

برنامه‌سازی پیشرفته

تمرین کامپیوتری شماره ۵



مدرس: رامتین خسروی

طراحان: کوروش علی‌نقی، سروش صحرائی، امیرعلی وحیدی، آرین باستانی، سهیل حاجیان منش

مهلت تحویل: چهارشنبه ۱۴ آذر ۱۴۰۳، ساعت ۲۳:۵۹

مقدمه

هدف از این تمرین آشنایی شما با برنامه‌نویسی شیء‌گرای رویدادمحور^۱ و استفاده از آن در کنار کتابخانه‌های گرافیکی است. انتظار می‌رود از فنون برنامه‌نویسی که تاکنون در کلاس درس فراگرفته‌اید یا در هنگام تحویل تمرین‌ها به شما تذکر داده شده است را 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 SFML^۲ استفاده کنید و همچنین باید با مفاهیمی چون ساخت پروژه به صورت چندفایلی و استفاده از گیت آشنا باشید. قبل از شروع به انجام این تمرین توصیه می‌شود حتما مستندات موجود در این [لینک](#) را مطالعه کنید.

^۱ Event Driven Programming

^۲ Simple and Fast Multimedia Library

Sheep Fight



شما می‌توانید این بازی را از طریق این [لینک](#) روی گوشی خود نصب کرده و بازی کنید و با محیط بازی آشنا شوید. توجه کنید که جزئیات این تکلیف با جزئیات بازی تفاوت دارد، بنابراین صورت تکلیف را با دقت مطالعه کنید.

معرفی

بازی Sheep Fight، یک بازی دو نفره است که در آن بازیکنان علیه یکدیگر به رقابت می‌پردازند. نقشه بازی متشکل از چند سطر است که گوسفندان در امتداد یک سطر از طرفی به طرف دیگر حرکت می‌کنند. هدف بازی این است که بازیکن بتواند گوسفندان خود را به خانه بازیکن مقابل رسانده و همچنین از رسیدن گوسفندان حریف به خانه خود جلوگیری کند. در ادامه به جزئیات بیشتر روند بازی می‌پردازیم.

نقشه بازی

همانطور که گفته شد، نقشه بازی به صورت تعدادی سطر موازی است که گوسفندان هر بازیکن از یک طرف آن (مثلاً سمت چپ برای بازیکن اول و سمت راست برای بازیکن دوم) وارد نقشه شده و در امتداد آن به سمت طرف مقابل حرکت می‌کنند. توجه کنید که امکان جابه‌جایی گوسفندها بین دو سطر وجود ندارد.



توجه کنید که در نسخه اصلی بازی، نقشه به صورت عمودی است اما در نسخه‌ای که شما پیاده‌سازی خواهید کرد، نقشه به صورت افقی قرار دارد.

روند بازی

مقدمات جنگ

در ابتدای بازی، هر بازیکن یک جان مشخص دارد و با یکدیگر برابر است. مقدار جان بازیکنان باید در صفحه در هر لحظه از بازی نمایش داده شود.

در بازی، انواع مختلفی گوسفند داریم که هر کدام در ویژگی‌های مختلفی از قبیل قدرت، میزان صدمه و اندازه با یکدیگر تفاوت دارند. توجه کنید که برای اینکه نوع گوسفند مشخص باشد از نمایه‌های مختلفی استفاده کنید. در آموزش‌های داده شده در مخزن گیت‌هاب، تعدادی راهنما برای پیدا کردن این نمایه‌ها داده شده است. توجه داشته باشید که این انواع برای هر دو بازیکن یکسان است اما برای تشخیص گوسفندان بازیکنان در بازی، نیاز است تا گوسفندان بازیکن ۱ با گوسفندان بازیکن ۲ از لحاظ بصری با هم تفاوت داشته باشند. به طور مثال در بازی اصلی، گوسفندان یک بازیکن سیاه و گوسفندان بازیکن دیگر سفید هستند.

در بازی برای هر بازیکن یک صف از گوسفندان وجود دارد و باید در صفحه بازی، ۳ گوسفند ابتدای صف هر بازیکن نمایش داده شوند. مکان قرارگیری این پنل (تا جایی که روی نقشه قرار نگیرد) به اختیار شماست. توجه کنید که نوع گوسفندانی که در صف قرار دارند باید تصادفی باشد و در این انتخاب تصادفی، انواع مختلف گوسفندان با توجه به احتمال نمایششان، شانس متفاوتی برای حضور در صف دارند. این پارامترها (قدرت، میزان صدمه و احتمال نمایش) باید در کد قابل تنظیم باشند و می‌توانید این پارامترها را در فایل‌های دیگری (مانند جیسون³ و ...) قرار دهید یا برای راحتی می‌توانید پارامترها را در فایل به شکل زیر مقداردهی کنید (بعضی از پارامترها جلوتر توضیح داده خواهند شد):

```
#ifndef CONFIGS_HPP_INCLUDE
#define CONFIGS_HPP_INCLUDE

#include <vector>
#include <string>

struct SheepConfigs {
    std::string name;
    int damage;
    int strength;
    float displayProb;
};

const std::vector<SheepConfigs> SHEEP_CONFIGS = {{
    {"Timmy", 50, 50, 0.5f},
    {"Shaun", 30, 150, 0.3f},
    {"Meow", 10, 250, 0.2f},
}};

constexpr int COOLDOWN_MS = 2000;
constexpr int INITIAL_HEALTH = 400;
constexpr int CONSTANT_SPEED = 300;

#endif // CONFIGS_HPP_INCLUDE
```

دقت کنید که می‌توانید به شکل دلخواه خود این فایل را تغییر دهید و الزامی به استفاده دقیقاً به همین شکل نیست تا زمانی که به شکل مشخصی تعریف شده باشند و به راحتی قابل تغییر باشند و همچنین همه این پارامترها تعریف و در کد استفاده شده باشند. همچنین دقت داشته باشید که مقادیر داده شده باید توسط خودتان تنظیم شوند و به گونه‌ای باشند که بازی قابل بازی کردن باشد.

³ json

شروع جنگ

هر بازیکن یک صف با طول ۳ از گوسفند دارد که هر بار فقط می‌تواند از گوسفندی که در ابتدای صف است در یکی از سطرها استفاده کند و آن را به سمت حریف روانه⁴ کند. توجه داشته باشید که گوسفندان یک بازیکن تا انتهای بازی تمام نمی‌شوند و در صورت روانه کردن گوسفندی که در ابتدای صف است، گوسفند بعدی به ابتدای صف می‌آید و یک گوسفند جدید به انتهای صف اضافه می‌شود. نوع این گوسفند جدید همانطور که اشاره شد، باید به صورت تصادفی و متناسب با احتمالی که برای هر نوع گوسفند در پارامترها آمده باشد. پس از انتخاب گوسفندی که در ابتدای صف است و روانه کردن آن به سمت حریف، مدت زمان ثابتی طول می‌کشد تا بازیکن بتواند گوسفند بعدی را روانه کند. این زمان آتش‌بس⁵ نام دارد و باید برای هر دو بازیکن یکسان باشد (این زمان یکی دیگر از پارامترهای موجود در فایل کانفیگ است). سرعت حرکت همه گوسفندان تا وقتی که به هم برخورد نکرده باشند، با هم برابر و ثابت است و مقدار این سرعت در پارامترهای فایل کانفیگ مشخص می‌شود. گوسفندانی که از دو طرف یک سطر به سمت یکدیگر حرکت می‌کنند، مادامی که به گوسفندی در مسیر خود برخورد نکرده‌اند به حرکتشان ادامه می‌دهند.

شاخ به شاخ!

لحظه‌ای که گوسفند بازیکن به گوسفند حریف در یک سطر می‌رسد، گوسفندان شاخ به شاخ شده و یکدیگر را هل می‌دهند! در این حالت، جهت حرکت گوسفندان، به سود بازیکنی است که گوسفند قدرتمندتری دارد. سرعت حرکت گوسفندان به سمت خانه یکی از بازیکنان نیز متناسب با اختلاف مجموع قدرت گوسفندان دو طرف است. در بعضی مواقع، با اضافه شدن گوسفندان دیگر به سطر مورد نزاع و برخورد با گوسفند(ان) حریف، شاخ به شاخ دسته‌جمعی شکل می‌گیرد! در این حالت نیز جهت حرکت و سرعت حرکت گوسفندان متناسب با مجموع اختلاف قدرت دو طرف درگیری مشخص می‌شود.

همانطور که گفته شد، هدف بازی این است که هر بازیکن بتواند گوسفندان خود را به خانه حریف برساند. به ازای هر گوسفندی که به خانه حریف برسد، از جان بازیکن حریف به میزان مشخصی کم می‌شود. این مقدار، معادل با میزان صدمه‌ای⁶ است که برای هر گوسفند تعیین شده است. این میزان صدمه با کمک پارامترهای فایل کانفیگ تعریف می‌شود.

دقت کنید بعد از رسیدن گوسفند (چه حریف و چه خودی) به خانه یکی از بازیکنان، گوسفند از صفحه بازی محو می‌شود.

⁴ Launch

⁵ Cooldown

⁶ Damage

پیروزی عظیم!

هنگامی که جان بازیکنی بر اثر رسیدن گوسفندان حریف صفر یا منفی شود، بازی به پایان می‌رسد و بازیکن حریف برنده بازی خواهد بود. مطمئن شوید که با پیغام مناسبی بر روی صفحه این پیروزی را به بازیکن برای جشن گرفتن اعلام کنید!



نحوه انتخاب سطر برای روانه کردن گوسفند

همانطور که گفته شد، بازی دو نفره است. هر بازیکن در هر لحظه، یکی از سطرها را برای خودش انتخاب کرده است و می‌تواند بین سطرهای خود بالا و پایین برود. با فشردن کلیدی مشخص (در صورتی که از آخرین روانه کردن گوسفند آن بازیکن حداقل به اندازه مدت زمان آتش‌بس گذشته باشد)، گوسفندی که در ابتدای صف گوسفندان آن بازیکنان قرار دارد از آن سطر شروع به حرکت می‌کند. مهم است که سطر انتخاب شده برای هر

بازیکن به نحوی مشخص باشد تا بازیکن بداند در حال حاضر کدام سطر را انتخاب کرده است، به طور مثال می‌توانید رنگ آن سطر را عوض کنید و یا یک فلش در ابتدای سطر نشان دهید. نحوه مشخص کردن سطر انتخاب شده به دلخواه خودتان می‌باشد. در جدول زیر، کلیدهایی مربوطه برای هر بازیکن آمده‌است:

انتخاب سطر بالاتر	انتخاب سطر پایین‌تر	روانه کردن گوسفند از سطر انتخاب شده	
Arrow Up	Arrow Down	Enter	بازیکن ۱
w	s	Space	بازیکن ۲

این که با فشردن کلید بالا وقتی بالاترین سطر انتخاب شده چه اتفاقی رخ بدهد به عهده خودتان است. می‌توانید روی همان سطر بمانید و می‌توانید به پایین‌ترین سطر بروید. اگر در زمان آتش بس اقدام به روانه کردن گوسفند خود کنید، نباید اتفاق خاصی رخ دهد.

موسیقی

همچنین پس از شروع بازی آهنگی (ترجیحاً حماسی!) در پس‌زمینه شروع به پخش شدن می‌کند تا حس غرور و رقابتی شما را برانگیزد!

نکات تکمیلی

- در این تمرین تعدادی پارامتر وجود دارد (مانند جان ابتدایی بازیکنان، تعداد سطرهای نقشه، میزان صدمه و قدرت انواع مختلف گوسفندان 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 بخش‌های خواسته شده مشخص باشد (خیلی روی اعداد دقیق حساس نباشید). در صورت تمرین کلیت انتظارات از بازی بیان شده و در تصمیم‌گیری شیوه پیاده‌سازی آن‌ها (مانند ساختار نقشه ورودی، فیزیک بازی، اندازه اشیا و ...) دست شما باز است. هدف نهایی این است که برنامه نهایی قابل بازی کردن باشد.
- برای اینکه در زمان تحویل برای دستیار مربوطه مشکلی ایجاد نشود ابعاد صفحه را حداکثر $1080 * 720$ در نظر بگیرید.
- توصیه می‌شود که پروژه را از ساده به پیچیده جلو ببرید و ابتدا کلیت بازی را پیاده‌سازی کنید. به عنوان مثال می‌توانید ابتدا فقط حرکت گوسفندان در صفحه را بنویسید. سپس در طی چند مرحله، برنامه را پیچیده‌تر کنید. همچنین روند ارزیابی شما به صورتی قسمت‌بندی شده است که حتی در صورتی که پروژه را کامل نکرده باشید، نمره مربوط به قسمت‌های کامل شده به شما تعلق بگیرد. به عنوان مثال، ابتدا صرفاً حرکت گوسفندان در سطرها را پیاده‌سازی کنید. بعد از آن مکانیزم هل دادن گوسفندان و به همین ترتیب موجودیت‌های بازی را کامل کنید.
- دقت کنید که تصمیم‌گیری در خصوص مواردی که به طور صریح در صورت تمرین ذکر نشده، بر عهده شما بوده و می‌توانید هر فرضی که می‌خواهید را پیاده‌سازی کنید.
- از آنجایی که پروژه باید به صورت چند فایل پیاده‌سازی شود، توصیه می‌شود که پروژه را از ابتدا در چند فایل پیاده‌سازی کنید و از پیاده‌سازی کل پروژه در یک فایل و سپس تبدیل آن به چند فایل خودداری کنید.
- توجه داشته باشید که در این تمرین سبک برنامه‌نویسی رویداد محور را در ساختار کد خود در نظر بگیرید.

نحوه تحویل

- می‌توانید این تمرین را در قالب گروه‌های دو نفره انجام دهید. گروهی انجام دادن این تمرین، الزامی نیست و می‌توانید به صورت تکی نیز آن را تحویل دهید.
- توجه داشته باشید که در صورت گروهی انجام دادن تمرین، هر دو عضو گروه باید اشراف کامل روی تمرین و کد آپلود شده داشته باشند و این موضوع در تحویل تمرین در نظر گرفته می‌شود. در صورت عدم رعایت این مورد، نمره‌ای برای فردی که روی کد اشراف ندارد، در نظر گرفته نمی‌شود. همچنین، هر دو نفر باید مشارکت کافی در انجام پروژه را داشته باشند؛ این موضوع از طریق `commit` ها در گیت، بررسی خواهد شد.
- در صورت انجام تمرین به صورت گروهی، آپلود تمرین از طرف یکی از اعضای گروه کفایت می‌کند.
- برای تحویل این پروژه، لازم است کد خود را در یک مخزن⁷ در GitHub بارگذاری کنید و سپس لینک مخزن به همراه Hash آخرین کامیت⁸ را در صفحه eLearn درس بارگذاری نمایید.
نمونه متن خواسته شده در سامانه ای‌لرن (سه بخش `<username>` و `<repository_name>` و `<last_commit_hash>` را جایگزین کنید):

```
https://github.com/<username>/<repository_name>  
<last_commit_hash>
```

- پروژه شما باید به صورت **چند فایل**⁹ و با استفاده از **makefile** پیاده‌سازی شده باشد. دقت کنید که فایل `makefile` باید در صفحه اول مخزن باشد و در پوشه‌ای قرار نداشته باشد و در آن مشخص کنید که از استاندارد C++20 استفاده می‌کنید.
- فایل‌های خود را در مخزنی به نام `<SID>-A5-F03-AP` قرار دهید (دقت کنید که به جای SID از شماره دانشجویی خود استفاده کنید) یا اگر به صورت گروهی پروژه را انجام می‌دهید به صورت `<SID2>-<SID1>-A5-F03-AP` و گزینه "Private" را انتخاب کنید. همچنین، کاربر **@AP-UT** را به مخزن اضافه کنید.
- سعی کنید مرحله به مرحله کد را `commit` کنید و از `commit` کردن کل کد در انتهای پروژه خودداری کنید. همچنین، پیام `commit`-های شما باید خوانا باشد و تغییر ایجاد شده را توضیح دهد. دقت کنید که بخشی از نمره شما را استفاده صحیح از گیت تشکیل می‌دهد. برای مطالعه در خصوص نحوه صحیح `commit` کردن، می‌توانید [این لینک](#) را مطالعه کنید.

⁷ Repository

⁸ Commit

⁹ multi-file

- نام برنامه قابل اجرای شما باید SheepFight (بدون هیچ پسوندی مانند exe یا out) باشد و پس از ساخته شدن در کنار makefile قرار بگیرد (داخل پوشه‌ای فایل خروجی ساخته شده را قرار ندهید).
- توجه کنید که رعایت نکردن ساختار گفته شده در **نام‌گذاری مخزن، فایل کد، فایل اجرایی و آپلود موارد خواسته شده** باعث کسر 5 درصد از نمره شما خواهد شد.
- برای ایجاد رابط کاربری گرافیکی¹⁰ و تمامی افکت‌های برنامه خود باید از کتابخانه SFML استفاده کنید.
- مخزن شما باید پوشه‌ی کامل پروژه باشد که شامل کد کامل برنامه شما به همراه تصاویر و سایر موارد است.
- در این تمرین بازی شما توسط دستیاران آموزشی آزموده می‌شود و آزمون 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 حتما طراحی شی‌گرا داشته باشد.
- هدف این تمرین یادگیری شماست. لطفا تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

¹⁰ GUI

نمرات

- تمیزی کد

- رعایت کردن نام‌گذاری صحیح و انسجام¹¹
- عدم وجود کد تکراری
- رعایت دندان‌گذاری¹²
- عدم استفاده از متغیرهای گلوبال
- استفاده صحیح از متغیرهای ثابت به جای Magic Value-ها

- بازی

- پیاده‌سازی موارد ذکر شده در صورت پروژه

- طراحی

- کاهش Coupling
- رعایت Encapsulation
- طراحی مناسب چند فایل
- استفاده از Header Guard-ها
- استفاده صحیح از Makefile
- ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند

- گیت

- پیام‌های خوانا برای commit-ها
- Commit کردن مرحله به مرحله
- مشارکت همه اعضای گروه و اشراف کامل بر روی کد

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

¹¹ Consistency

¹² Indentation