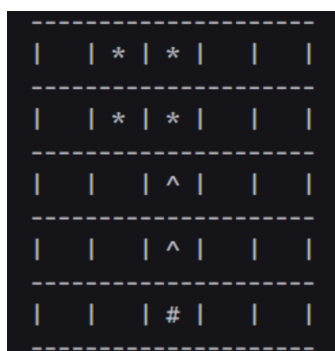
2- بازنده (game over): اگه تعداد هیل (heal) سفینه خودی به 0 برسه نشان دهنده اتمام بازی و بازی game over میشه.

۱. در بالای مپ باید قابلیت های بازی نمایش داده بشه؛ از قبیل : هیل سفینه خودی، امتیاز، سائز مپ، گزینه های پایه ای منو (Menu) و شرایطی که ارتباط کاربر رو با محیط بازی راحت تر کنن. (این بخش و طراحی تمام به عهده ی شماست! در صورتی که به خوبی پیاده سازی بشه نمره ی بیشتری نداره؛ اما اگه گزینه های یه منوی ساده رو نداشته باشه، باعث کسر نمره می شه. پیشنهاد می کنیم که یه بازی حتی از بازی های موبایلی چک کنید و ببینید منوهایی که برای متوقف کردن موقت بازی دارن، چیه. دقت کنید که بازی باید توی هر مرحله (پس از هر حرکت بازیکن) سیو بشه و اگه کاربر خواست منوی بازی رو باز کنه و موقتا بازی رو متوقف کنه، بدون این که اطلاعاتش از دست بره، این کار رو بتونه انجام بده. پس شما توی منویی که به کاربر به صورت جداگانه نمایش می دید باید هم گزینه ی exit و هم گزینه ی resume داشته باشید).

۲. موقع insert کردن یا قرار دادن سفینه دشمن تو مپ (که باید ردیف اول مپ باشه)، اگه تعدادی گلوله تو اون موقعیت وجود داشت، باید از هیل دشمن کم بشه. به عنوان مثال این شکل رو در نظر بگیرید:



اگه یه سفینه دشمن از نوع **striker** از بالا وارد مپ بشه، بستگی به تعداد گلوله هایی که تو اون موقعیت وجود داره باید از هیل دشمن کم بشه.



در ابتدا سفینه دشمن وارد می شه و با تشخیص اینکه 2 گلوله تو موقعیت اون وجود داره از بین میره.

۳. موقع استفاده از فایل زمانی که بازی به پایان میرسه (Win Or Game Over) اگه فایلی وجود داشت باید پاک بشه چون اگه بازیکن مجدد میخواست بازی کنه ، بازی جدید باید ساخته بشه.

۴. اگه موقعیت های سفینه دشمن یا گلوله ها یا ... رو در آرایه ای جداگانه (جدای map) نگه داری میکنید , موقع از بین رفتن اون سفینه دشمن یا گلوله از map باید اطلاعات مرتبط با اون تو آرایه ای که جدا درنظر گرفتید نیز پاک بشه.
به عنوان مثال فرض کنید یه آرایه در نظر میگیریم برای نگه داری موقعیت گلوله ها:

```
struct Bullet
{
    int x ;
    int y ;
};
```

```
Bullet arr[Size] ;
```

اگه موقعیت گلوله ها رو این شکلی نگه داری می کنید, زمان از بین رفتن گلوله در مپ باید این آرایه هم آپدیت بشه.

۵. میتونید برای تعاریف ANSI escape codes از شکل زیر الهام بگیرید:

```
// ANSI escape codes for text color
#define RED_TEXT "\033[1;31m"
#define Green_Text "\033[0;32m"
#define Blue_Text "\033[0;34m"
#define Yellow_Text "\033[0;33m"
#define Magenta_Text "\033[0;35m"
#define Cyan_Text "\033[0;36m"
#define Black_Text "\033[0;30m"
#define White_Text "\033[0;37m"

// background color
#define White_Background "\033[47m"

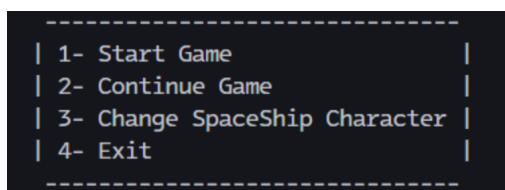
// default mode
#define RESET_TEXT "\033[0m"
```

۶. درنظر داشته باشید که سفینه خودی نمی تونه از مپ خارج بشه و در صورت خروج از مپ پیغام خطای مناسب چاپ شود.

موارد مربوط به پیاده سازی:

۱. از struct استفاده کنید.
۲. از فایل استفاده کنید.(استفاده نکردن از فایل و ذخیره نکردن اطلاعات به صورت مناسب و طی هر حرکت بازیکن رسماً بخش عظیمی از نمره‌ی شما از بین می‌بره!)
۳. استفاده از دستور goto و متغیرهای global مجاز نیست.
۴. از آرایه یک بعدی برای مپ استفاده نکنید.(در صورت استفاده از آرایه‌ی یک بعدی، نمره‌ی کل بخش پیاده‌سازی مپ رو از دست خواهید داد!)

۵. از Menu برای تعامل با کاربر استفاده کنید. به این صورت که گزینه های مختلفی با قابلیت های متفاوت وجود داشته باشد . به عنوان مثال تصویر زیر رو ببینید:



درنظر داشته باشید که این menu صرفا برای نمونه طراحی شده و باتوجه به نیاز های بازی گزینه های دیگر رو هم قرار بدین.(قاعداً منوهای مختلفی که باید در هر قسمت و برای راحتی کاربر تعریف کنید، با توجه به موقعیت متفاوت هستن و این عکس فقط یک نمونه هست! نه حتی یک منوی اصلی...! خودتون موظفید منوهای مربوط به قسمت های مختلف رو طراحی کنید).

۶. پروژه پایانی دارای **۵۰ درصد کاتاف** می باشد(در صورت کسب حداقل ۵۰ درصد از نمره ی پروژه پایانی شرط گذراندن درس را پاس خواهید کرد).

۷. نوشتن گزارش کار برای پروژه **الزامی** ست. عدم ارائه ی گزارش کار به منزله ی عدم ارائه ی پروژه پایانی بوده و نمره ی صفر لحاظ خواهد شد. فایل گزارش کار شما باید مطابق با استانداردهای فایل های گزارش کار بوده و فارغ از هر نوع طراحی گرافیکی باشد. (برخی از استانداردها: وجود لوگوی دانشگاه، پس زمینه ی ساده (ترجیحاً سفید)، نام استاد درس، نام اعضای تیم، ترم تحصیلی، شماره صفحه، فهرست، توضیح کلی و سپس توضیحات ضروری دیگر مانند چالش های پیاده سازی، ارور هندلینگ، توضیح توابع و یا بخش های کلیدی پروژه و... توضیحات بیشتر در رابطه با طراحی و آماده سازی گزارش کار استاندارد، یا به صورت متن در گروه درس در تلگرام و درس افزار و یا به صورت آنلاین در گوگل میت به دانشجویان ارائه خواهد شد).

۸. پیاده سازی پروژه به صورت تک نفره یا تیم دو نفره قابل قبول است.

۹. فایل گزارش کار رو به صورت pdf به همراه فایل پروژه به صورت zip. در بخش مخصوص پروژه پایانی در Quera آپلود کنید.

۱۰. پروژه پایانی شامل ارائه می باشد و عدم اعلام آمادگی برا ارائه به منزله تقلب می باشد و نمره ی پروژه شما صفر خواهد شد.

بخش امتیازی:

۱. اضافه کردن قابلیت به عنوان Level در کنار سایر ویژگی های موجود (امتیاز و هیل سفینه خودی و سائز مپ). به این صورت که اگر کاربر از ۲۰۰ امتیاز گذر کرد ، یکی به Level های اون افزوده بشه.

۲. نشان دادن تعداد دشمنان و نوع هایشان که درحین بازی از بین رفته اند در انتهای بازی (Win Or Game Over)

۳. نگه داری تاریخچه بازی ها در فایل هایی جداگانه (به عنوان مثال اگر کاربری ۳ بار بازی رو به اتمام رسونده ، اطلاعات مربوط به بازی ها در فایلی جداگانه وجود داشته باشه و قابل نمایش باشه).

Example :

Game 1 - Level : 3 - Point : 400 - Heal : 2 - Win

Game 2 - Point : 153 - Heal : 0 - Lose

Game 3 - Point 0 - Heal : 3 - Continue

۴. وجود داشتن ۳ نوع سفینه خودی با قابلیت شلیک گلوله با قدرت های متفاوت:(قدرت شلیک نمایانگر تعدادی هیلی (Heal) که از دشمن کم میکنه).

نوع سفینه	قدرت شلیک
Space Ship 1	1
Space Ship 2	2
Space Ship 3	3

۵. همزمان چندین سفینه دشمن insert شوند (در نظر داشته باشید که به دلیل محدود بودن سائز مپ باید دشمنانی که وارد میشوند از نظر سائز تداخل نداشته باشن و در مپ جا بشن).

۶. زمان بندی کردن حرکت های حین بازی به این صورت که شلیک گلوله ها وابسته به اکت های کاربر بازی نداشته باشه و به طور خودکار بعد از گذشت یه بازه ی زمانی خاصی گلوله ها پشت سر هم شلیک بشن (مثلا هر نیم ثانیه یه گلوله شلیک بشه).

۷. دشمنانی که وارد بازی می شن سطح بندی بشن و اگر کاربر به طور مثال تونست از مقدار امتیاز ۵۰ تا بالاتر بره، دیگه هیچ دشمنی از نوع ۱ وارد بازی نشه و فقط از نوع دوم به بالا وارد مپ بشن.

۸. سوال خوب پرسیدن در استک-اور-فلو: در برنامه نویسی سوالات زیادی برای شخص بوجود می آید که خب طبیعتا باید به جواب آنها برسه و هرچه زودتر بهتر. گاهی هم ممکن است سوالی که ما داریم تا حال برای کسی پیش نیامده و اونجاس که باید خودمون طرح کننده سوال باشیم و این خیلی مهمه که بتونیم بطور درست سوال طرح کنیم تا بقیه هم بتونند راحت به ما پاسخ بدن. در این پروژه مقداری نمره مثبت برای طرح سوال خوب گذاشته شده که شما هم مشتاق بشید تا با بقیه مردم دنیا ارتباط برقرار کنید و مشکلاتتون رو برطرف کنید.

گیت

استفاده از گیت در پروژه‌ی پایانی اختیاری ست. کامنت‌های پروژه‌ی شما بررسی خواهند شد و نباید کامیت غیرعادی داشته باشید. پیشنهاد می‌شود بعد از هر تغییر کوچکی در کد، در انتهای آن تغییر یک کامیت ثبت کنید و توضیح مناسبی برای آن درج کنید.

- مثال از کامیت غیرعادی:

کل پروژه در کمتر از ۳۰ کامیت ثبت شود.

کامیت‌های نامفهوم/بی‌مفهوم در جهت طبیعی سازی کامیت‌ها وجود داشته باشد. نمره‌ی مثبت گیت تنها به پروژه‌هایی تعلق خواهد گرفت که این شرایط را رعایت کرده باشند.

مصادیق تقلب

- استفاده مستقیم از کد بقیه دانشجویها و یا منابع دیگر؛ که در صورت رویت، نمره پروژه متقلب و تقلب‌رسان پروژه صفر میشود.
- اگر از کد پروژه‌های متن باز استفاده کردید، حتما در گزارش کار قید کنید که چه مقدار و در کجا از آنها استفاده کردید. و حتما روش پیاده سازی آن را باید بدانید؛ چرا که در جلسه ارائه از آنها سوال پرسیده میشود.
- استفاده از کدهای متن باز، کامال هم مجاز نیست. اگر مقدار زیادی از برنامه را کپی کرده باشید، نمره آن قسمت را نمی‌گیرید. (اگر بدون ذکر منبع استفاده کنید، ممکن است کل نمره پروژه را نگیرید.)

- مثال‌های استفاده غیرمجاز:

- اگر یک کالس، یا حتی یک تابع، را عینا و بدون تغییر کپی کنید.
- اگر یک پروژه یا یک ماژول را صرفا با تغییر نام و جابهجایی عناصر کپی کنید.

موفق باشید! :

