

۱ پیش‌تمرین RSDL

SDL^۱ یک کتابخانه‌ی چندسکویی، رایگان و متن‌باز است که برای بازی‌سازی و تولید برنامه‌های گرافیکی استفاده می‌شود. کتابخانه‌ی RSDL^۲ یک کتابخانه‌ی کمکی بسته‌بندی‌کننده^۳ است که امکان استفاده‌ی آسان‌تر از SDL2 را فراهم می‌کند. نسخه‌ی فعلی RSDL را می‌توانید از آدرس <https://github.com/UTAP/RSDL> دریافت کنید. همچنین، راهنمای این کتابخانه در آدرس <https://github.com/UTAP/RSDL/wiki> در دسترس است.

در این پیش‌تمرین برنامه‌ای ساده را با کتابخانه‌ی RSDL پیاده‌سازی می‌کنید تا بیشتر با آن آشنا شوید.

با کمک دستور `draw_img` و استفاده از آرگومان `src` آن، می‌توانید تکه‌ای از یک تصویر را روی صفحه رسم کنید. در پوشه‌ی `warmup` تصویری از یک جدول 3×3 است که با اعداد ۱ تا ۹ پر شده. با استفاده از روش بالا برنامه‌ای بنویسید که به صورت تصادفی این جدول را به هم ریخته و روی صفحه رسم کند.

حالا می‌خواهیم با زدن دکمه‌ی R ترتیب خانه‌ها تغییر کند. برای این کار داخل یک حلقه با استفاده از تابع `poll_for_event` و `get_pressed_key` چک کنید که آیا دکمه‌ی R زده شده است یا نه. سپس مستطیل‌ها را دوباره محاسبه کنید و صفحه را به‌روزرسانی کنید.

تصویر زیر پنجره‌ی این برنامه را نشان می‌دهد.

Warmup		
5	9	7
2	1	4
6	8	3

توجه کنید که این بخش برای آشنایی بیشتر شما با RSDL است و نیازی به تحویل آن نیست.

¹ Simple DirectMedia Layer

² Ramtin SDL

³ wrapper

۲ سوپرماریو

[محل درج مقدمه‌ی سوپرماریو]

برای مشاهده‌ی یک پیاده‌سازی کامل از بازی سوپرماریو می‌توانید به آدرس <https://supermariobros.io/> مراجعه کنید.

در این تمرین از شما انتظار می‌رود یک نسخه‌ی ساده از بازی سوپرماریو را با توجه به توضیحاتی که در ادامه می‌آید پیاده‌سازی کنید. تمرین از چند بخش مختلف تشکیل شده است که در ادامه به توضیح هر یک می‌پردازیم.

۱.۲ نقشه

نقشه‌ی بازی به صورت یک جدول ۲ بعدی از کاراکترها به شما داده می‌شود. هر کاراکتر نشان‌دهنده‌ی محتوای یک خانه ^۴ از نقشه‌ی بازی است. جدول زیر معنی هر کاراکتر را مشخص می‌کند.

عنوان	کاراکتر معادل	تصویر
آجر ساده	b	•
آجر شگفت انگیز دارای سکه	?	•
آجر شگفت انگیز دارای قارچ	m	•
آجر شگفت انگیز دارای قارچ سلامتی	h	•
بلوک معمولی	@	•
بلوک زمینی	#	•
ماريو	M	•
گومبا کوچولو	l	•
کوپا تروپا	k	•
لوله		•
پرچم	f	•

عکس‌های مربوط را می‌توانید در پوشه‌ی assets پیدا کنید. همچنین عکس پس‌زمینه نیز داخل همین پوشه قرار دارد که باید پشت تمامی تصاویر دیگر رسم شود.

توجه کنید که پیاده‌سازی شما نباید به یک نقشه‌ی خاص برای بازی وابسته باشد و باید بتواند با هر نقشه‌ی دلخواهی که مطابق فرمت گفته شده باشد، بدون نیاز به کامپایل مجدد، اجرا شود. به این منظور برنامه‌ی شما باید آدرس نقشه‌ی مرحله‌ی مورد نظر را از خط فرمان^۵ دریافت کند. در هنگام تحویل پروژه، برنامه‌ی شما با یک نقشه‌ی جدید که قبلاً ندیده‌اید آزموده خواهد شد.

۲.۲ دوربین

همان‌طور که احتمالاً متوجه شده‌اید، نقشه‌ی بازی از ناحیه‌ای که دوربین بازی می‌تواند نشان دهد، بسیار بزرگ‌تر است. در نتیجه نیاز است که با حرکت ماریو به سمت جلو، دوربین نیز او را دنبال بکند. یعنی وقتی ماریو به لبه‌ی راستی صفحه نزدیک می‌شود باید دوربین نیز کمی به راست برود تا ماریو از صفحه خارج نشود. اینکه مرز این جابه‌جایی دوربین چقدر باشد به عهده‌ی خود شما است و صرفاً طبیعی بودن آن کافیست.

⁴tile

⁵command line

توجه کنید که ماریو هیچ گاه نباید بتواند از لبه‌ی سمت چپ صفحه خارج شود، در نتیجه دوربین هیچ‌گاه نیاز نیست به سمت چپ حرکت کند.

نگران خارج شدن ماریو در راستای عمودی صفحه نباشید، نقشه‌هایی که در اختیار شما قرار می‌گیرند نیازی به جابه‌جایی دوربین در این راستا نخواهند داشت.

همچنین در نظر داشته باشید که عکس پس‌زمینه بازی هم باید حرکت کند. در واقع در هر لحظه باید بخشی از آن را نمایش دهید و با جابه‌جایی دوربین بخش بعدی عکس را به کاربر نمایش دهید تا حس حرکت در کاربر القا شود.

۳.۲ حرکات

۱.۳.۲ ماریو

حرکات ماریو در این بازی بسیار ساده هستند. ماریو باید با فشار دادن کلیدهای d و a به ترتیب به سمت راست و چپ حرکت کند، و هم‌چنین با فشار دادن کلید w بپرد. توجه کنید که ساختار جدول‌مانند نقشه، فقط به منظور راحتی ذخیره کردن و خواندن نقشه است و حرکات‌های داخل بازی ریزدانه‌تر از خانه‌های نقشه و در مقیاس پیکسل هستند. حرکت ماریو در راستای محور y ها همواره از فرمول‌های حرکت با شتاب ثابت پیروی می‌کند، یعنی:

$$\Delta y = v_y \Delta t$$

$$\Delta v_y = g \Delta t$$

و حرکت در راستای محور x ها نیز از فرمول حرکت با سرعت ثابت تبعیت می‌کند، یعنی:

$$\Delta x = v_x \Delta t$$

که در این فرمول‌ها v_x و v_y به ترتیب سرعت در راستای y و سرعت در راستای x بوده و g شتاب گرانشی است.

معمولاً فرمول‌های سقوط آزاد را به شکل $y = -\frac{1}{2}gt^2 + v_{0y}t + y_0$ می‌بینیم، که برای محاسبه‌ی مستقیم جابه‌جایی آسان‌تر است ولی استفاده از فرمول‌های بالا برای پیاده‌سازی کامپیوتری راحت‌تر هستند.

با توجه به این فرمول‌ها در واقع فشار دادن دکمه‌ی پرش باعث ایجاد یک سرعت اولیه در راستای y می‌شود. مشخص کردن مقدار دقیق ثابت g و مقدار v_x به عهده‌ی خود شما است و لازم و کافی است که این مقادیر طوری انتخاب شوند که حرکت ماریو طبیعی بوده و مشابه لینک داده شده در ابتدای تمرین باشد. طبیعتاً این فرمول‌ها فقط در حالتی برقرار هستند که مانعی در سر راه ماریو وجود نداشته باشد، چرا که در این صورت حرکت ماریو در آن راستا متوقف می‌شود.

۲.۳.۲ دشمنان

در این بازی از شما خواسته می‌شود که دو نوع دشمن را پیاده‌سازی کنید. توضیحات این دو نوع در بخش دشمنان آمده است. اما همه‌ی دشمنان ویژگی‌های مشترکی در حرکت خود دارند که در این بخش به آن‌ها می‌پردازیم.

دشمنان در این بازی در ابتدا در مکان اولیه‌ی خود ساکن هستند و در همین وضعیت می‌مانند تا وقتی که وارد کادر دوربین بازی شوند. پس از ورود به کادر دوربین دشمنان شروع به حرکت به سمت چپ کرده و تا وقتی مانعی جلوی آن‌ها نباشد به حرکت خود ادامه می‌دهند. هر وقت که دشمنی به مانعی در مسیر حرکت خود برخورد کند، جهت حرکت خود را تغییر می‌دهد (از راست به چپ، یا از چپ به راست).

برخلاف بازی اصلی، در این تمرین دشمنان برای هم مانع نیستند، یعنی دشمنان می‌توانند از یکدیگر عبور کنند. تنظیم سرعت حرکت دشمنان نیز، مانند ماریو، بر عهده‌ی خود شما است و کافی است که حرکت و سرعت دشمنان طبیعی و مشابه لینک داده شده در ابتدای تمرین باشد.

از شما انتظار می‌رود با استفاده از عکس‌های داده شده، حس اینکه ماریو و دشمنان در حرکت پاهای خود را تکان می‌دهند را به‌وجود بیاورید.

۴.۲ موانع

در این بازی سه مانع مختلف وجود دارد، که جلوی ورود ماریو را به خانه‌ای که داخل آن هستند می‌گیرند:

- **بلوک‌ها** ساده‌ترین موانع موجود در بازی. اندازه‌ی آن‌ها همیشه به اندازه‌ی یک خانه از نقشه‌ی بازی است. بلوک‌ها خود دو نوع معمولی و زمینی دارند، که تفاوت آن‌ها فقط در ظاهرشان است. این تفاوت را می‌توانید در جدول بخش نقشه‌ی بازی مشاهده کنید.
- **لوله‌ها** این موانع کمی از بلوک‌ها پیچیده‌تر هستند. ارتفاع این موانع می‌تواند تا حد دلخواه زیاد باشد، ولی عرض آن‌ها همواره برابر دو خانه از نقشه است. همچنین این موانع همواره عمودی هستند. شما باید از عکس‌های داده شده استفاده کنید، تا ظاهر لوله را مشابه لینک داده شده در ابتدای پروژه بسازید.
- **آجرها** این موانع، برخلاف دو مانع قبلی که روی زمین هستند، همواره از زمین فاصله دارند. وقتی ماریو از پایین به آجری ضربه می‌زند، آجر کمی به بالا پرتاب می‌شود و اگر دشمنی روی آن باشد، آن را از بین می‌برد، سپس به موقعیت اول خود برمی‌گردد. آجرها ۲ نوع مختلف به شرح زیر دارند:
- ◇ **معمولی** این آجرها از آجرهای شگفت‌انگیز ضعیف‌ترند. به این معنا که به هنگام ضربه زدن ماریو به آن‌ها از پایین، اگر ماریو بزرگ باشد، می‌شکنند. این آجرها هیچ‌وقت محتوایی ندارند.
- ◇ **شگفت‌انگیز** این آجرها که با علامت سوال مشخص شده‌اند، محکم هستند و هیچ‌گاه نمی‌شکنند. اما وقتی ماریو از پایین به آن‌ها ضربه می‌زند، محتوای آن‌ها بیرون می‌آید. پس از بیرون آمدن محتوایشان، این آجرها تغییر شکل داده و علامت سوالشان از بین می‌رود.

۵.۲ دشمنان

در این بازی دو نوع دشمن وجود دارد، که قبل از اینکه به ویژگی‌های هر یک بپردازیم، ابتدا ویژگی‌های مشترک دشمنان را بررسی میکنیم. همه‌ی دشمنان، همانطور که در بخش حرکت توضیح داده شد، وقتی که وارد کادر بازی می‌شوند، به طور خودکار شروع به حرکت به سمت چپ می‌کنند. وقتی ماریو به یکی از دشمنان برخورد کند، اگر این برخورد از بالای سر دشمن باشد، آن دشمن آسیب می‌بیند، ولی ماریو آسیبی نمی‌بیند و در عوض فقط کمی به بالا پرتاب می‌شود (مهم است که این به بالا پرتاب شدن را پیاده سازی کنید). اما اگر این برخورد از جهتی دیگر باشد، ماریو یک جان خود را از دست داده و به ابتدای مرحله برمی‌گردد. دشمنان این بازی عبارتند از:

- **گومبا کوچولو**^۶ گومباها ساده‌ترین دشمنان در این بازی هستند. این دشمنان به محض برخورد با ماریو از بالای سرشان از بین می‌روند. البته قبل از اینکه کامل از بازی حذف شوند، برای چند ثانیه بدن مرده‌ی آن‌ها روی زمین باقی می‌ماند و پس از مدت کوتاهی دیگر اثری از آن‌ها در بازی باقی نمی‌ماند.
- **کوپا تروپا**^۷ این دشمنان کمی پیچیده‌ترند. وقتی ماریو از بالا به کوپا تروپا برخورد می‌کند کوپا تروپا به صورت برعکس روی زمین قرار می‌گیرد و دیگر حرکت نخواهد کرد. با برخورد بعدی ماریو با کوپا تروپا از سمت بالا، این موجود در جهت مناسب پرتاب می‌شود. در این حالت، کوپا تروپا با هر چیزی که برخورد کند، آن را بلافاصله می‌کشد (کوپا تروپاهای دیگر نیز بلافاصله می‌میرند و برعکس روی زمین نمی‌افتند) و مشابه تمامی حرکت‌های دیگر، وقتی تروپا در این حالت به مانعی برخورد بکند، جهت حرکتش عوض می‌شود. و این حالت تا جایی ادامه دارد که کوپا تروپا از یک جای خالی در نقشه بیرون بیافتد و از نقشه خارج شود.

^۶Little Gomba

^۷Koopa Troopa

۶.۲ محتوای آجرها

همانطور که در بخش موانع گفته شد، آجرهای شگفت‌انگیز می‌توانند محتویاتی داشته باشند، که به هنگام ضربه زدن ماریو به آن‌ها، این محتوا از آن‌ها خارج می‌شود. این محتوا می‌تواند یکی از موارد زیر باشد:

○ **سکه** وقتی ماریو به آجر دارای سکه ضربه می‌زند، باید یک نشانه‌ی بصری برای اینکه سکه‌ای داخل آن آجر وجود داشت (مثلاً اینکه یک سکه از آجر خارج شود) ظاهر شود و سپس یکی به تعداد سکه‌های ماریو اضافه شود.

○ **قارچ قرمز** این قارچ‌ها همانطور که از اسمشان پیداست، قارچ‌های قرمزی هستند که ماریو با خوردن آن‌ها تغییر اندازه می‌دهد. این تغییر اندازه باعث می‌شود که ماریو بتواند:

۱. توانایی شکستن آجرها را داشته باشد.

۲. در برابر یک ضربه از دشمنان ایمن باشد و در صورت اصابت هرگونه ضربه تنها اندازه‌اش کوچک‌تر شود و از جان‌های او کم نشود. توجه کنید که بلافاصله پس از این برخورد، ماریو باید حدود یک ثانیه غیرقابل‌کشتن باشد. در طول این مدت ماریو و دشمنان هیچ تاثیری نمی‌توانند بر یکدیگر بگذارند.

○ **قارچ جان** ماریو با خوردن این قارچ یک جان به جان‌هایش اضافه می‌شود.

○ **گل آتش** ماریو با خوردن این گل توانایی پرتاب گلوله‌های آتشین در جهت محور x را به دست می‌آورد. این گلوله‌های آتش با اصابت به دشمنان آن‌ها را نابود می‌کند. پیاده‌سازی گل آتش امتیازی است.

۷.۲ سربرگ بازی

در طول بازی، باید اطلاعات مشخصی در بالای صفحه در قالب یک سربرگ^۸ نمایش داده شوند. این اطلاعات به شرح زیرند:

○ **سکه‌ها**^۹ تعداد سکه‌هایی که تا به آن لحظه توسط ماریو جمع‌آوری شده است.

○ **جان‌ها**^{۱۰} تعداد جان‌های باقی‌مانده‌ی ماریو. در ابتدا تعداد جان‌های ماریو ۳ است.

۸.۲ پایان بازی

هر بار که ماریو با یکی از دشمنان برخورد می‌کند، اگر از جان‌های ماریو باقی مانده باشد، باید ماریو به ابتدای مرحله بازگشته و یکی از جان‌های آن کم شود. به جز این دو مورد (جان و مکان ماریو)، تغییر دیگری نباید در وضعیت مرحله (از جمله خالی یا پر بودن آجرها، اینکه کدام دشمن‌ها مرده اند و غیره) داده شود.

اگر برای ماریو جانی باقی نمانده باشد، باید با نشان دادن پیغام You Lose بازی تمام شود.

اگر ماریو بتواند به پرچم انتهای مرحله برسد، بازیکن بازی را برده است یا نمایش پیغام You Win بازی خاتمه می‌یابد.

^۸header

^۹coins

^{۱۰}lives

۹.۲ نکات پایانی

- در بازی اصلی، اصطکاک ماریو با زمین به حدی نیست که وقتی کاربر دستش را از روی دکمه‌ی راست/چپ برداشت ماریو بلافاصله از حرکت بایستد. همچنین موقع شروع حرکت نیز سرعت ماریو به تدریج زیاد می‌شود تا به سقف خود برسد. در واقع ماریو در بازی اصلی روی سطح سر می‌خورد. پیاده‌سازی سُر بودن سطح زمین بازی را جذاب‌تر می‌کند و انجام آن اختیاری بوده و برای شما نمره‌ی امتیازی خواهد داشت. (در این حالت دیگر حرکت ماریو در راستای x از فرمول حرکت با سرعت ثابت تبعیت نمی‌کند.)
- اینکه آجرهای شگفت‌انگیز بین سه رنگ موجود به صورت چرخشی تغییر رنگ بدهند (مانند آنچه که در لینک نمونه هست) برای شما نمره‌ی امتیازی خواهد داشت.
- در صورت نیاز می‌توانید ساختار تمامی پرونده‌های^{۱۱} مورد نیاز برای بازی (از جمله تصاویر و پرونده‌های نقشه) را تغییر دهید و لزومی ندارد ساختار اولیه را حفظ کنید. توجه کنید که ممکن است کتابخانه‌ی RSDL به‌روزرسانی شود؛ پس سعی کنید تغییری در کتابخانه ایجاد نکنید.
- در داخل پوشه‌ی sound، موسیقی پس‌زمینه‌ی بازی اصلی به همراه تعدادی sound effect قرار گرفته است. شما می‌توانید از با استفاده از امکانات پخش موسیقی RSDL به بازی خود صدا نیز اضافه کنید. انجام این کار امتیازی است.

۳ نحوه‌ی تحویل

پرونده‌های مربوط به برنامه‌ی خود را در پوشه‌ای با نام A5-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۷۹۹۹ باشد، نام پوشه‌ی شما باید A1C-810197999.zip باشد.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و اجرا شود.
- برای ایجاد رابط کاربری گرافیکی^{۱۲} و تمامی افکت‌های صوتی و تصویری برنامه‌ی خود باید از کتابخانه‌ی SDL2 و کتابخانه‌ی RSDL استفاده کنید.
- شما باید پوشه‌ی کامل پروژه باشد شامل کتابخانه‌ی RSDL، تصاویر و – در صورت استفاده – پرونده‌های صوتی را بارگذاری کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

¹¹file

¹²GUI