



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 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^۲ استفاده کنید. برای راحتی کار شما، در این **لینک** یک کتابخانه‌ی واسط به نام RSDL^۳ 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. **ویدیوهای** جهت آشنایی و آموزش ایجاد برنامه‌های چندفایلی با استفاده از makefile در صفحه‌ی درس قرار داده شده که می‌توانید از آن‌ها استفاده کنید.

^۱ Event Driven Programming

^۲ Simple DirectMedia Layer

^۳ Ramtin Simple DirectMedia Layer



مینی گلف!

مینی گلف یک بازی ورزشی مشابه گلف است که در زمینی کوچک و با موانع مختلف بازی می‌شود. برای آشنایی بیشتر با نمونه کامپیوتری این 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.

روند بازی

این بازی در واقع یک شبیه‌سازی ساده از بازی گلف است. در این بازی باید توپ را در طی یک یا چند ضربه به سوراخ بیندازیم. موانعی وجود دارد که برخورد با آن‌ها منجر به باخت در بازی می‌شود. همچنین سرعت توپ بر اثر اصطکاک رفته رفته کم می‌شود و نهایتاً می‌ایستد. زمین بازی یک مستطیل محصور به دیوار است. همچنین ممکن است دیوارهایی درون این مستطیل نیز وجود داشته باشد که با برخورد با آن‌ها جهت حرکت توپ تغییر کند. بازی زمانی به پایان می‌رسد که توپ

به داخل سوراخ بیفتد یا همانطور که در ادامه توضیح داده خواهد شد بازیکن ببازد. در حین بازی، تعداد دفعاتی که به توپ ضربه وارد شده است، در صفحه نمایش داده می‌شود. همچنین هنگام پایان بازی پیام برد یا باخت نشان داده شود.

اجزای بازی

در این بازی اجزای مختلفی وجود دارد که در ادامه توضیح داده خواهد شد. شکل و ظاهر این اجزا کاملاً به اختیار شما است و فقط لازم است که از هم قابل تشخیص باشند.

سطح: زمین بازی را تشکیل می‌دهند و شکلی مستطیلی دارند. توپ روی سطوح حرکت می‌کند. در کل سه نوع سطح داریم:



- **سطح چمنی (عادی):** توپ به صورت عادی روی این سطح حرکت می‌کند. این سطح دارای اصطکاک است و توپ پس از مدتی حرکت روی آن متوقف می‌شود.
- **آب:** در صورتی که توپ وارد این سطح شود بازنده می‌شویم و بازی تمام می‌شود.
- **سطح شنی (اصطکاک‌دار):** ضریب اصطکاک این سطح بیشتر از سطح عادی است و زمانی که توپ در این سطح حرکت می‌کند بیشتر از سطح چمنی از سرعتش کاسته می‌شود. پیاده‌سازی این بخش اختیاری است و نمره امتیازی دارد.

منظور از روی یک سطح بودن این است که مختصات توپ درون آن سطح باشد.

دیوار: مانعی است که توپ نمی‌تواند از آن عبور کند و پس از برخورد با آن بازتاب می‌شود. دیوار را بصورت یک



پاره‌خط در نظر می‌گیریم. در کل سه نوع دیوار داریم:

- **دیوار عادی:** توپ پس از برخورد با این نوع دیوار بازتاب می‌شود و به حرکت خود ادامه می‌دهد.
- **دیوار خاردار:** اگر توپ با این دیوار برخورد کند بازنده می‌شویم و بازی تمام می‌شود.

● **دیوار متحرک:** مانند دیوار عادی است اما در یک راستای مشخصی به صورت رفت و برگشتی حرکت می کند.

این راستا در واقع همان راستای پاره خط است. پیاده سازی این بخش اختیاری است و نمره امتیازی دارد.

توپ:

● **نحوه پرتاب:** پرتاب توپ از طریق ماوس کنترل می شود. در پرتاب توپ دو عامل وجود دارد:

۱. جهت پرتاب: با کشیدن ماوس از روی توپ به یک سمت، توپ در جهت مخالف پرتاب می شود.

۲. شدت پرتاب: میزانی که ماوس کشیده می شود (فاصله نقطه ای که دکمه ماوس رها می شود تا نقطه ای

که کلیک شده بوده)، شدت پرتاب (سرعت اولیه پرتاب توپ) را تعیین می کند.

برای نشان دادن جهت و شدت پرتاب، از یک خط استفاده می شود. جهت خط در جهت پرتاب توپ (و نه جهت ماوس) و طول خط، شدت پرتاب توپ را نشان می دهد.

برای محاسبه پرتاب اولیه می توانید از فرمول زیر استفاده کنید (توجه کنید که این کار اجباری نیست و می توانید از فرمول دلخواه دیگری استفاده نمایید):

$$\begin{aligned} d &= prevMousePosition - currentMousePosition \\ v &= \frac{d}{throwRadius} \times maxInitSpeed & ||d|| < throwRadius \\ v &= \frac{d}{||d||} \times maxInitSpeed & ||d|| \geq throwRadius \end{aligned}$$

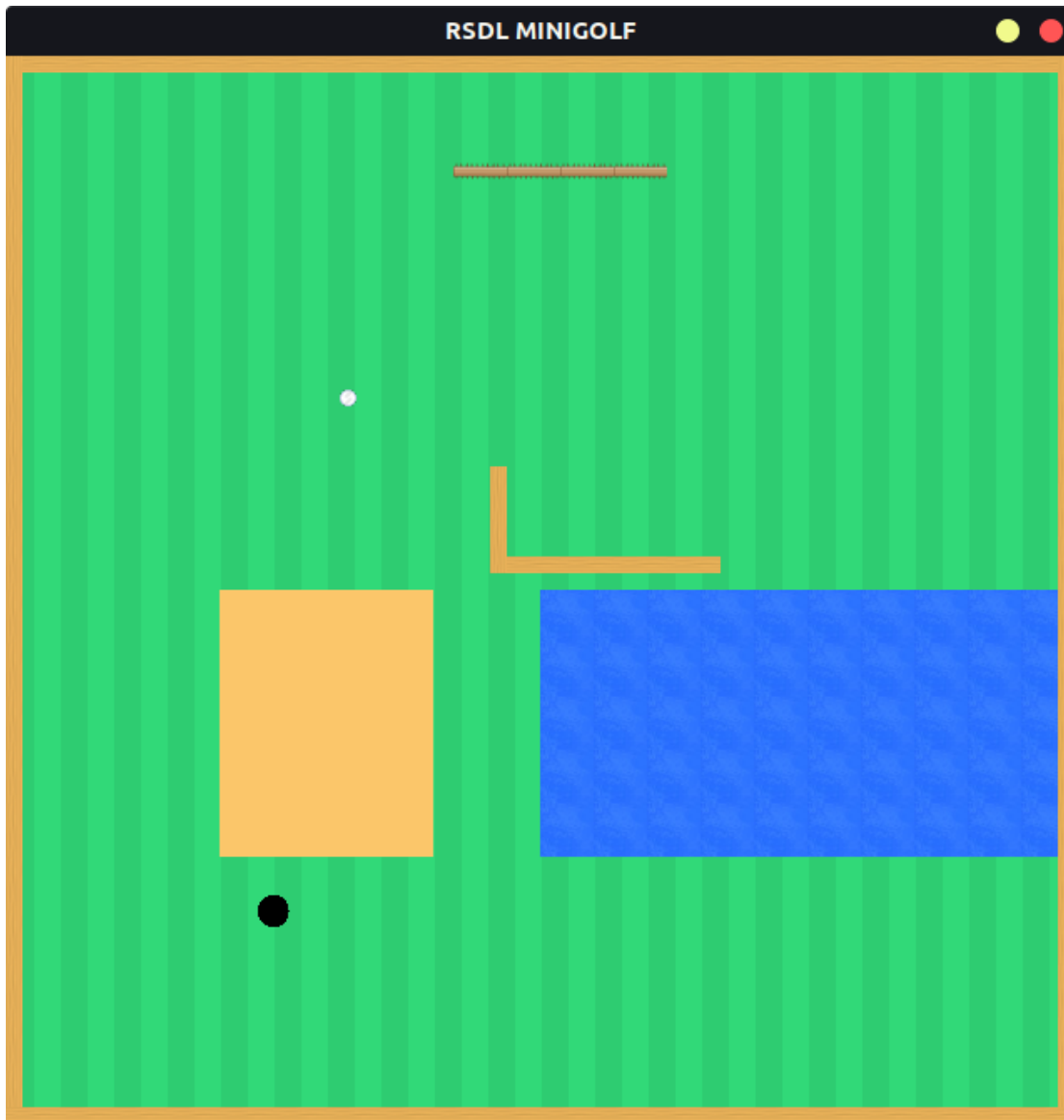
در این فرمول `prevMousePosition` مکان اولی است که کلیک چپ ماوس فشار داده شده و `currentMousePosition` مکانی است که کلیک چپ ماوس در آن جا رها شده. $||d||$ به معنای اندازه این بردار است. `throwRadius` نیز فاصله ای است که در صورت قرارگیری موس در آن فاصله یا بیشتر نسبت به مکان اولیه، توپ با بیشترین سرعت پرتاب خواهد شد. `maxInitSpeed` هم بیشترین مقدار سرعت اولیه توپ است. این دو عدد را می توانید به دلخواه تغییر دهید و آزمایش کنید.

- **اصطکاک:** سرعت حرکت توپ در طول حرکت کم می‌شود و نهایتاً می‌ایستد. در واقع همواره نیروی اصطکاک سرعت حرکت توپ را کاهش می‌دهد. محاسبه فرمول و پیاده‌سازی نیروی اصطکاک به عهده شماست.
- **نحوه برخورد با دیوار:** می‌توان برخورد توپ با دیوارها را به صورت کاملاً کشسان در نظر گرفت. به این صورت که تنها جهت توپ با برخورد به دیوارها تغییر کند و این برخورد بر روی سرعت توپ تاثیری نگذارد. نحوه محاسبه و پیاده‌سازی برخورد توپ به عهده شماست.

نقشه بازی

نقشه بازی باید به صورت فایل به برنامه داده شود که محل اولیه اجزای مختلف در زمین را توصیف می‌کند. ساختار فایل ورودی کاملاً به عهده شما است. ساختاری که انتخاب می‌کنید باید بتواند تمام اجزای بازی (مانند محل شروع توپ، محل قرار گیری سوراخ، محل و مساحت سطوح و ...) را توصیف کند و قابلیت تغییر دادن را داشته باشد. در انتها باید یک فایل نقشه نمونه که شامل تمام اجزای پیاده سازی شده توسط شما است را در کنار کدهای این بخش آپلود کنید.

برای مثال به ساختار نقشه زیر توجه کنید:



```
window_size 640 640
normal_area 0 0 640 320
sand_area 128 320 400 584
water_area 320 320 640 480
normal_area 0 320 256 584
normal_area 256 320 400 480
normal_area 0 480 640 640
ball 200 200
```

```
wall 0 0 640 10
wall 0 0 10 640
wall 630 0 10 640
wall 0 630 640 10
sharp_wall 268 64 128 10
wall 300 300 128 10
wall 290 246 10 64
hole 160 512
```

در این توصیف، ابتدا نام موجود و سپس مختصات آن و در صورت نیاز عرض و ارتفاع آن داده شده است. توجه کنید که گوشه بالا چپ صفحه مختصات (0, 0) است. لزومی ندارد شما نیز با همین ساختار نقشه را تولید کنید و می‌توانید به شکل دلخواه دیگری این کار را انجام دهید.

پایان بازی

در صورتی که بتوانیم توپ را به سوراخ بیندازیم برنده می‌شویم. اگر توپ به آب یا دیوار خاردار برخورد کند می‌بازیم و بازی تمام می‌شود. پایان بازی باید با پیامی مناسب به بازیکن اعلام شود.

نکات تکمیلی

- در این تمرین تعدادی پارامتر وجود دارد (مانند اندازه اشیا در صفحه، سرعت اولیه توپ و اصطکاک سطوح و...). از شما انتظار می‌رود که طبق صلاح‌دید 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. برنامه نهایی باید قابل بازی کردن باشد.
- توصیه می‌شود که پروژه را از ساده به پیچیده جلو ببرید و ابتدا کلیت بازی را پیاده‌سازی کنید. به عنوان مثال می‌توانید ابتدا توپ را با تعداد اجزای کمتری از بازی در صفحه به حرکت درآورید. 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 پرتاب توپ و حرکت بدون اصطکاک را داخل محوطه بازی پیاده‌سازی کنید. بعد از آن اصطکاک را اضافه کنید، بعد پایان بازی را، و به همین ترتیب اجزای بازی را کامل کنید.

- از آنجایی که از این پس پروژه‌ها 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.

نحوهٔ تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی zip با نام A5-<SID>.zip در صفحهٔ ایلرن درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۹۹۹ است، نام پروندهٔ 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  
|— makefile  
|— ...
```

- برای ایجاد رابط کاربری گرافیکی⁴ 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.

⁴ GUI

- نمره هر بخش در صورت 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 minigolf.out باشد.
- نحوه اجرای برنامه به صورت map.txt ./minigolf.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.