

به نام خدا

و اینک آخرین پروژه درس برنامه‌نویسی پیشرفته! © امیدواریم از این ترم و درسی که ارائه شد، لذت برده باشید و مطالب مفیدی را یاد گرفته باشید. الان وقت این است که همه مهارت‌ها و توانایی‌هایی که در این درس تاکنون کسب کرده‌اید را به معرض نمایش بگذارید. پیش از تعریف پروژه، به نکات زیر توجه کنید:

۰. فایل مربوط به توضیحات نحوه ارسال تمرین‌ها را که در مدل قرار دارد، برای بار آخر مطالعه کنید.

۱. برای انجام این پروژه، در اولین فرصت هم‌گروهی مناسبی انتخاب کنید! این پروژه برای گروه‌های دوفره (و نه بیشتر!) تعریف شده است. هم‌گروهی شما می‌تواند هر یک از دانشجویان این درس باشد و محدودیت مشترک بودن گروه کارگاه وجود ندارد.

۲. تمامی فایل‌های مربوط به کد و مستندات را به صورت یک فایل آرشیو (zip != rar) که به قالب زیر نام‌گذاری شده باشد، بارگذاری نمایید. دقت کنید که این پروژه تحویل حضوری دارد.

StudentNumber1_StudentNumber2_LastName1_LastName2.zip

۹۰۳۱۰۶۶_۹۰۳۱۸۰۶_Edalat_Ahmadpanah.zip

۳. مستندسازی به کمک JavaDoc، کامنت‌گذاری و رعایت اصول کدنویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است.

۴. تحلیل و طراحی مناسب و نیز رعایت اصول شی‌گرایی الزامی است. در هنگام تحویل حضوری نمودارهای Use case و کلاس بررسی خواهند شد.

۵. استفاده از مخزن خصوصی (private repository) گیت‌لب (GitLab) برای کنترل نسخه‌های برنامه الزامی است. شما باید دسترسی خواندن مخزن مربوط به پروژه را در اختیار مدرس کارگاه خود قرار دهید. توضیحات بیشتر در مستند آموزش گیت موجود در مدل قرار دارد و می‌توانید از مدرس کارگاه خود کمک بگیرید. باید commit‌های مناسب و منطقی در نسخه‌های مختلف برنامه شما وجود داشته باشد.

۶. سوالات خود را درباره این پروژه می‌توانید از طریق فروم موجود در مدل بپرسید. مطالب گفته شده در آن فروم توسط مدرسان درس یا کارگاه، بخشی از تعریف پروژه محسوب می‌شود.

۷. هر دو نفر گروه باید به تمامی جزئیات تحلیل، طراحی و پیاده‌سازی تسلط کامل داشته باشند. علاوه بر این که در تحویل حضوری از هر دو نفر سوالاتی در این خصوص خواهد شد، تناسب commit‌های هر یک از افراد گروه در گیت‌لب نیز بررسی می‌شود. نمرات برای هر دو نفر به صورت یکسان در نظر گرفته می‌شود.

۸. همان‌طور که می‌دانید، با توجه به محدودیت پایان ترم تحصیلی، مهلت تحویل این پروژه به هیچ‌وجه و تحت هیچ شرایطی تمدید نمی‌شود. با همکاری مؤثر و برنامه‌ریزی زمانی مناسب حتماً

می‌توانید در مهلت تعیین‌شده این پروژه را انجام دهید. از آنجایی که زمان دقیق آخرین مهلت برای وارد کردن نمرات در پورتال آموزشی در حال حاضر مشخص نیست(!)، تاریخ بارگذاری و تحویل حضوری به شکل زیر خواهد بود. با توجه به اعلام تاریخ نهایی از طرف آموزش، تاریخ زیر ممکن است تغییر کند. ولی به صورت حدودی شما باید تا موعد مشخص شده، پروژه را انجام دهید.

مهلت تحویل (بارگذاری در مودل): تا دوشنبه ۷ بهمن ۱۳۹۸ ساعت ۷:۰۰ صبح

تحویل حضوری: روز ۷ بهمن ۱۳۹۸

۹. در قسمت‌های مختلف پروژه باید خطاهای مختلف بررسی شوند و در قبال آن رفتار مناسبی از برنامه دریافت شود. پس شما باید برای تمامی قسمت‌ها عملیات Exception Handling را برای استثناهای Checked و همچنین در مواقع لزوم Unchecked پیاده‌سازی کنید.

در ادامه تعریف پروژه پایانی آمده است. ابتدا تمام آن را با دقت و حوصله مطالعه کنید و سپس پروژه را آغاز کنید.

منتظر پروژه‌های کامل، جذاب و عالی شما هستیم! ☺

تعریف پروژه Super Mario Bros

هدف از این پروژه، پیاده‌سازی یک بازی مشابه بازی [Super Mario Bros](#) است. برای آشنایی با این بازی می‌توانید از این [لینک](#) استفاده کنید. همچنین، در کنار این مستند شبیه‌ساز اجرای بازی در محیط ویندوز و iOS قرار داده شده است. برای آشنایی با این بازی می‌توانید آن را هم به صورت آنلاین و هم آفلاین ببینید. برنامه‌ای که تحویل می‌دهید باید مشابه بازی Super Mario Bros باشد؛ منظور از مشابه بودن، شباهت در عملکرد و نحوه بازی است ولی لزومی ندارد که ظاهر بازی نیز دقیقاً مشابه بازی اصلی باشد (ظاهر بازی کاملاً به سلیقه دانشجو وابسته است!). بنابراین به همه دانشجویان توصیه می‌شود ابتدا به اندازه کافی Super Mario Bros را بازی کنند؛ و بعد پیاده‌سازی پروژه را شروع کنند! در زیر برخی جزئیات پروژه و تفاوت‌ها با بازی اصلی بیان شده است:

- در این بازی هدف کنترل کاراکتر اصلی بازی، یعنی ماریو است. او مأموریت دارد تا شاهزاده را از قلعه نجات دهد و در طی این عملیات باید با دشمنان مبارزه کند. کنترل بازی باید مشابه نسخه اصلی، پیاده‌سازی شود. یعنی با کلیدهای W، A و D حرکت کند و با دکمه space بتوان تیر شلیک کند. (راهنمایی: دکمه‌های A و D برای حرکت به سمت چپ و راست و دکمه W برای پرش است.)

کاراکتر اصلی بازی (ماریو):

- کاراکتر اصلی بازی مقداری جان به صورت گسسته دارد که ۳ واحد است. این سه واحد به شکل سه ستاره در گوشه سمت راست بالا نمایش داده می‌شود.
- مقداری امتیاز دارد که در بالای صفحه نمایش در گوشه سمت چپ صفحه، نمایش داده می‌شود.
- تعدادی تیر دارد که در پایین صفحه نمایش در گوشه سمت راست نمایش داده می‌شود.
- شماره مرحله فعلی نیز در بالای صفحه نمایش و در مرکز صفحه نمایش داده می‌شود.

دشمنان:

در بازی اصلی تعداد زیادی از دشمنان تعریف شده‌اند که هرکدام عملکرد متفاوتی دارد. اما به طور کلی می‌توان آنها را در دو دسته دشمنان متحرک و ثابت تعریف کرد. شما باید در هر مرحله از بازی حداقل ۴ نوع دشمن (۳ نوع متحرک و ۱ نوع ثابت) پیاده‌سازی کنید. برای مثال می‌توانید دشمن‌های زیر را پیاده‌سازی کنید. (لازم نیست که حتماً از دشمن‌های تعریف شده در زیر استفاده کنید. می‌توانید به سلیقه خود هر کدام از دشمن‌های موجود در بازی را پیاده‌سازی کنید. تنها پیاده‌سازی یکی از لاکپشت‌ها اجباری است.) گاهی اوقات (به صورت تصادفی) با نابودکردن دشمنان مقداری امتیاز به حساب ماریو اضافه می‌شود. دشمن‌های مختلف امتیازهای متفاوتی دارند.

لاکپشت سبز: دشمن متحرک است. روی سطح زمین حرکت می‌کند. چنانچه روی سطح بالاتر باشد بعد از پایان سطح، روی زمین می‌افتد و به حرکت خود ادامه می‌دهد. با برخورد لاکپشت به ماریو، یکی از جان‌های ماریو کم می‌شود.

لاکپشت قرمز: دشمن متحرک است. مانند لاکپشت‌های سبز هستند اما روی سطوح بالاتر وقتی به انتهای سطح می‌رسند، جهت حرکتشان تغییر می‌کند و روی همان سطح می‌ماند و با برخورد به ماریو یکی از جان‌های ماریو را کم می‌کند.

گیاه گوشتخوار: دشمن ثابت است. این گونه از گیاه‌ها در لوله‌های سبز رنگ زندگی می‌کنند و به صورت متناوب از لوله بیرون می‌آیند و بعد از چند ثانیه برمی‌گردند، اگر در لحظه‌ای که بیرون هستند ماریو به آن‌ها برخورد کند، یکی از جان‌های ماریو کم می‌شود. لوله‌های سبزرنگ می‌توانند در سقف نیز وجود داشته باشند، پس این گونه از گل‌ها نیز می‌توانند در آنها وجود داشته باشند.

جوجه تیغی: دشمن متحرک است. مانند لاکپشت‌های سبز هستند. با برخورد جوجه تیغی به ماریو، یکی از جان‌های ماریو کم می‌شود.

نحوه صدمه دیدن دشمنان:

لاکپشت‌ها: تمامی لاکپشت‌ها با دوبار پریدن ماریو روی سرشان می‌میرند به این صورت که با پرش اول، خودشان نابود می‌شوند و با پرش دوم لاک آنها از بین می‌رود. اگر ماریو بعد از یک بار پریدن روی لاکپشت، به لاک باقی‌مانده ضربه بزند، لاک در جهت ضربه روی سطح حرکت می‌کند. اگر به یک شی مثل لوله سبز برخورد کند، جهت حرکت آن برعکس می‌شود. ماریو می‌تواند با ضربه زدن به لاک و حرکت آن، دشمنان متحرک دیگر را از بین ببرد. اگر لاکپشت‌ها یک تیر بخورند، درجا از بین می‌روند.

گیاه گوشتخوار: این گیاه قابل مردن نیست.

جوجه تیغی: ماریو تنها با استفاده از تیر می‌تواند آن را از بین ببرد.

نقشه بازی

در نقشه بازی چندین المان وجود دارد که باید پیاده‌سازی شوند.

انواع دیوارها (یا طبقه‌های بازی): در سطح زمین یک دیوار غیر قابل تخریب وجود دارد که ماریو روی آن حرکت می‌کند. روی زمین ممکن است چاله وجود داشته باشد که با افتادن در آن باعث کم شدن یک جان ماریو می‌شود و بعد از آن بازی باید ماریو را در نقطه قبل از چاله ظاهر کند.

دیوارهای عمودی: این دیوارها ممکن است ارتفاعی داشته باشند و امکان پریدن از آنها بر اساس نقشه می‌تواند وجود داشته باشد.

دیوارهای (طبقه‌های) تخریب‌پذیر: گونه‌ای از دیوارهای تخریب‌پذیر هستند. این گونه از دیوارها با پرش‌های ماریو و برخورد سر ماریو به زیر دیوار (طبقه)، بلافاصله تخریب می‌شود. برخی از دیوارها هستند که با ۳ ضربه سر ماریو تخریب می‌شوند. در این گونه از دیوارها، با احتمال ۵۰ درصد سکه وجود دارد و با احتمال ۲۰ درصد قارچ وجود دارد. (یک دیوار می‌تواند هم قارچ داشته باشد و هم سکه اما سکه با هر ضربه سر ماریو به دیوار، به حساب ماریو وارد می‌شود اما قارچ تنها در هنگام خراب شدن کامل به ماریو داده می‌شود.)

نقشه بازی نباید به‌گونه‌ای باشد که کاربر نتواند بازی را ادامه دهد، مثلاً نباید یک چاله خیلی بزرگ بدون راه عبور یا یک دیوار بلند بدون امکان پرش از روی آن وجود داشته باشد. نقشه انتخابی شما باید به اندازه کافی بزرگ باشد که تمامی نمونه دشمن‌ها و المان‌های نقشه را بتوان در آنها پیاده‌سازی کرد. برای مثال ۱-۳ World و ۲-۲ World پیاده شوند. البته پیاده‌سازی بیشتر از دو نقشه نمره امتیازی خواهد داشت.

خوردن قارچ: با خوردن قارچ جان ماریو یک واحد افزایش پیدا می‌کند. در صورتی که جان ماریو کامل بود، ماریو بزرگ می‌شود! در این حالت اگر ماریو در چاله بیفتد یا به دشمنی برخورد کند، به جای اینکه یک واحد از جان آن کم بشود، صرفاً اندازه‌اش که بزرگ بود، کوچک می‌شود و دشمن نیز نابود می‌شود.

خوردن سکه: با خوردن سکه به امتیاز ماریو اضافه می‌شود.

پرتاب تیر: ماریو امکان پرتاب تیر با دکمه space روی کیبورد را دارد. ماریو با خوردن گل‌های آفتاب‌گردان که به صورت تصادفی در زمین بازی قرار داده می‌شوند، می‌تواند تیر شلیک کند. البته ماریو باید پیش از این قارچ خورده باشد و در حالت بزرگ باشد تا بتواند از تیرها استفاده کند. در غیر این صورت خوردن گل‌ها تنها به امتیاز او اضافه می‌کند. نحوه حرکت تیرها به صورت زیگزاگ است که بعد از ۴ بار برخورد به زمین، تیر محو می‌شود. اگر تیر از صفحه دوربین بازی نیز خارج شد، محو می‌شود. ماریو می‌تواند فقط ۳ تیر فعال شلیک شده داشته باشد و اگر بخواهد تیر بعدی را شلیک کند باید صبر کند تا یکی از تیرها محو شود.

مراحل بازی:

باید حداقل دو مرحله از بازی را طراحی کنید، هر مرحله یک نقشه است که شامل المان‌هایی است که در بالا توضیح داده شد. در پایان هر نقشه یک قلعه وجود دارد که با برخورد به آن قلعه ماریو وارد مرحله بعد می‌شود. در پایان هر مرحله، امتیاز بازیکن ذخیره شده و تمام امتیازهای قبلی آن نمایش داده می‌شود. در پایان مرحله آخر به جای قلعه یک شاهزاده منتظر نجات داده شدن است! اگر جان ماریو صفر شود Game Over شده و باید بازی را از اول شروع کند. بازی در حالت آفلاین باید قابلیت Pause / Resume داشته باشد.

- بازی را برای سه سطح ساده، متوسط و سخت طراحی کنید. سطوح دشواری بازی بر اساس تعداد و تنوع دشمنان تعیین می‌شود.

- گرافیک بازی اهمیت دارد! ظاهر بازی می‌بایست قابل قبول و کاربرپسند باشد. به طور خاص، نباید تصویر بازی پرش داشته باشد و اجرای بازی باید کاملاً روان باشد. بازی‌هایی که از گرافیک خیلی خوب (یعنی ظاهر زیبا و جذاب) برخوردار باشند، نمره بیشتری دریافت خواهند کرد. برای بهبود گرافیک حتماً از کدی که ضمیمه این مستند است استفاده کنید.

- بازی باید قابلیت دونفره شدن تحت شبکه را نیز داشته باشد. به این صورت که در منوی اولیه بازی، یک بازیکن در نقش سرور بازی را ایجاد کند و بازیکن دیگر به بازی متصل شود. قابلیت انجام بازی دو

نفره به این صورت که هر دو بازیکن با هم در یک نقشه قرار می‌گیرند و همزمان با هم بازی را شروع می‌کنند.

تصویر بازی برای هر بازیکن باید به گونه‌ای باشد که کاراکتر هر بازیکن در مرکز صفحه باشد و در صورتی که بازیکن رقیب نزدیک بود (در شعاع دید کاراکتر شما قرار داشت)، او نیز نمایش داده می‌شود. تیرهای بازیکنان برای یکدیگر قابل نمایش است اما به همدیگر آسیبی نمی‌رساند. چنانچه بازیکنی دشمنی را از بین ببرد، برای بازیکن رقیب نیز آن دشمن مرده است. عملاً تمامی اتفاقات برای هر دو بازیکن یکسان و قابل مشاهده خواهند بود. برنده، بازیکنی است که زودتر به پایان نقشه برسد. اگر هر یک از بازیکنان Game Over شود، بازی ادامه پیدا می‌کند تا بازیکن دیگر نیز Game Over شود و یا به شاهزاده برسد.

مواردی که پیاده‌سازی آنها نمره امتیازی دارد:

۱. اگر یکی از بازیکنان در حالت آنلاین جاننش تمام شد و Game Over شد، بتواند تصویر بازیکن حریف را تا زمانی که برنده یا Game Over شود مشاهده کند. (قابلیت spectator در بازی‌ها)

۲. امکان pause کردن بازی آنلاین با ارسال درخواست به کاربر حریف و قبول شدن درخواست pause وجود داشته باشد.

۳. امکان save بازی و load مجدد آن.

۴. صداگذاری برای بازی.

۵. ایجاد مدل‌های جدید دشمن برای بازی.

۶. ایجاد کد تقلب (Cheat Code) برای بازی.

۷. امکان طراحی نقشه بازی توسط کاربر. (Map Editor)

۸. پیاده‌سازی هوش مصنوعی برای دشمنان متحرک. برای مثال دشمنان ماریو را تعقیب کنند و تا زمانی که ماریو آنها را از بین نبرده است، آنها به تعقیب خود ادامه دهند. (می‌توانید از الگوی Observer استفاده کنید.)

۹. استفاده از Maven برای ساخت پروژه.

۱۰. نوشتن Unit Test برای بخشی از پروژه و اجرای آنها و نشان دادن درستی کد.

۱۱. استفاده از الگوهای طراحی مختلف در طراحی بازی.

برای بدست آوردن منابع بصری بازی و صداهای مربوط به آن می‌توانید به لینک‌های زیر مراجعه کنید:

لینک مربوط به منابع بصری:

<https://www.sprites-resource.com/nes/supermariobros/>

لینک مربوط به صداهای بازی:

<https://themushroomkingdom.net/media/smb/wav>