

مقدمه

هدف از این تمرین آشنایی شما با برنامه‌نویسی شیء‌گرایی رویدادمحور^۱ و استفاده از آن در کنار کتابخانه‌های گرافیکی است. انتظار می‌رود از تکنیک‌های برنامه‌نویسی که تاکنون در کلاس درس فرا گرفته‌اید یا در هنگام تحویل حضوری تمرین‌ها به شما تذکر داده شده است به طور کامل در این تمرین استفاده کنید.

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.



ستاره‌های فوتبال^۴

ستاره‌های فوتبال یک بازی در سبک بازی‌های ورزشی و فوتبال است. این بازی را وب‌سایت بازی Miniclip تولید و منتشر کرده است. برای آشنایی بیشتر با 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 تمرین را در صورت پروژه مطالعه کنید.

^۱ Event Driven Programming

^۲ Simple DirectMedia Layer

^۳ Ramtin Simple DirectMedia Layer

^۴ Soccer Stars

روند بازی

این بازی یک بازی دو نفره است. بازی شامل یک زمین فوتبال است که تصویر پس‌زمینه⁵ آن به شما داده است. یک توپ در بازی وجود دارد که تصویر آن به شما داده شده است. دو تیم آبی و قرمز هر کدام با ۵ بازیکن در بازی وجود دارند که به نوبت، حرکت خود (ضربه زدن به توپ) را اجرا می‌کنند. هر بازیکن با یک مهره دایره‌ای نشان داده می‌شود که تصویر آن به شما داده شده است. در ابتدای هر دور بازیکن‌های هر تیم به شکل ثابتی مقابل هم قرار می‌گیرند که در ادامه به توضیح کامل آن پرداخته شده است. بازی از n دور تشکیل شده و در هر دور اولین تیمی که m بار توپ را وارد دروازه حریف کند، برنده دور می‌شود. برنده بازی اولین تیمی است که در $n/2$ از دورها پیروز شود.

ورودی‌های برنامه:

ورودی‌های برنامه پارامترهای n و m هستند که به ترتیب تعداد دور⁶ها و تعداد گل⁷های لازم برای پایان هر دور را نشان می‌دهند.

وضعیت اولیه بازی:

برنامه با دریافت این ورودی‌ها، باید صفحه بازی را با چینش اولیه مهره‌ها نشان دهد و منتظر اولین حرکت تیم آبی باشد. چینش

اولیه وضعیتی دلخواه است که خود شما آن را تعیین می‌کنید. صرفاً کافی است هر ۵ مهره‌ی هر تیم، در زمین خود باشند.



مثالی از چینش اولیه بازیکن‌ها

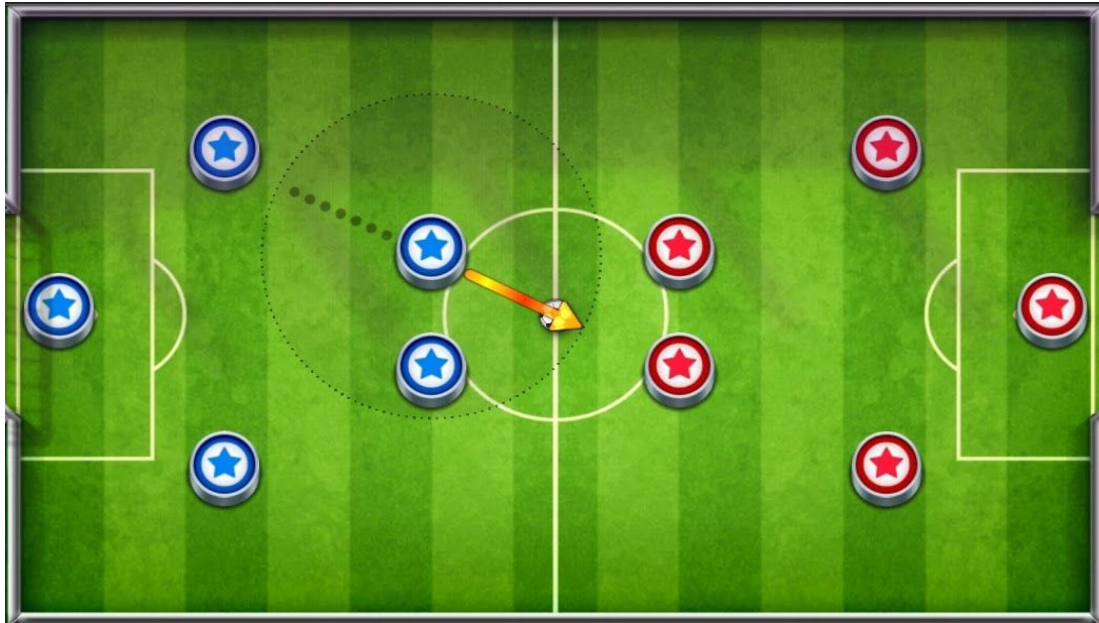
⁵ background

⁶ round

⁷ goal

حرکت‌های بازیکن‌ها:

در هر مرحله، بازیکنی که نوبت او است، با فشردن و نگه‌داشتن دکمه‌ی چپ موس از روی یکی از مهره‌های تیم خود، عملیات ضربه زدن را شروع می‌کند. در این حالت دایره‌ای فرضی وجود می‌شود که فاصله‌ای را نشان می‌دهد که حداکثر سرعت اولیه با گذشتن موس از آن حاصل می‌شود. با حرکت دادن موس، جهت و قدرت ضربه (سرعت اولیه) را مشخص می‌کند. زمانی که بازیکن دکمه‌ی موس را رها می‌کند مهره با سرعت اولیه پرتاب می‌شود. پیشنهاد می‌شود حتما نسخه اصلی بازی را امتحان کنید تا از این موضوع درک بهتری پیدا کنید.



نشان دادن این دایره و فلش جهت حرکت لازم نیست و می‌تواند نمره‌ی امتیازی برای شما داشته باشد.

سرعت اولیه‌ی مهره به این صورت محاسبه می‌شود.

$$d = \text{mousePosition} - \text{pawnPosition}$$

$$\left\{ \begin{array}{ll} v = -\frac{d}{\text{ThrowRadius}} \times \text{MaxInitialSpeed} & \|d\| < \text{ThrowRadius} \\ v = -\frac{d}{\|d\|} \times \text{MaxInitialSpeed} & \|d\| \geq \text{ThrowRadius} \end{array} \right.$$

در این فرمول pawnPosition مکان مرکز مهره و mousePosition مکان نشان‌گر موس است. همچنین $\|d\|$ به معنای اندازه این بردار است. مقادیر پیشنهادی ThrowRadius و MaxInitialSpeed در ادامه نوشته شده‌اند.

فیزیک اجسام و برخوردها:

هر بازیکن و توپ دارای برداری دو بعدی سرعت و بردار دو بعدی مکان هستند. با انجام هر حرکت توسط هر بازیکن، تعدادی برخورد ممکن است رخ دهد، این برخوردها می‌تواند بین بازیکن‌ها یا بازیکن و توپ باشد که به کمک روابط زیر باید آن‌ها را مدیریت کنید:

$$\mathbf{v}'_1 = \mathbf{v}_1 - \frac{2m_2}{m_1 + m_2} \frac{\langle \mathbf{v}_1 - \mathbf{v}_2, \mathbf{x}_1 - \mathbf{x}_2 \rangle}{\|\mathbf{x}_1 - \mathbf{x}_2\|^2} (\mathbf{x}_1 - \mathbf{x}_2)$$

$$\mathbf{v}'_2 = \mathbf{v}_2 - \frac{2m_1}{m_1 + m_2} \frac{\langle \mathbf{v}_2 - \mathbf{v}_1, \mathbf{x}_2 - \mathbf{x}_1 \rangle}{\|\mathbf{x}_2 - \mathbf{x}_1\|^2} (\mathbf{x}_2 - \mathbf{x}_1)$$

در این فرمول‌ها 1 \mathbf{v}' بردار سرعت جسم اول بعد از برخورد و 2 \mathbf{v}' بردار سرعت جسم دوم بعد از برخورد است و همچنین V_1 و V_2 به ترتیب بردارهای سرعت جسم اول و دوم قبل از برخورد است. m در اینجا به معنای جرم جسم است و X بردار مکان اولیه مرکز جسم را نشان می‌دهد. $\langle a, b \rangle$ به معنای ضرب داخلی دو بردار a و b است و $\|a\|$ به معنای اندازه این بردار است.

اجسام بعد از مدتی به دلیل وجود اصطکاک می‌ایستند. اصطکاک را شتاب ثابت برابر ۳۰۰ پیکسل بر مربع ثانیه در خلاف جهت حرکت در نظر بگیرید.

همچنین زمین بازی توسط دیوارهای فرضی احاطه شده است که توپ و بازیکن‌ها در اثر برخورد به آن‌ها بازتاب می‌شود (به جز دروازه‌ها که در صورت برخورد به آن نواحی، توپ باید از صفحه خارج شود)

گل شدن توپ:

اگر توپ از طرفین صفحه به ازای $y \in [160, 320]$ به طور کامل عبور کند، گل محسوب می‌شود. بعد از هر گل، بازیکن‌ها به چیش اولیه خود باز می‌گردند و تیمی که گل خورده است، بازی را شروع می‌کند.

مقادیر پیشنهادی:

اندازه‌ی صفحه (عرض×طول)	۸۰۰×۴۸۰
بیشترین سرعت اولیه (اندازه بردار) (MaxInitialSpeed)	۵۰۰

۶۰ پیکسل	شعاع دایره‌ای که با آن میتوان حداکثر ضربه را زد (ThrowRadius)
۱۵ پیکسل	شعاع مهره‌ها
۱۰ پیکسل	شعاع توپ
۲	جرم مهره‌ها
۱	جرم توپ
۳۰۰ پیکسل بر مربع ثانیه	شتاب اصطکاکی

توجه داشته باشید که این مقادیر پیشنهادی هستند و می‌توانید آن‌ها را تغییر دهید.

نکات تکمیلی

- تعداد گل‌های هر تیم باید روی صفحه قابل رویت باشد.
- عکس مربوط به زمین بازی، توپ فوتبال و مهره‌های هر تیم در پوشه‌ی assets قرار گرفته است که می‌توانید از آن‌ها برای نشان دادن اجزای بازی از آن‌ها استفاده کنید. همچنین می‌توانید برای زیبایی بیشتر بازی از تصاویر مناسب دیگر استفاده کنید.
- در صورتی که تمامی دورها تمام شود، باید تیم برنده و تعداد دورهایی که برنده شده است روی صفحه نمایش داده شود.
- به طور کلی هر گونه خلاقیت و پیاده‌سازی هر موردی که موجب زیبایی بیشتر و جذاب‌تر شدن بیشتر بازی شود، می‌تواند برای شما نمره امتیازی داشته باشد. همچنین دقت کنید که به مواردی که توسط دستیار آموزشی، خلاقیت تشخیص داده شود در مجموع تا سقف ۲۰ درصد نمره امتیازی تعلق می‌گیرد.
- توجه کنید که ممکن است کتابخانه‌ی RSDL به‌روزرسانی شود؛ بنابراین سعی کنید تغییری در کتابخانه ایجاد نکنید تا بتوانید در صورت به‌روزرسانی به‌راحتی از نسخه‌ی جدید آن استفاده کنید.
- شما مجاز به استفاده از وراثت^۸ و نکاتی که در رابطه با آن در کلاس درس آموخته‌اید هستید. همین‌طور استفاده از کتابخانه‌های استاندارد C++ مانعی ندارد.

^۸ Inheritance

نحوه‌ی تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی zip با نام A5-<SID>.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ است، نام پرونده‌ی شما باید A5-810198999.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 ایجاد رابط کاربری گرافیکی^۹ 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 اتوماتیک ندارد.
- نمره هر بخش در صورت کار کردن در بازی شما اختصاص می‌یابد و داشتن کد یک بخش که در بازی قابل امتحان کردن نیست نمره‌ای برای شما ندارد.
- برنامه شما باید حتماً طراحی شیء‌گرا داشته باشد.
- دقت کنید که پرونده‌ی 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 soccerStars.out باشد.
- 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.

^۹ GUI