



برنامه‌نویسی پیشرفته

زمستان ۱۴۰۱ - بهار ۱۴۰۲

فاز دوم پروژه

تیم تعریف و طراحی پروژه:

امیر محمد شعبانی، آرین همتی

* آیدی‌ها روی اسامی لینک شده‌اند و کافیست روی آنها کلیک کنید

مقدمه‌ای بر فاز دو (حتماً بخوانید!)

در این فاز اکثر بخش گرافیک و لاجیک بازی ساخته خواهند شد و شما یک بازی آفلاین کامل که ترکیبی نامتعارف و نامتوافق از دارک‌سولز و سوپر ماریو است تولید خواهید کرد. اهداف آموزشی این فاز شامل گرافیک پیشرفته، (تشخیص برخورد و همپوشانی، پیاده‌سازی انواع حرکت و مدل‌سازی المان‌های گرافیکی و ...) تکمیل لاجیک بازی، مولتی‌تردینگ، کار با فایل و ... است. اما یکی از اهداف عملی و پیاده‌سازی که در این فاز مورد اهمیت بسیاری واقع می‌شود، کلینگ کدینگ، استفاده صحیح از پارادایم OOP و همچنین رعایت اصول SOLID است. در این فاز لایه‌های بسیاری از لاجیک و گرافیک که شما روی هم اضافه می‌شوند و با رعایت این اصول و انتخاب یک معماری مناسب، می‌توانید در عین راحتی کد زدن و جلوگیری از وقوع مشکلات مکرر و زمان‌بر در میانه پروژه، توسعه‌پذیری و تغییرپذیری آن را نیز حفظ کنید و در فاز سوم پروژه نیز زمان کمتری برای رفع مشکلات بالقوه در معماری لایه‌های کد خود صرف کنید. این در حالیست که رعایت نکردن این اصول می‌تواند منجر به تولید یک کد پیچیده و غیرقابل فهم و غیرقابل توسعه شود که مشکلات فراوانی در توسعه فاز سوم پروژه (شبکه، مولتی‌پلیر و ...) ایجاد خواهد کرد. فلذا اکیداً توصیه می‌شود بر این اصول مسلط باشید و قبل از دست به کد شدن، چندین بار داک را به طور کامل بخوانید و زمان کافی صرف کنید تا معماری‌ای قابل اطمینان و نسبتاً دقیق از لایه‌های کد خود طراحی کنید و بتوانید به راحتی هر قسمت از کد را تکمیل کنید. همچنین اکیداً توصیه می‌شود مدل‌های گرافیک خود را کاملاً از لاجیک جدا نگه دارید (می‌توانید Design Pattern‌هایی مثل MVC را به کار ببرید و یا هر معماری دلخواه دیگری استفاده کنید) و کلاس‌های مربوط به مدل‌های پایه خود (که در فرایند سیو باید در فایل ذخیره شوند) را طوری طراحی کنید که تا حد ممکن به Plain Old Java Object (POJO) نزدیک باشند. (تا در فرایند serialization به مشکلات بنیادی که نیاز به تغییر اساسی در کد دارند، بر نخورید)

همچنین کارگاه‌هایی برای آشنایی با مباحث گرافیکی مورد نیاز برگزار خواهد شد که زمان آن متعاقباً اطلاع داده می‌شود

مرور فاز اول

هدف از فاز اول پیاده‌سازی صفحه ورود بازی و بخش‌هایی جزئی از لاجیک و گرافیک بازی بوده است. در ادامه به پیاده‌سازی کامل لاجیک می‌پردازیم و در فاز بعدی به پیاده‌سازی بازی چندکاربره و چت با دوستان مشغول خواهید شد. در ادامه تمام جزئیات لاجیک و گرافیکی‌ای که باید پیاده‌سازی کنید را شرح می‌دهیم.

هر مرحله شامل چند بخش (section) است. هر مرحله شما باید حداقل شامل 5 بخش باشد. شما در کل باید 3 مرحله متفاوت طراحی کنید که در پایان داک توضیحات لازم در این باره داده شده است. از شما انتظار می‌رود که هر یک از مراحل شما در مجموع شامل تمام اجزای زیر باشند که هر یک از آنها در ادامه به تفصیل توضیح داده شده‌اند. همچنین برای هر یک از این اجزا، تمام انواع تعریف شده باید در هر یک از مراحل شما موجود باشند.

- بلوک‌ها
- آیتم‌ها
- لوله‌ها
- دشمن‌ها
- زمین خالی

2.....	مقدمه‌ای بر فاز دو (حتماً بخوانید!)
2.....	مرور فاز اول
4.....	محیط بازی (Game Env.)
4.....	اطلاعات بازی (HUI)
4.....	صدا
4.....	دکمه توقف
4.....	دکمه‌های بازی
5.....	حالتهای ماریو (Modes)
6.....	مرگ (Death Mechanics)
6.....	ریسک پیشروی (PR)
6.....	چکپوینت‌ها
6.....	مجازات مرگ
6.....	ناحیه خالی
7.....	آسیب دیدن ماریو
7.....	GameOver و اما
7.....	رجعت الأعداء (همون Respawn خودمون)
8.....	آیتم‌ها (Item Types)
9.....	بلوک‌ها (Block Types)
10.....	دشمن‌ها (Enemy Types)
11.....	لوله‌ها (Pipe Types)
12.....	کینگ کوبای (Bowser)
12.....	ساختار محیط باس‌فایت
12.....	تریگر شدن باس‌فایت
12.....	هیبت‌باکس و جان باس
12.....	محیط باس‌فایت
13.....	مکانیک‌های باس
13.....	بارگشت به سنت‌ها
13.....	انک‌های باس
14.....	عملکردها و واکنش‌های باس
15.....	بارگذاری بازی
15.....	الگوی فایل (JSON Object)
15.....	الگوی مرحله
15.....	الگوی بخش
16.....	الگوی لوله
16.....	الگوی بلوک
16.....	الگوی دشمن
17.....	پیاده‌سازی مراحل بازی
17.....	مرحله اجباری
17.....	نکات پایان
17.....	مخنومه‌ای بر Game Design
18.....	باز هم بخش امتیازی! (تمامی قسمت‌های توضیح داده شده در این بخش امتیازی هستند)
18.....	استفاده از ابزار گیت
18.....	استفاده از Build Tool (میون و گریدل)
18.....	استفاده از ابزار JLINK
18.....	استفاده از روش‌های مبتنی بر PCA

محیط بازی (Game Env.)

اطلاعات بازی (HUI)

شما باید در مکانی از صفحه (عموماً بالای صفحه یا گوشه‌های پایینی صفحه) اطلاعات بازی را به کاربر نشان دهید. همچنین این اطلاعات می‌توانند مخفی باشند و با زدن دکمه ای خاص، به مدت محدودی (مثلًا 3 ثانیه) در محل تعریف شده نمایش داده شوند و دوباره مخفی شوند. (تصمیم برای نحوه پیاده‌سازی به عهده خودتان خواهد بود) این اطلاعات باید شامل موارد زیر باشند:



- امتیاز
- تعداد سکه‌های کسب شده
- مرحله بازی
- زمان باقی‌مانده
- تعداد قلب‌های باقی‌مانده

تصویر تزئینی است

صدا

بازی شما باید دارای صدای پس زمینه باشد. می‌توانید از همان صدای بازی ماریو استفاده کنید. همچنین قسمت‌های پایان مرحله، پایان بازی، آسیب دیدن ماریو، کشتن یک دشمن و باس فایت باید دارای صدای دشمنها و آهنگ‌های پس زمینه منحصر به فرد باشند.

دکمه توقف

کاربر باید بتواند با فشردن دکمه Esc بازی را متوقف کند. پس از توقف بازی، کاربر باید قادر باشد صدای بازی را قطع / وصل کند، بازی را ادامه دهد و یا از بازی خارج شود. در صورتی که تصمیم به خروج از بازی بگیرد، بازی باید آخرین وضعیت سیو شده بازی را ذخیره کند. کاربر با وارد شدن به نسخه ذخیره شده، باید قادر باشد از همان وضعیت ذخیر شده بازی را ادامه دهد.

دکمه‌های بازی

در بازی باید شما دکمه‌هایی برای اجرا شدن یک سری اتفاقات پیاده‌سازی کنید. انتخاب دکمه مورد نظر به انتخاب خودتان است.
(تعاریف مینی ماریو، مگا ماریو و ماریوی آتشین در قسمت بعد بررسی شده است)

- **حرکت افقی:** کاربر باید قادر باشد با فشردن دکمه‌های مشخصی، ماریو را به چپ یا راست حرکت دهد.

نشستن: کاربر باید بتواند با فشردن دکمه مربوطه، کاری کند که ماریو بنشیند و ارتفاع خود را کم کند. این حرکت برای مینی ماریو امکان پذیر نیست اما اگر ماریو در حالات مگا ماریو و ماریوی آتشین باشد، ارتفاعش به یک بلوک کاهش می‌یابد.

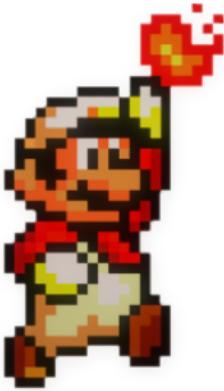
- **پرش:** کاربر باید بتواند با فشردن دکمه مربوط به پرش، بپرد. ارتفاع پرش ماریو به اندازه 4 بلوک است.

پرتاب تیر: اگر ماریو قابلیت پرتاب تیر داشته باشد، باید با فشردن دکمه مربوطه، یک تیر پرتاب کند. تیرهای ماریو در دو ارتفاع مختلف می‌توانند پرتاب شوند. در صورتی که ارتفاع ماریو در لحظه پرتاب تیر یک بلوک باشد، تیر باید از ارتفاع 0.5 بلوک پرتاب شود و در صورتی که ارتفاع ماریو دو بلوک باشد، تیر از ارتفاع 1.5 بلوک پرتاب شود. (دقت کنید ماریو نمی‌تواند در حالی که روی سطح بلوکی قرار ندارد، تیری پرتاب کند) تیر در راستای افقی و از نقطه پرتاب شده و در جهت ایستادن ماریو با سرعتی برابر با سرعت ماریو حرکت می‌کند. برد تیر ماریو 8 بلوک است. در صورتی که تیر ماریو 8 بلوک را طی کند و یا قبل از اینکه این اتفاق بیفتد، به یک بلوک و یا دشمن اصابت کند، از بین می‌رود. در صورت اصابت تیر به دشمن، 1 واحد از جان آن دشمن کاسته خواهد شد. (که به استثنای کینگ کوپا، باعث مرگ دیگر دشمنان می‌شود) همچنین پرتاب تیر به اندازه 3 ثانیه cooldown خواهد داشت.

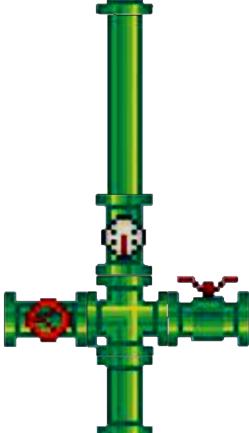
"**اگر دکمه‌های مربوط به پرش و نشستن همزمان فشرده شده باشند، ماریو نباید بپرد یا بنشیند**"

حالت‌های ماریو (Modes)

در ابتدای بازی ماریو در حالت مینی ماریو قرار دارد اما با خوردن برخی آیتم‌ها می‌تواند حالت خود را ارتقا دهد و با آسیب دیدن حالتش افت می‌کند. اگر در پایین ترین حالت خود (مینی ماریو) باشد، با ضربه بعدی یک قلب از دست می‌دهد و به آخرین نقطه سیو شده باز می‌گردد. همچنین اگر ماریو در بالاترین حالت خود باشد، با خوردن آیتم‌ها هیچ تغییری در وضعیتش ایجاد نمی‌شود.

ماریو آتشین (تیرانداز)	مگا ماریو (ماریو معمولی)	مینی ماریو (ماریو کوچک)
		
<ul style="list-style-type: none"> - مانند مگا ماریو توانایی از بین بردن بلوک‌های ساده را دارد و ارتفاع آن به اندازه دو بلوک است - با آسیب دیدن به مگا ماریو بدل می‌شود و قلبی از دست نمی‌دهد - قابلیت پرتاب تیر دارد 	<ul style="list-style-type: none"> - قوی تر از مینی ماریو است و توانایی شکاندن بلوک‌های ساده را دارد. ارتفاع آن به اندازه دو بلوک است - با آسیب دیدن به مینی ماریو تبدیل می‌شود و قلبی از دست نمی‌دهد 	<ul style="list-style-type: none"> - ضعیف است و توانایی از بین بردن بلوک‌های ساده را ندارد. ارتفاع آن به اندازه یک بلوک است - با یک بار آسیب دیدن، قلب از دست می‌دهد و به آخرین نقطه سیو شده باز می‌گردد و بازی را ادامه می‌دهد. در صورتی که قلبی باقی نماند باشد، می‌میرد

توجه کنید که ماریو در هر حالتی که باشد باید بتواند به اندازه‌ای بپرد که بالای بلوک‌ها و لوله‌ها برسد

	شمشیر ماریو :
	<p>اگر کاربر به مدت دو ثانیه، دکمه‌های مربوط به پرش و نشستن را با هم فشار دهد، در صورت داشتن سکه‌های کافی، 3 سکه از کاربر کاسته می‌شود و یک شمشیر که طول آن برابر با دو بلوک است، در دست ماریو ظاهر می‌شود. (از آنجا که ماریو لوله‌کش است، شمشیر لوله‌ای برای او طراحی شده است  اما شما می‌توانید از هر مدل گرافیکی‌ای که می‌خواهید استفاده کنید) سپس شمشیر به صورت افقی در ارتفاع یک بلوک باقی می‌ماند و با سرعت دو بلوک بر ثانیه به اندازه حداقل چهار بلوک به جلو پرتاب می‌شود. (در صورت برخورد هر دشمنی به جز Bowser، آن را از بین می‌برد و در صورت برخورد به Bowser نیز 3 واحد آسیب به او وارد می‌شود) در صورتی که قبل از رسیدن به فاصله 4 بلوک، به یک بلوک یا هر یک از دشمنان (اعم از کینگ کوپا) برخورد کند، شمشیر از بین می‌رود و در غیر این صورت، به همان صورت بر من‌گردد تا به دست ماریو برسد و سپس از بین می‌رود. همچنین استفاده از آن به مدت 5 ثانیه cooldown دارد. یعنی پس از نابود شدن شمشیر، کاربر تا 5 ثانیه دیگر نمی‌تواند مجدداً از این شمشیر استفاده کند.</p>

مرگ (Death Mechanics)

یکی از مهم ترین اجزای تعیین کننده در ساختار مکانیک هر بازی، مکانیک های مربوط به مرگ است. در این بخش، آسیب دیدن ماریو و مکانیک های مربوط به مرگ و NFL (Negative Feedback Loop) را توضیح خواهیم داد:

ریسک پیشروی (PR)

در هر نقطه از یک مرحله، نرخ پیشروی (progress rate) را تعریف می کنیم که برابر با $\frac{\text{چابجایی طولی ماریو از ابتدای مرحله تا نقطه فعلی}}{\text{چابجایی طولی موردنیاز برای رسیدن از ابتدا به انتهای مرحله}}$ است. در هر نقطه از مرحله، "ریسک پیشروی" را برابر با $| \text{progress rate} \times \text{coins} |$ تعریف می کنیم.

چکپوینت ها

در این بازی مثل بسیاری از بازی های دیگر، می توان پیشروی کاربر را در نقاطی به غیر از پایان مراحل ذخیره کرد. برای این کار، باید در بازی دروازه های چکپوینت طراحی کنید. هر مرحله شما باید یک یا دو دروازه چکپوینت داشته باشد. در صورتی که ماریو روی این دروازه ها بپردازد، یک پنل برای کاربر باز می شود و دو گزینه را پیش روی او قرار می دهد.



- کاربر می تواند با پرداخت PR تا سکه، پیشروی و زمان مرحله خود را در این نقطه سیو کند. در این صورت، پنل بسته شده و دروازه سر جای خود باقی می ماند. در این حالت اگر ماریو بمیرد یا کاربر بازی را مجدداً لود کند، این مرحله از این نقطه ادامه خواهد یافت. (مگر اینکه کاربر موفق شود این مرحله را به پایان برساند، که در این صورت مرحله از ابتدا آغاز خواهد شد)

- کاربر می تواند با رد کردن این پیشنهاد، به اندازه 25 درصد PR تا سکه دریافت کند. در این صورت پنل بسته شده و دروازه چکپوینت نابود می شود و تا زمانی که کاربر مرحله را به پایان نرساند (که باعث می شود این مرحله از ابتدا آغاز شود)، با مردن ماریو و یا لود دوباره بازی، این دروازه دیگر وجود نخواهد داشت و کاربر از آخرین نقطه سیو شده، به بازی ادامه خواهد داد.

مجازات مرگ

اگر کاربر در مرحله جاری بازی را در n چکپوینت سیو کرده باشد، هنگام مرگ علاوه بر توضیحاتی که در ادامه داد آمده است، مقدار $D(n) = \frac{(n+1) \times \text{coins} + PR}{n+4}$ سکه نیز از دست می دهد که در آن coins تعداد سکه های کاربر در لحظه مرگ است.

فراموش نکنید که هر بار که ماریو به هر دلیلی "بمیرد"، باید این بند اجرا شود

ناحیه خالی

در بعضی بخش ها ناحیه خالی ای وجود دارد که اگر ماریو در آن بیفتند:



- 30 امتیاز از کاربر کسر می شود. (امتیاز نمی تواند از صفر کمتر شود)
- یک قلب از دست می دهد. (تعداد قلبهای نمی تواند از صفر کمتر شود)
- اگر کاربر بازی را در یک چکپوینت سیو کرده باشد، به آخرین نقطه سیو شده برگشته و زمان بازی هم به آخرین زمان سیو شده تغییر می کند.
- اگر بازی در هیچ چکپوینتی سیو نشده باشد، ماریو به ابتدای مرحله برگشته و زمان مرحله ریست می شود و مجدداً ادامه پیدا می کند.
- حالت ماریو به مینی ماریو تغییر می کند.

در این قسمت به طور دقیق آسیب دیدن ماریو را تعریف می‌کنیم. برای این کار، ابتدا نیاز است که تعدادی تعریف ارائه دهیم:

- ضربه:** به معنای آسیب زدن به یک دشمن از روش‌های معتبر و تعریف شده در ادامه داک است.
- برخورد:** به منزله تماس با هر یک از دشمن‌های تعریف شده در بازی (به استثنای کینگ کوپا) است، به گونه‌ای که ضربه تلقی نشود.
- آسیب:** به معنای هرگونه برخورد با دشمن، قرار گرفتن در معرض حملات کینگ کوپا و یا قرار داشتن در ناحیه انفجار یک بمب است.

اگر ماریو ضد ضربه نشده باشد، با آسیب دیدن باید اعمال زیر را انجام دهید: (توضیح ضد ضربه بودن در ادامه ذکر شده است)

- 20 امتیاز از کاربر کسر می‌شود. (امتیاز نمی‌تواند از صفر کمتر شود)
- یک قلب از دست می‌دهد. (تعداد قلوب‌ها نمی‌تواند از صفر کمتر شود)
- اگر بازی در یک چک‌پوینت سیو شده باشد، ماریو به آخرین نقطه سیو برگشته و زمان هم به آخرین زمان سیو شده تغییر می‌کند.
- اگر بازی در چک‌پوینت سیو نشده باشد، ماریو به ابتدای سکشن فعلی برگشته و زمان ریست می‌شود و مجدداً ادامه پیدا می‌کند.
- حالت ماریو به مینی ماریو تغییر می‌کند.

GameOver و اما

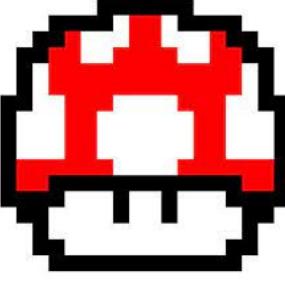
اگر ماریو در حالتی که هیچ قلب باقی‌مانده‌ای ندارد یا آسیب ببیند یا درون ناحیه خالی بیفت، باید مرحله تمام‌ریست شود و از ابتدا آغاز شود. در این حالت زمان ریست می‌شود، قلوب‌ها به تعداد اولیه برگشته و مرحله مجدداً لود می‌شود.



رجعت الأعداء (همون Respawn خودمون)

در نهایت دقت کنید تا وقتی که ماریو زنده است، دشمنان بعد از کشته شدن دیگر هیچوقت زنده نخواهند شد. به این معنا که اگر ماریو در نقطه‌ای آسیب ببیند و یا به ناحیه خالی بیفتند اما نمیرد، دشمنان دوباره زنده نخواهند شد. این مکانیک باعث می‌شود کاربر نتواند از زنده شدن دشمنان سود کند و اصطلاحاً جلوی Farm کردن کاربر را می‌گیرد.

آیتم‌ها (Item Types)

ستاره جادویی	قارچ جادویی	گل جادویی	سکه
			
<ul style="list-style-type: none"> - همانند قارچ جادویی 3 ثانیه در بالای بلوک قرار می‌گیرد و سپس به طرف راست حرکت می‌کند و رفتارش با بلوک‌ها نیز مانند قارچ جادویی است - ستاره جادویی علاوه بر حرکت افقی، بعد از هر یک ثانیه که روی زمین قرار گرفته باشد، به اندازه نصف پرش ماریو به طرف بالا می‌پرد - با خوردن آن 40 امتیاز کسب می‌کنید و یک مرحله حالت ماریو ارتقا پیدا می‌کند - همچنین برای 15 ثانیه ماریو ضد ضربه می‌شود و هر دشمنی که با ماریو برخورد کند می‌میرد. (به استثنای غول مرحله) 	<ul style="list-style-type: none"> - پس از ضربه بلوک در بالای آن قرار می‌گیرد و پس از 3 ثانیه به سمت راست با سرعتی نصف سرعت ماریو حرکت می‌کند - در طول حرکت، اگر در زیر آن بلوک دیگری نباشد باید به پایین سقوط کند و دوباره به راه خود ادامه دهد - اگر در طول حرکت به بلوکی برخورد کند باید جهت حرکت خود را تغییر دهد - همچنین اگر در طول حرکت، این آیتم از بخش خارج شود بازیکن آیتم را از دست می‌دهد - با خوردن آن 30 امتیاز کسب می‌کنید. یک مرحله حالت ماریو ارتقا پیدا می‌کند 	<ul style="list-style-type: none"> - این گل پس از ضربه به بلوک، در بالای بلوک قرار می‌گیرد و ثابت می‌ماند - با خوردن آن 20 امتیاز کسب می‌کنید و یک مرحله حالت ماریو ارتقا پیدا می‌کند 	<ul style="list-style-type: none"> - با گرفتن آن 10 امتیاز دریافت خواهید کرد - این آیتم اثری بر وضعیت ماریو ندارد

	<p style="color: blue;">بخش امتیازی : Electric Shield :</p> <p>اثر ضد ضربه شدن ستاره با یک حلقه (shield) دور ماریو مشخص شود که با برخورد دشمن (در زمان 15 ثانیه‌ای) منفجر شده و دشمن را از بین ببرد. (مشابه شکل رویرو) همچنین باید صدای انفجار را نیز در این لحظه پخش کنید.</p>
---	--

بلوک‌ها (Block Types)

بلوک‌ها یکی از اجزای اصلی سازنده محیط بخش‌های بازی هستند. این بلوک‌ها باید به نوعی در هوا قرار بگیرند که ماریو بتواند به آن‌ها ضربه بزند یا به بالای آن‌ها برود. در ادامه انواع متفاوت بلوک‌هایی که وجود دارد را معرفی می‌کنیم.

پیاده‌سازی 5 بلوک اول اجباریست و بالطبع مرحله‌های شما نیز باید دارای این بلوک‌ها باشند. پیاده‌سازی بلوک آخر امتیازیست.

- "ضربه" به معنای پریدن و برخورد کردن به پایین بلوک است.

بلوک جایزه‌دار	بلوک پر از سکه	بلوک سکه دار	بلوک توحالی	بلوک ساده
<ul style="list-style-type: none"> - با ضربه زدن به آن یک آیتم به صورت شناسی بالای آن قرار می‌گیرد و یک امتیاز به خاطر ضربه به آن داده می‌شود. سپس این بلوک تبدیل به بلوک توحالی می‌شود - احتمال ظاهر شدن آیتم‌های مختلف بعد از ضربه زدن به این بلوک باید با هم متفاوت باشند 	<ul style="list-style-type: none"> - مانند بلوک سکه دار است اما با هر ضربه سکه آن کسب می‌شود و نیازی به رفتن به بالای بلوک نیست - می‌توان تا 5 بار به این بلوک ضربه زد و هر بار یک سکه از آن دریافت کرد. بعد از 5 ضربه به بلوک توحالی تبدیل می‌شود 	<ul style="list-style-type: none"> - ماریو با اولین ضربه مانند بلوک ساده یک امتیاز کسب می‌کند - این بلوک با ضربه از بین نمی‌رود بلکه در بالای آن یک سکه قرار می‌گیرد و به بلوک ساده تبدیل می‌شود 	<ul style="list-style-type: none"> - با ضربه زدن به آن هیچ اتفاقی نمی‌افتد - توجه کنید این بلوک از بین نمی‌رود 	<ul style="list-style-type: none"> - این بلوک با یک ضربه نابود می‌شود و از بین می‌رود - با نابود شدن آن یک امتیاز به کاربر داده می‌شود

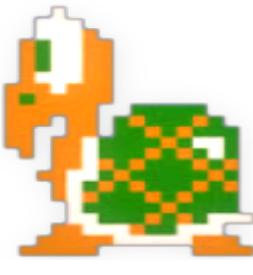
- توجه کنید باید برای سکه‌ها و آیتم‌های دیگر بازی، **گرانش** تعریف شود. برای مثال وقتی یک ضربه به بلوک سکه دار زده شود یک سکه در بالای آن قرار می‌گیرد و تبدیل به بلوک ساده می‌شود و با ضربه دوم به آن، بلوک ساده از بین می‌رود و در صورتی که سکه همچنان بالای آن قرار داشته باشد، باید به پایین بیفتد تا یا از دست برود و یا روی بلوک دیگری قرار گیرد.

	بخش امتیازی : Slime Block
	<ul style="list-style-type: none"> - با ضربه زدن به این آیتم هیچ اتفاقی نمی‌افتد. - ماریو می‌تواند با پرش روی این بلوک، 1.5 برابر ارتفاع پرش عادی خود به بالا پرتاپ شود. در صورتی که مجدداً روی یک بلوک اسلامی فرود بیاید، همچنان به همان میزان (1.5 برابر ارتفاع پرش عادی) به بالا پرتاپ خواهد شد و بیشینه افزایش ارتفاع ماریو تغییری نخواهد کرد.

دشمن‌ها (Enemy Types)

این موجودات با ماریو مقابله می‌کنند تا کار او برای رسیدن به پایان مرحله را سخت کنند. در هر مرحله باید تمام آنها موجود باشند.

- توجه کنید که اثر **گرانش** روی تمام این چهار دشمن در بازی را پیاده‌سازی کنید، به این معنا که اگر به هر دلیل، یک دشمن در مکانی از صفحه قرار بگیرد که روی هیچ بلوک نباشد، باید به پایین بیفتد تا به اولین سطح ثابت برسد و یا از بین برود.
- **مکانیک حرکت** تمام دشمن‌های غیر ساکن باید به گونه ای طراحی شود که وقتی روی بلوک در مسیر حرکت خود، جهت حرکت خود را تغییر دهند.
- "ضربه زدن" به معنای فرود آمدن بالای سر آن یا شلیک تیر است. ارتفاع هر یک از این دشمنان به اندازه یک بلوک است.
- "برخورد" به معنای تماس با دشمن است (به طوری که ضربه تلقی نشود) و در صورت برخورد، ماریو آسیب می‌بیند.
- گومپا، کوپا و اسپینی دشمنان متحرک هستند و همواره باید با سرعت ثابت 0.5 بلوک بر ثانیه در حرکت باشند.
- با کشتن هر دشمن (به جز کینگ کوپا) 3 سکه دریافت خواهد کرد.

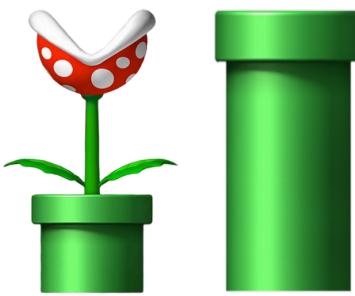
گل آدم‌خوار	جوچه تیغی (اسپینی)	لاکپشت (کوپا)	دشمن ساده (گومپا)
			
<ul style="list-style-type: none"> - این دشمن حرکت نمی‌کند و تنها در لوله‌ها یافت می‌شود - با یک ضربه کشته می‌شود و تنها راه آسیب رساندن به آن استفاده از تیر یا شمشیر است و با کشتن آن 1 امتیاز کسب می‌کنید 	<ul style="list-style-type: none"> - این دشمن مانند گومپا حرکت می‌کند - اگر ماریو در فاصله کمتر از 4 بلوک از اسپینی قرار بگیرد، اسپینی با شتاب 0.5 بلوک بر مจذور ثانیه به سوی ماریو می‌دود و تا وقتی که با ماریو همسطح است و فاصله اش با ماریو از 4 بلوک بیشتر نشده، او را دنبال خواهد کرد. دقت کنید حتماً باید این حرکت را به صورت شتاب دار پیاده کنید - هرگونه تماسی (اعم از فرود روی آن) برخورد تلقی می‌شود و تنها راه کشتن آن، ضربه زدن با تیر و یا استفاده از شمشیر است. در نهایت با کشتن اسپینی 3 امتیاز دریافت می‌کنید 	<ul style="list-style-type: none"> - کوپا مانند گومپا حرکت می‌کند - با ضربه زدن به آن مقداری به چپ یا راست پرتاپ می‌شود و برای 3 ثانیه دیگر حرکت نمی‌کند و با ضربه دوم می‌میرد. در صورتی که در این 3 ثانیه کشته نشود، مانند حالت اولیه‌اش عمل می‌کند - با کشتن آن 2 امتیاز می‌گیرید 	<ul style="list-style-type: none"> - این دشمن روی سطح صاف حرکت می‌کند و به چپ و راست می‌رود - با یک ضربه می‌میرد - با کشتن آن 1 امتیاز کسب خواهد کرد

بخش امتیازی :

	<p>- این دشمن در آسمان و بالاتر از ماریو به طور ثابت پرواز می‌کند. این دشمن باید در مکانی قرار بگیرد که ماریو بتواند با ایستادن روی بلوک‌های موجود در مرحله، آن را با تیر بزند. ماریو به هیچ طریقی جز استفاده از تیر نمی‌تواند به این دشمن آسیب بزند.</p> <p>- هر 5 ثانیه یک بار، یک بمب به سمت زمین پرتاپ می‌کند. این بمب بعد از برخورد به اولین سطح، منفجر شده و به هر موجودی (اعم از ماریو یا دشمن‌ها) در شعاع یک بلوک از نقطه انفجار، آسیب می‌زند.</p> <p>- با دو ضربه می‌میرد، اما هرگونه برخورد مستقیم ماریو با آن، برخورد تلقی می‌شود. همچنین در صورت کشتن آن 3 امتیاز دریافت خواهد کرد.</p>
---	---

(Pipe Types) لوله‌ها

لوله‌ها مانند مواعی هستند که ماریو می‌تواند از بالای آنها رد شود. لوله‌ها می‌توانند ارتفاع‌های متفاوتی داشته باشند اما باید دقیق کنید که طوری طراحی شوند که ماریو بتواند به بالای آنها برسد. در ادامه انواع لوله‌ها را معرفی می‌کنیم. شما باید همه آنها را در مرحله خود پیاده‌سازی کنید.

لوله مخفی	لوله با گل	لوله ساده
		
<ul style="list-style-type: none"> - این لوله به شکل لوله ساده یا لوله با گل است. اگر ماریو در بالای آن قرار بگیرد و بنشیند (و آسیب نبیند) به بخش مخفی می‌رود. - بخش مخفی یک بخش تنها است که چپ و راست آن هیچ بخش دیگری وجود ندارد. در این بخش باید یک لوله مخفی دیگر وجود داشته باشد که ماریو را به همان لوله باز گرداند. در این بخش می‌توانید به سلیقه خود پر از دشمن یا پر از بلوک‌های سکه دار کنید. 	<ul style="list-style-type: none"> - مانند لوله ساده است اما هر 4 ثانیه یک بار دشمن "گل آدمخوار" از آن بیرون می‌آید - به مدت 3 ثانیه این دشمن در بالای آن قرار خواهد داشت و سپس به داخل لوله باز می‌گردد و تا مدت زمان 4 ثانیه دیگر بیرون نمی‌آید. - اگر ماریو گل آدمخوار را بکشد دیگر از آن دشمنی بیرون نمی‌آید. 	<ul style="list-style-type: none"> - این لوله یک مانع ساده است که ماریو می‌تواند از بالای آن رد شود. - در صورتی که ماریو بالای آن قرار بگیرد و یا حرکت خاصی انجام دهد، اتفاقی نمی‌افتد.

	بخش امتیازی : Pipe of Deceit
	<p>- این لوله ظاهری شبیه به لوله ساده دارد و باعث می‌شود کاربر طمع کند و روی آن بنشیند تا وارد بخش مخفی شود اما بجای ورود به بخش مخفی، یک بلوک توخالی در ارتفاع 4 بلوک از لوله ظاهر می‌شود و سپس یک شراره آتش به طول 4 بلوک، از زاویه 45 درجه (پادساعت‌گرد) نسبت به خط واصل لوله به بلوک ظاهر شده بالای آن (حرکت 1 در تصویر) شروع به حرکت کرده و سپس با حرکت آونگ (بدون در نظر گرفتن مقاومت هوا) شروع به تاب خوردن می‌کند (متناوباً حرکات 1,2 نشان داده شده در تصویر تکرار خواهد شد) طبعاً اگر این آتش به ماریو برخورد کند ماریو آسیب خواهد دید و شما نیز باید مطابق مکانیک‌های آسیب عمل کنید. دقیق کنید که باید حرکت این المان را مطابق حرکت آونگ در فیزیک مدل و پیاده‌سازی کنید. همچنین توجه کنید این المان تنها در صورت ریست شدن کل مرحله از بین می‌رود و در غیر این صورت کماکان در همان نقطه از زمین بازی باقی می‌ماند و به حرکت ادامه می‌دهد.</p>

کینگ کوپا (Bowser)



این دشمن در پایان مرحله سوم بازی ظاهر می‌شود و آخرین مانع ماریو برای آزاد کردن (انشالله اگر صیغه محرومیت خوانده باشند، اینمیشن به هم رسیدن آنها را هم می‌توانید پیاده کنید) در ادامه نحوه پیاده‌سازی و رفتارهای این باس بازی توضیح داده شده‌اند:

ساختار محیط باس‌فایت

تریگر شدن باس‌فایت

به محض قرار گرفتن ماریو در فاصله مناسب، دو دیوار از بلوک‌های تو خالی دو طرف صفحه را پر می‌کنند تا ماریو نتواند از باس فایت فرار کند. (فراموش نکنید که در صورت آسیب دیدن ماریو و ادامه یافتن شدن مجدد بازی از ابتدای مرحله یا یک چکپوینت، این دیوارها را مجدداً از بین ببرید تا ماریو دوباره بتواند باس‌فایت را تریگر کند. همچنین در این صورت، HP کینگ کوپا به اندازه حداقل 25 درصد از وضعیت آخر خود بیشتر می‌شود) سپس یک HP Bar در پایین یا بالای صفحه ایجاد کنید که نشان‌دهنده میزان HP باقی‌مانده Bowser است. همچنین در این لحظه آهنگ مخصوص باس‌فایت پخش شده و حملات Bowser آغاز خواهند شد.

هیت‌باکس و جان باس

برای سادگی کار، کافیست هیت‌باکس این باس را صرفاً یک ناحیه مستطیلی از آن در نظر بگیرید و صرفاً ضربه به سر (ناحیه بالای هیت‌باکس) را از بقیه نواحی تشخیص دهید. در کل این باس 20 واحد HP دارد که هر ضربه عادی 1 واحد و هر جامپ اتک موفق 3 واحد آسیب به آن وارد می‌کند. در نهایت وقتی HP باس به صفر برسد، می‌میرد و بازی به سلیقه خودتان به پایان خواهد رسید. دقت کنید که شما باید HP باس را در هر لحظه روی HP Bar بالا یا پایین صفحه نمایش دهید.

محیط باس‌فایت

تعدادی بلوک و آیتم را به صورت دلخواه در محیط باس‌فایت قرار دهید تا در طول باس‌فایت بتوان آیتم‌های مختلف را دریافت کرد و همچنین با بالا رفتن از این بلوک‌ها و پریدن روی باس از بالا، به آن آسیب زد. (دقت کنید با هر بار ورود دوباره به محیط باس‌فایت، آیتم‌های موجود در بلوک‌های ستاره‌دار محیط باس‌فایت را تمدید کنید) بلوک‌های محیط باس‌فایت زمان‌دار هستند. به این معنا که هر یک از این بلوک‌ها دارای یک تایمر هستند که با ایستادن ماریو روی آنها به فعالیت ادامه می‌دهند و با جدا شدن ماریو از سطح آنها، متوقف می‌شوند. همچنین این تایمرها با رسیدن ماریو به سطح زمین، ریست خواهند شد. اگر تایمر یکی از این بلوک‌ها به زمان 2 ثانیه برسد نابود می‌شوند (مستقل از نوع این بلوک) و تنها زمانی که ماریو بمیرد (که موجب لود شدن مجدد کل مرحله می‌شود) این بلوک دوباره ایجاد خواهد شد.



مکانیک‌های باس

بازگشت به سنت‌ها

اگر HP Bar و موسیقی مخصوص باس فایت به اندازه کافی برای شما خاطره souls-like را تداعی نکرده است، با 2 Phase داشتن در باس فایت قطعاً روح سری دارک سولز را در ماریو حلول خواهد داد. در این باس فایت بعد از اینکه HP Bar کینگ کوپا به 50 درصد یا کمتر برسد، شما باید حرکت ماریو را برای چند ثانیه کاملاً غیرفعال کنید (به طوری که با زدن کلیدها ماریو واکنشی نشان ندهد (به استثنای Alt+F4)) و در این بین کینگ کوپا با مشت زدن به زمین باعث از بین رفتن تمام بلوك‌های موجود در محیط باس فایت می‌شود و زمین بازی آتش می‌گیرد. سپس ماریو به طور خودکار به حالت آتشی تبدیل می‌شود. وقت کنید در این صحنه‌ها ماریو نباید آسیب ببیند و کینگ کوپا نباید حمله کند (این صحنه حالت cut scene خواهد داشت) و در نهایت باس فایت مجدداً مانند حالت قبل ادامه پیدا خواهد کرد. (اتک‌های کینگ کوپا مجدداً شروع خواهد شد و ماریو بار دیگر توان حرکت و حمله به کینگ کوپا خواهد داشت) همچنین در این مرحله یک اتک به کینگ کوپا اضافه خواهد شد که در قسمت بعد توضیح داده شده است.
(برای اینکه کاملاً شباهت این مکانیک به بازی‌های سری دارک سولز را درک کنید این [ویدیو](#) را ببینید)

اتک‌های باس

Nuke Attack	Fireball Attack	Jump Attack	Grab Attack
			
<ul style="list-style-type: none"> - در این حمله کینگ کوپا با فشار دادن دکمه ای که در دست دارد (پیاده‌سازی گرافیکی فشار دادن دکمه به انتخاب خودتان است) یک (x-position) بمب در عرض (x-position) رندهم از محیط باس فایت از آسمان رها می‌شود - این بمب بعد از برخورد به اولین سطح، منفجر شده و به هر موجودی (اعم از ماریو یا کینگ کوپا) در فاصله کمتر یا مساوی با 1 بلوك، آسیب می‌زند - این حمله کینگ کوپا تنها در Phase2 فعال می‌شود 	<ul style="list-style-type: none"> - در این حمله کینگ کوپا گلوله‌های آتشین به سوی ماریو پرتاب خواهد کرد. این گلوله‌ها در دو ارتفاع مختلف به سوی ماریو پرتاب خواهند شد. این گلوله‌ها با سرعتی برابر با سرعت ماریو به سوی او در راستای افقی پرتاب می‌شوند و تنها در صورت برخورد به ماریو یا یک بلوك، از بین می‌روند - در صورت اصابت گلوله به ماریو، ماریو آسیب خواهد دید. اندازه هر یک از این گلوله‌ها، 2 بلوك خواهد بود - گلوله‌های با ارتفاع پایین باید طوری تنظیم شوند که با ثابت ایستادن یا نشستن ماریو به او برخورد کنند و ماریو با پریدن بتواند از آنها جاخالی دهد - گلوله‌های با ارتفاع بالا باید طوری تنظیم شوند که با ثابت ایستادن یا پریدن ماریو به او برخورد کنند و ماریو بتواند با نشستن از آنها جاخالی دهد 	<ul style="list-style-type: none"> - در این حمله کینگ کوپا به اندازه دو بلوك می‌پردد و وارونه در همان نقطه فرود می‌آید - در صورتی که ماریو موقع فرود زیر کینگ کوپا واقع شده باشد آسیب می‌بیند - در غیر این صورت در صورتی که ماریو در لحظه برخورد کینگ کوپا با زمین، روی زمین قرار گرفته باشد، به مدت 5 ثانیه گیج شده و توان حرکت درست را از دست خواهد داد. در این حالت، دکمه‌های بازی عملکرد اشتباھی خواهند داشت. (نحوه پیاده‌سازی بر عهده خودتان است، برای مثال ممکن است با فشردن کلید راست، ماریو بپردد...) و بعد از این زمان 5 ثانیه ای حرکت ماریو به حالت عادی باز خواهد گشت - در صورتی که در لحظه برخورد با زمین، ماریو روی هوا یا روی یکی از بلوك‌های بالاتر از سطح زمین باشد، این حرکت اثری بر او نخواهد داشت 	<ul style="list-style-type: none"> - در این حمله کینگ کوپا ماریو را در چنگ می‌گیرد (در زمانی که ماریو گرفتار است، توان حرکت و حمله را نخواهد داشت. اما در عین حال ماریو می‌تواند با حرکت به راست و چپ (فسردن متوالی دکمه‌های راست و چپ) خود را از چنگ کینگ کوپا نجات دهد. اگر مجموعاً 10 بار این کلیدها فشرده شوند، ماریو رها خواهد شد - کینگ کوپا 5 ثانیه ماریو را در چنگ خود نگه می‌دارد (مگر اینکه ماریو زودتر از پایان این مدت خود را آزاد کند. اگر ماریو نتواند خود را زودتر از این زمان آزاد کند، آسیب می‌بیند.) اگر ماریو در هر یک از این دو حالت از چنگ کینگ کوپا رها شود، به سمت دیگر کینگ کوپا پرتاب می‌شود و به اندازه دو بلوك دور می‌شود

3 ثانیه cooldown دارد

2 ثانیه cooldown دارد

3 ثانیه cooldown دارد

4 ثانیه cooldown دارد

عملکردها و واکنش‌های باس

- کینگ کوپا در طول باس فایت همواره رو به ماریو خواهد بود تا در صورت حمله، واکنش مناسب نشان دهد.
- کینگ کوپا در طول باس فایت قابلیت پرش خواهد داشت و ارتفاع پرش او به اندازه 2 بلوك است.
- کینگ کوپا در طول حرکاتش در محیط باس فایت، همواره فاصله **حداقل دو بلوك** را از دیوارهای محیط باس فایت حفظ خواهد کرد. وقت کنید این اصل در اولویت بر باقی مکانیزم‌های حرکتی کینگ کوپا خواهد بود. به طور مثال اگر کینگ گوپا امکان انجام Grab Attack را داشته باشد و با انجام آن فاصله آن با دیوار کمتر از دو بلوك بشود، کینگ کوپا این حرکت را انجام نداده و به نحوه دیگری عمل می‌کند تا شرط فاصله را نقض نکند.
- در صورتی که ماریو به سمت کینگ کوپا تیر پرتاب کند، کینگ کوپا با احتمال (D) با یک پرش از تیر پرتاب شده جا خالی خواهد داد که در آن D فاصله بین ماریو و کینگ کوپا است. برای جلوگیری از اینکه کاربر دور از کینگ کوپا ایستاده و صرفاً با پرتاب تیر او را شکست دهد، (اصطلاحاً Cheesing) توزیع این احتمال را به صورت زیر تعریف می‌کنیم:

- If $D \geq 8$ Blocks, then $p(D) = 1$
- If $D \leq 8$ Blocks, then $p(D) = \frac{D}{8}$

- ارتفاع کینگ کوپا باید به اندازه‌ای باشد که ماریو در هیچ حالتی نتواند از سطح زمین روی سر او بپرد. (حداقل 4 بلوك)
- بلوك‌ها مانع از حرکت کینگ کوپا نمی‌شوند! در صورتی که کینگ کوپا به هر نحوی به بلوك پرخورد کند، آن بلوك (از هر نوعی که باشد) از بین می‌رود و کینگ کوپا به حرکت برنامه‌ریزی شده ادامه می‌دهد.
- در صورتی که کینگ کوپا به هر نحوی آسیب ببیند، به مدت یک ثانیه قادر به حرکت/حمله نخواهد بود و پس از مدت زمان یک ثانیه، به حرکت برنامه‌ریزی شده خود ادامه می‌دهد.
- در زمانی که ماریو روی بلوك‌های محیط باس فایت در حرکت است، کینگ کوپا نباید بیش از یک بلوك جابجا شود. این مکانیزم به ماریو کمک می‌کند که در تلاش برای آسیب زدن به کینگ کوپا از طریق فرود روی او موفق‌تر باشد.
- مکانیزم‌های تهاجمی (Aggro): این مکانیزم‌ها در راستای جلوگیری از بازی محتاطانه و فرصت‌طلبانه (اصطلاحاً Camping) و درگیر شدن بیشتر پلیر و فعال‌تر بودن او (Engaging Gameplay) طراحی می‌شوند. مکانیزم‌های تهاجمی کینگ کوپا به شرح زیر خواهند بود:

- در صورتی که کینگ کوپا در فاصله بیشتر از 8 بلوك از ماریو قرار داشته باشد، با سرعت ثابت 1.5 بلوك بر ثانیه به سمت ماریو می‌دود تا فاصله خود با ماریو را به کمتر از 3 بلوك کاهش دهد و سپس متوقف می‌شود.
- در صورتی که فاصله کینگ کوپا از ماریو بین 6 تا 10 بلوك باشد از Fireball Attack استفاده می‌کند.
- در صورتی که ماریو در فاصله کمتر از 2 بلوك از کینگ کوپا قرار داشته باشد از Grab Attack استفاده می‌کند.
- وقت کنید در صورت جا خالی دادن ماریو و عدم لمس او توسط کینگ کوپا، این حمله ناموفق خواهد بود.
- در صورتی که ماریو 4 ثانیه به صورت ممتد روی زمین باقی بماند، کینگ کوپا از Jump Attack استفاده می‌کند.
- پیاده‌سازی استفاده از حملات در شرایط دیگر به سلیقه خودتان خواهد بود اما وقت کنید باید از همه حملات در طی پیاده‌سازی این باس فایت استفاده کنید.

- در صورتی که طبق مکانیزم‌های قبلی کینگ کوپا بی‌حرکت بماند، با سرعت ثابت 0.75 بلوك بر ثانیه به اندازه حداقل 2 بلوك به صورت تصادفی به یکی از جهت‌های مجاز حرکت می‌کند. (وقت کنید طبق مکانیزم سوم، در برخی شرایط کینگ کوپا نمی‌تواند به هر جهتی حرکت کند)

بارگذاری بازی

در این بخش می‌خواهیم یک Editor Mode ساده به بازی ماریو اضافه کنیم که در آن کاربر می‌تواند بازی خودش را بسازد و به او این امکان را بدهیم که یک فایل حاوی اطلاعات هر مرحله بارگذاری کند و یک بازی ماریو برای او بسازیم. در ادامه در مورد الگوی این فایل توضیح می‌دهیم. دقت کنید که نحوه بارگذاری فایل آماده به انتخاب خودتان می‌باشد. به طور مثال می‌توانید یک دکمه دیگر به منو بازی اضافه کنید تا با کلیک روی آن و انتخاب یک فایل مرحله، این مرحله ساخته و لود شود.

الگوی فایل (JSON Object)

```
File {  
    levels: LevelObject[]  
    hearts: Int  
    marioState: Int  
}
```

- **levels**: شامل لیست مراحل این بازی است.
- **hearts**: تعداد قلب‌های ماریو را مشخص می‌کند.
- **marioState**: مشخص‌کننده حالت ماریو است و در آن مقادیر 0,1,2 به ترتیب نمایان‌گر ماریوی آتشین، ماریوی معمولی (مگا ماریو) و ماریوی کوچک (مینی ماریو) هستند.

الگوی مرحله

```
LevelObject {  
    sections: SectionObject[]  
}
```

- **sections**: شامل لیست بخش‌های مرحله است.

الگوی بخش

هر بخش را یک صفحه با ارتفاع 20 بلوک و طول دلخواه در نظر بگیرید. حال می‌توانیم در کانفیگ، به هر نقطه از مختصات بلوکی یک شیء تخصصی دهیم. به طور مثال می‌توان گفت که در نقطه‌ای با مختصات (10, 15) یک بلوک توخالی قرار دارد. این مدل پیاده‌سازی به شما این امکان را می‌دهد که خودتان هر شیء را در هر نقطه‌ای که بخواهید در صفحه بازی نمایش دهید.

```
SectionObject {  
    length: Int  
    time: Int  
    blocks: BlockObject[]  
    enemies: EnemyObject[]  
    pipes: PipeObject[]  
    spawnPipe?: PipeObject  
}
```

- **length**: طول بخش را تعیین می‌کند.
- **time**: زمان بخش را به ثانیه مشخص می‌کند.
- **blocks**: شامل لیست بلوک‌های بخش است.
- **enemies**: شامل لیست دشمن‌های بخش است.
- **pipes**: شامل لیست لوله‌های بخش است.
- **spawnPipe**: یک فیلد اختیاری است. اگر ورود به این بخش از طریق یک لوله صورت بگیرد، این فیلد اجباری است (مثل ورود/خروج به یک بخش مخفی) و لوله‌ای که ماریو از طریق آن وارد این بخش می‌شود را مشخص می‌کند.

الگوی لوله

```
PipeObject {  
    x: Int  
    y: Int  
    type: PipeType  
    section?: SectionObject  
    activated?: boolean  
}
```

- **:x**: مولفه طول مکان لوله در صفحه بازی
- **:y**: مولفه عرض مکان لوله در صفحه بازی
- **:type**: نوع لوله را تعیین می‌کند، اعم از SIMPLE/PIRANHA_TRAP/TELE_SIMPLE/TELE_PIRANHA/DECEIT
- **:section**: یک فیلد اختیاری است. اگر لوله از نوع مخفی باشد (TELE_SIMPLE/TELE_PIRANHA) این فیلد اجباریست و بخشی که ماریو با نشستن روی این لوله مخفی، وارد آن می‌شود را مشخص می‌کند.
- **:activated**: یک فیلد اختیاری است. اگر لوله از نوع Pipe Of Deceit باشد (DECEIT) این فیلد اجباریست و مشخص می‌کند که آیا تله مربوط به این لوله قبلاً فعال شده است یا خیر.

الگوی بلوک

```
BlockObject {  
    x: Int  
    y: Int  
    type: BlockType  
    item?: ItemType  
}
```

- **:x**: مولفه طول مکان بلوک در صفحه بازی
- **:y**: مولفه عرض مکان بلوک در صفحه بازی
- **:type**: نوع بلوک را تعیین می‌کند، اعم از SIMPLE/COIN/COINS/EMPTY/QUESTION/SLIME/FIREBAR
- **:item**: یک فیلد اختیاری است. اگر مدل بلوک از نوع بلوک علامت سوال باشد، این فیلد اجباری است و مدل آیتم COIN/STAR/MUSHROOM/FLOWER درون بلوک را مشخص می‌کند، اعم از

الگوی دشمن

```
EnemyObject {  
    x: Int  
    y: Int  
    type: EnemyType  
}
```

- **:x**: مولفه طول مکان دشمن در صفحه بازی
- **:y**: مولفه عرض مکان دشمن در صفحه بازی
- **:type**: نوع دشمن را تعیین می‌کند، اعم از GOOMBA/KOOPA/SPINY/NUKEBIRD/BOWSER

پیاده‌سازی مراحل بازی

همانطور که در ابتدای داک ذکر شده، بازی شما باید شامل سه مرحله باشد و هر یک از مراحل نیز دارای حداقل 5 بخش است. در پایان مراحل اول و دوم باید مثل بازی اصلی ماریو یک بخش (section) تنها شامل یک پرچم طراحی کنید که بعد از اینکه ماریو آن را لمس کند، پرچم پایین کشیده شود و ماریو بعد از رد شدن از آن مرحله را به پایان برساند. اما در مرحله سوم بازی، باید در انتهای مرحله باس‌فایت کینگ کوپا را پیاده‌سازی کنید و در صورتی که ماریو موفق به شکست دادن او شود، بازی به پایان می‌رسد. (تصمیم‌گیری برای پایان داستان و پیاده‌سازی آن به سلیقه خودتان است) از بین این سه مرحله، طراحی مرحله اول و دوم به عهده خودتان است ولی مرحله سوم بازی به صورت یک فایل کانفیگ (با فرمتی که در بخش قبل ارائه شد) به شما داده می‌شود و شما صرفاً باید آن را لود و اجرا کنید.

مرحله اجباری

همانطور که اشاره شد، شما نیازی به طراحی مرحله سوم بازی ندارید و محیط این مرحله به صورت یک فایل کانفیگ به شما داده می‌شود. این فایل کانفیگ را می‌توانید از [این لینک](#) دانلود کنید.

نکات پایانی

در طی پیاده‌سازی این فاز، از طیف وسیعی از مباحث تدریس شده در کلاس و کارگاه‌ها می‌توانید کمک بگیرید پس توصیه می‌شود قبل از دست به کد شدن محتوا آموزشی تهیه شده توسط تیم درس و منابع معرفی شده توسط استاد را به دقت مطالعه کنید تا در پیاده‌سازی به مشکل نخورید. همچنین برای سادگی کار شما، یک فایل از asset‌های ماریو که در تولید این داک مورد استفاده قرار گرفته در [این لینک](#) اختیار شما گذاشته می‌شود اما شما را تشویق می‌کنیم تا خودتان تصاویر خودتان را برای بازی‌تان بسازید/پیدا کنید تا بازی منحصر به فردتر و جذاب‌تری تولید کنید.

مختومه‌ای بر Game Design

همانطور که در داک به وضوح مشخص بوده است، بازی تعریف شده تفاوت‌های بزرگی با بازی اصلی ماریو دارد که عمدۀ این تغییرات برای متوازن‌تر کردن مکانیزم‌های بازی هستند. در طی تولید و طراحی یک بازی، نکات تئوری زیادی وجود دارند که می‌توانند در عین ریز و کم اهمیت بودن، بازی‌ها را بسیار جذاب یا خیلی حوصله سر بر و تکراری کنند. چنین مواردی باعث می‌شوند بازی‌هایی بسیار ساده مثل Stardew Valley موفق‌تر از بازی‌هایی با گرافیک بسیار پیچیده‌تر مثل Final Fantasy و Demon's Souls ظاهر شوند.

فناوری‌هایی مثل Death Design, Game Economy, Combat System, Camera Design تفاوت‌های شگرفی در درگیر شدن پلیرها در بازی ایجاد می‌کنند و در نتیجه بازی‌هایی که در این موارد به نحو احسن عمل می‌کنند، در نهایت هم بسیار موفق‌تر از دیگر بازی‌ها خواهند بود. لینک‌های زیر برای علاقمندان به این موضوعات قرار داده می‌شود:

1. [GMTK: Design of games' economy](#)
2. [GMTK: Design of combat systems](#)
3. [GMTK: How physics affect platformer games](#)
4. [GMTK: Analysis of video game designs](#)

باز هم بخش امتیازی! (تمامی قسمت‌های توضیح داده شده در این بخش [امتیازی](#) هستند)

استفاده از ابزار گیت

برای توسعه پروژه خود پیشنهاد می‌شود از ابزارهای **version control** مثل Git استفاده کنید. استفاده صحیح از Git برای پروژه دارای نمره امتیازی خواهد بود. برای کسب نمره امتیازی لازم است موارد زیر را رعایت کنید:

- یک ریپازیتوری پرایوت داشته باشید و نسخه اولیه پوش شده مربوط به نسخه فاز یک پروژه باشد
- برای هر فاز پروژه یک Branch و یک Development Branch داشته باشید
- در طول توسعه پروژه در Git، حتماً کامیت‌های منظم داشته باشید و توضیحات کامیت را بنویسید

استفاده از ابزار Build Tool (میون و گریدل)

پروژه شما باید شامل یک کانفیگ کامل از ابزارهای Maven, Gradle باشد به طوری که فقط با استفاده از سورس پروژه بتوان آن را بیلد و اجرا کرد. به این معنا که کتابخانه‌های مورد استفاده در پروژه به طور خودکار دانلود و استفاده شوند. یک تمپلیت آماده برای Gradle در گیت‌هاب درس قرار خواهد گرفت.

استفاده از ابزار JLINK

باید با استفاده از Build Tool مورد نظر خود و ابزار JLINK بتوانید از پروژه یک فایل قابل اجرا دریافت کنید که روی هر سیستم عامل مشابهی بدون نیاز به نصب جاوا یا آماده‌سازی دیگری اجرا شود. برای این منظور می‌توانید از تمپلیت قرار داده شده برای پروژه روی گیت‌هاب درس استفاده کنید.

استفاده از روش‌های مبتنی بر PCA

همانطور که ذکر شد، تنها پیاده‌سازی دشمن‌ها، بلوک‌ها و لوله‌ها با هیتباکس مستطیل شکل از شما خواسته شده است، اما دقیق کردن این هیتباکس‌ها نمره امتیازی خواهد داشت. نحوه‌های پیاده‌سازی به شرح زیر است:

		
<p>پیاده‌سازی دقیق: هیتباکس هر یک از entity‌ها دقیقاً منطبق بر تصویر آن است و هیچ نقطه اضافه‌ای ندارد</p>	<p>پیاده‌سازی تقریب چندضلعی: هیتباکس هر یک از entity‌ها یک چندضلعی محصور کننده تصویر آن است که شکل آن را تقریب می‌زند برای مثال، پوش محدب تعدادی از نقاط روی محیط تصویر آن</p>	<p>پیاده‌سازی جعبه‌ای: هیتباکس هر یک از entity‌ها صرفاً مستطیل محصور کننده تصویر آن است</p>
نمره امتیازی بیشتری دارد	نمره امتیازی دارد	نمره امتیازی ندارد

با آرزوی موفقیت و سربلندی برای شما

تیم درس برنامه‌نویسی پیشرفته

