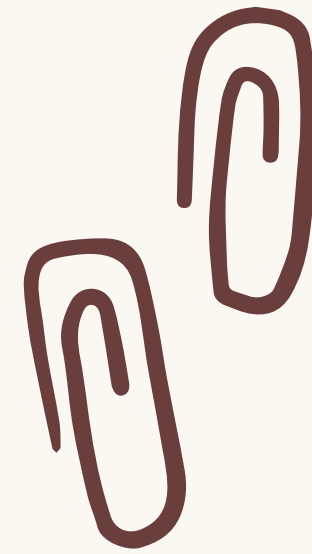
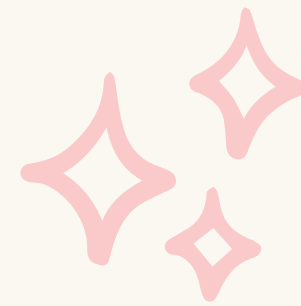




Game Design

Miriam López



Introducción

Vamos a aprender lo más básico para poder empezar el camino de diseño de videojuegos. En esta clase veremos los elementos básicos de un diseño de juego y luego exploraremos el diseño de niveles.

hello



Mecánicas

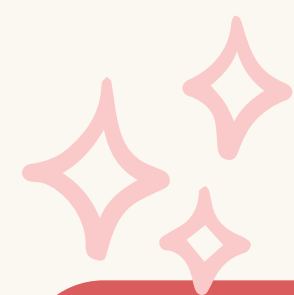
Mecánicas



**Ejemplo:
Mario Bros**

Las mecánicas son las reglas y elementos que rigen el juego, su comportamiento y las acciones que puede realizar el jugador.

Un ejemplo de mecánica es el salto del Mario. Como el jugador, al pulsar el botón correcto, Mario salta.



Otros ejemplos en Mario

Puntuaciones

Cuando Mario mata un enemigo, pisa una concha o hace una serie de acciones, ganará puntos. Son acumulativos, si consigues más en poco tiempo consigues cada vez más hasta llegar a una vida



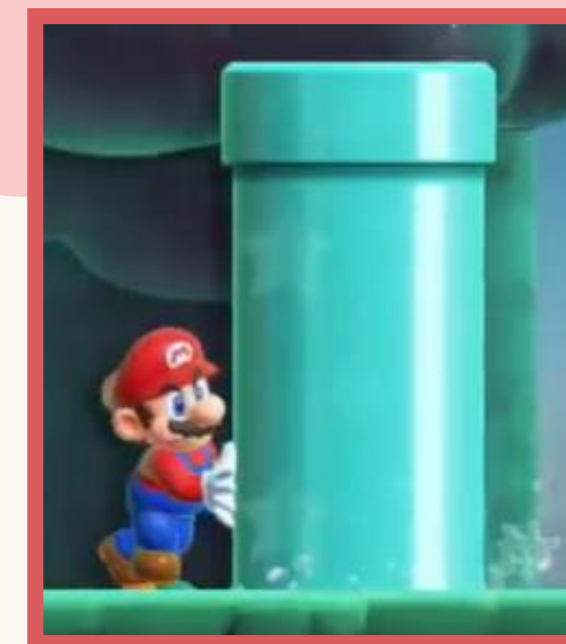
Coger / golpear objetos

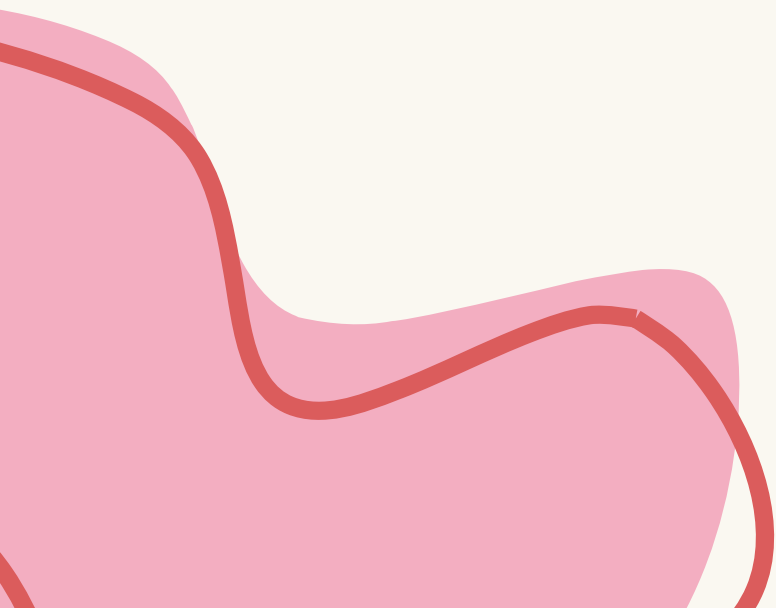
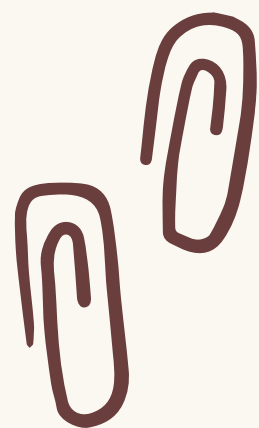
Mario puede coger y golpear algunos elementos del mapa, como muelles o conchas de Koopas (tortuguitas).



Mover objetos

Mario puede, además de coger, mover objetos en el mapa. Por ejemplo, en el nuevo Mario Wonder, podemos empujar tuberías.







Dinámicas

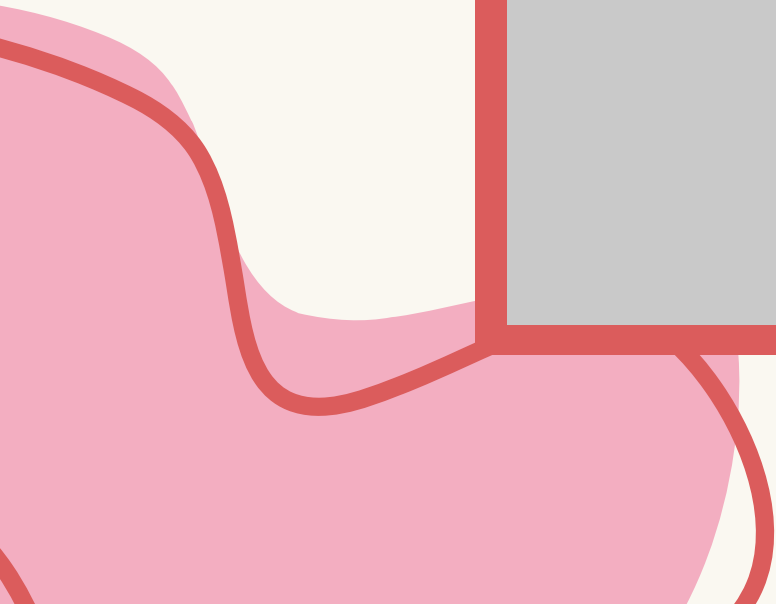
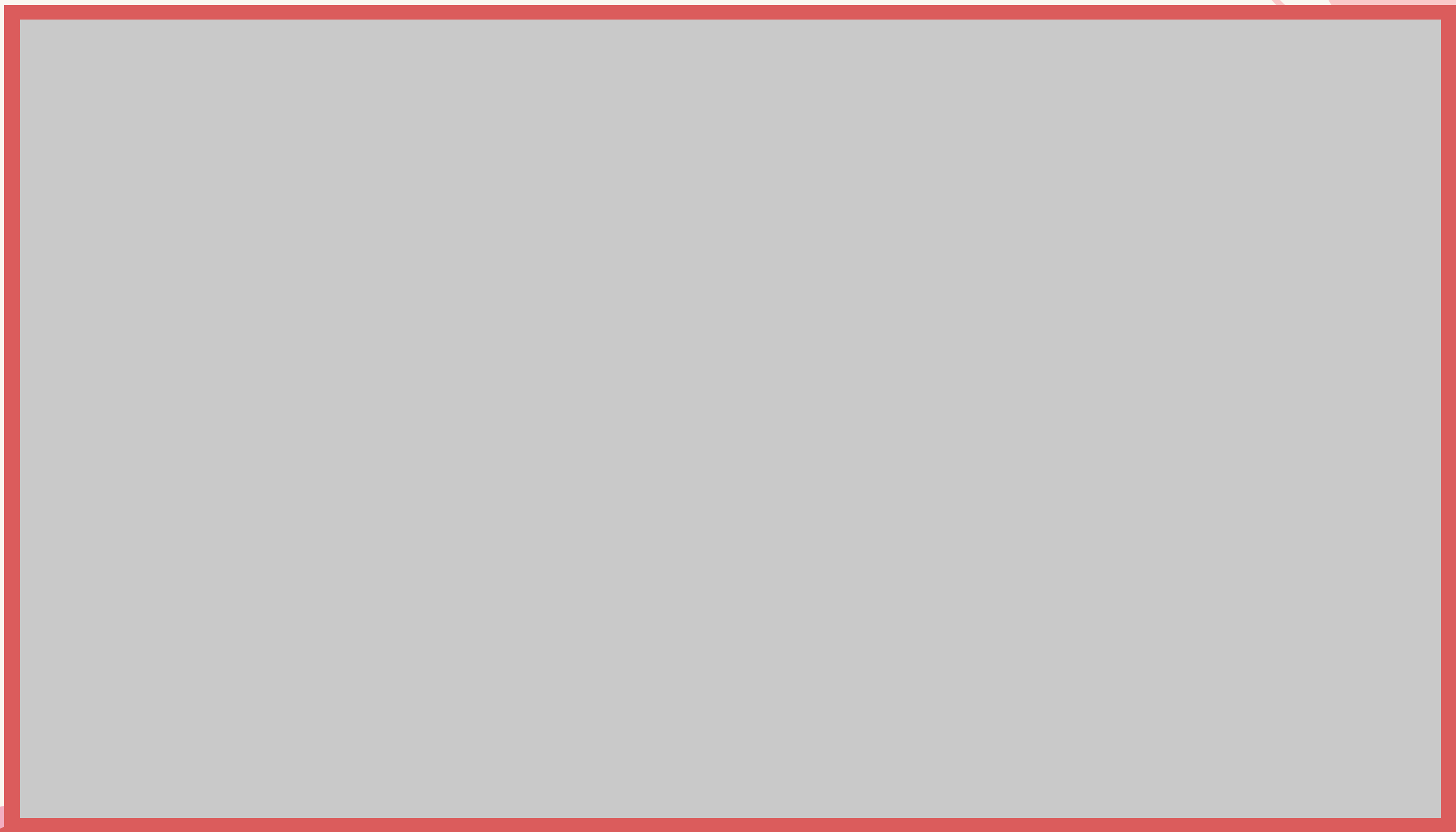


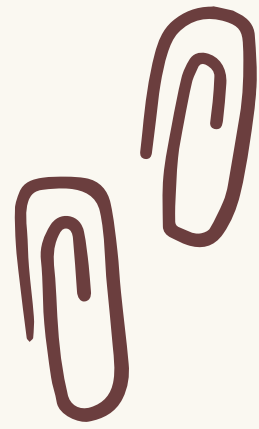
Dinámicas



Las dinámicas son el resultado de la interacción de las mecánicas del juego con ellas mismas y la interacción con el jugador.

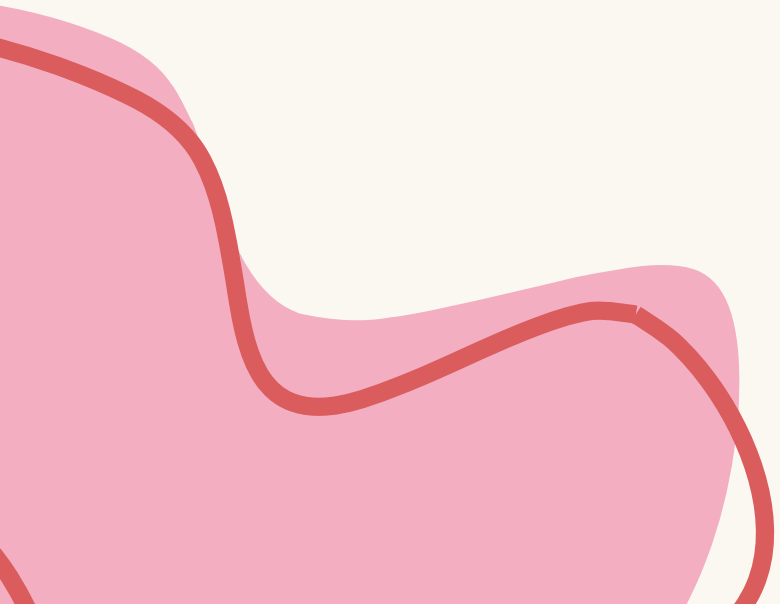
Siguiendo con Mario, si Mario al saltar aterriza en un Koopa, hará que se meta en su caparazón. Al golpear al caparazón, hará que se mueva y puede matar a los enemigos (Goombas por ejemplo) a su paso.





**Otros
ejemplos**

**Mecánicas en
nuestro
juego**

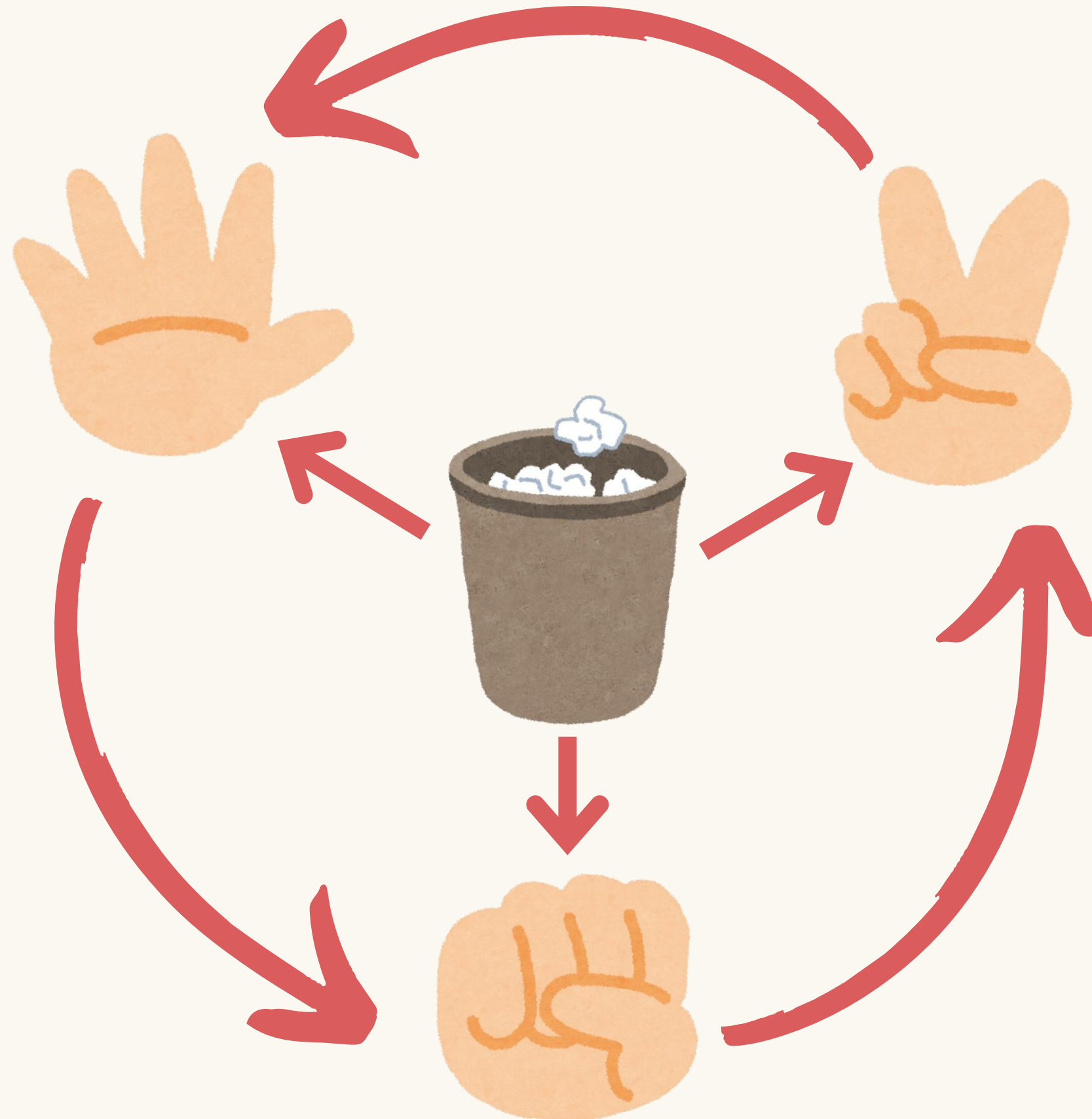




Estrategias dominantes

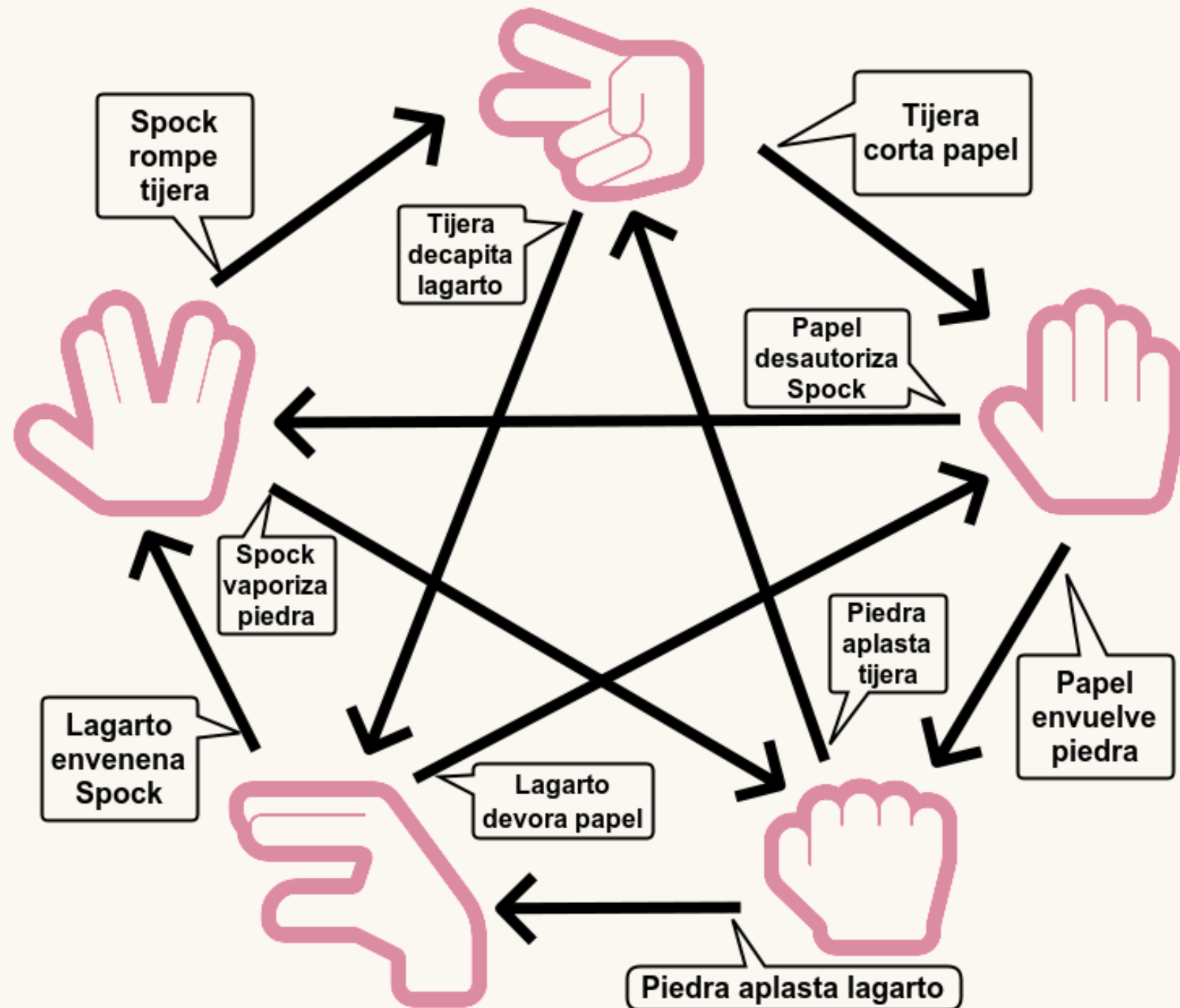
Estrategias dominantes

NO



Las estrategias dominantes son aquellas que hacen a un jugador superior a otro no importa lo que el otro pueda hacer. Por ejemplo, si en piedra, papel o tijera, metiésemos un cuarto elemento llamado papelera que ganase a todos, el juego se rompería, porque da igual lo que el otro jugador saque, va a perder igual.

Estrategias dominantes



Es por esto que siempre hay que tener en cuenta las opciones que tienen los jugadores y hacerlas justas. Es decir, **balancear el juego**. En el caso de piedra, papel o tijera, por ejemplo, sí se pueden añadir más opciones, pero haciéndolas justas, es decir, que ganen a unos elementos pero pierdan contra otros. Por ejemplo: Piedra, papel, tijeras, lagarto, Spock. Una variante famosa por la serie *Big Bang Theory*.



Tipos de diseño

Tipos



No hay un solo tipo

Para un buen diseño de videojuego, es necesario tener en cuenta muchos aspectos de este.

Desde el nivel, si tiene economía, el estilo artístico, el diseño combate...

Nos centraremos...

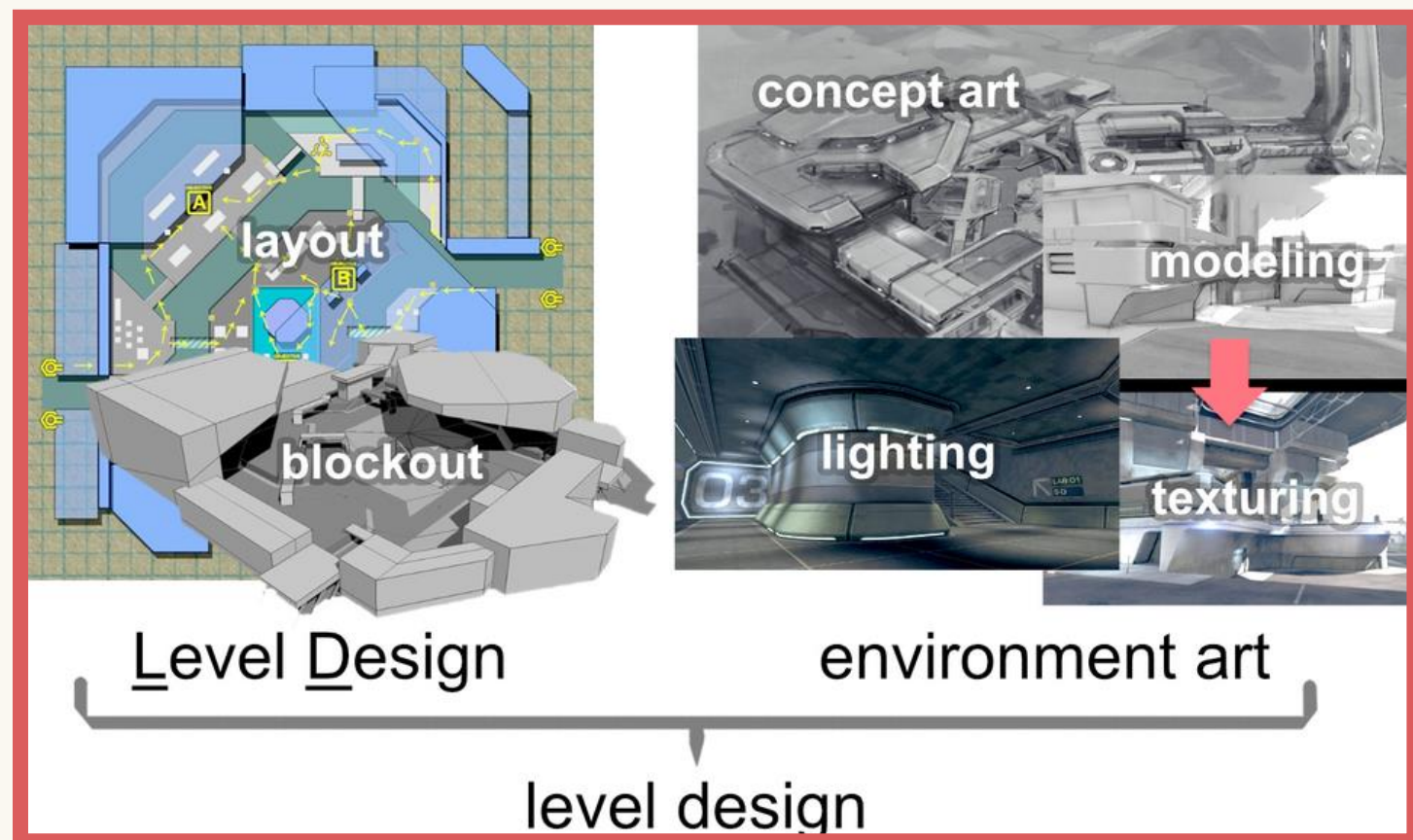
En esta sesión, haciendo uso de los assets vistos en la anterior de programación, nos vamos a centrar en el Level Design, o Diseño de Niveles.





Level Design

Level Design



El diseño de niveles es primordial en la creación de un juego.

Es necesario para poder, en primera instancia, enseñar al jugador a entender las mecánicas del juego, además de ofrecer una experiencia entretenida y retadora.

Dos pasos

Layout

El layout es el dibujo preliminar, los “planos” que se diseñan del nivel antes de crearlo.

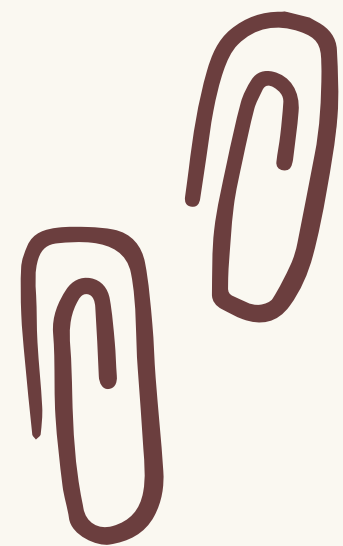
Se puede realizar en distintos programas como Adobe Illustrator o Krita.

Recomendación: [RPG map](#)

Blockout

Tras el layout, se pasa ese nivel al motor, creando un blockout, donde se verá si hay algo que cambiar en el layout. Se suelen realizar dentro del motor, para poder probar las mecánicas.





Ejemplo layout



Ejemplo blackout



Ejemplo blackout





Apectos a tener en cuenta

Mecánicas

Hay que tener en cuenta las mecánicas del juego. Mediante el diseño de niveles hay que enseñar al jugador a usarlas, ponerlas a prueba y demostrar que las ha entendido.

Riesgo-Recompensa

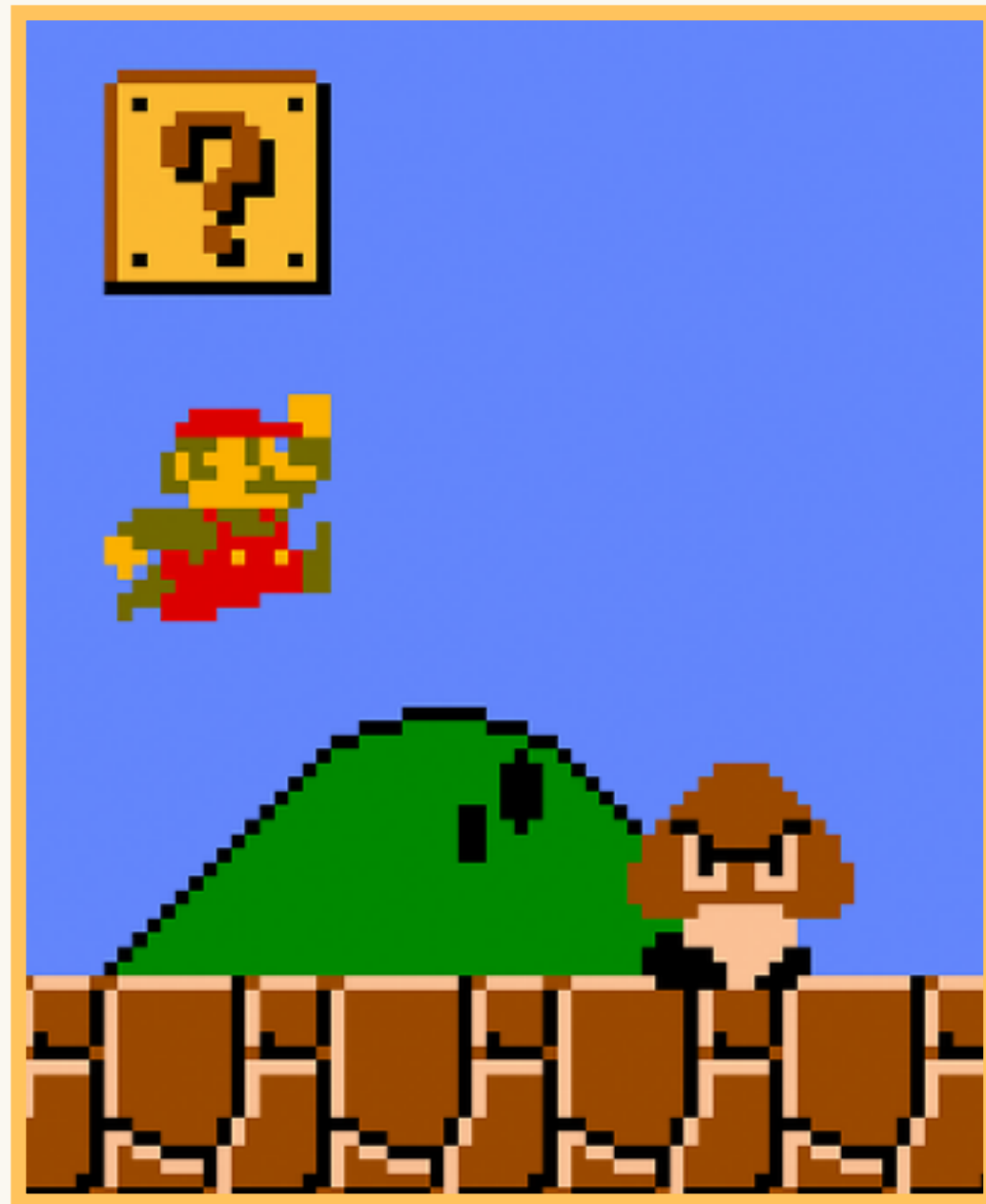
Método por el que se le ofrece al jugador un riesgo mayor al que normalmente se encontraría a cambio de conseguir una recompensa igual de poco habitual.

Guiar al jugador

No solo basta con hacer un buen layout, un buen diseño de nivel base, es importante saber cómo transmitimos al jugador qué camino debe seguir.

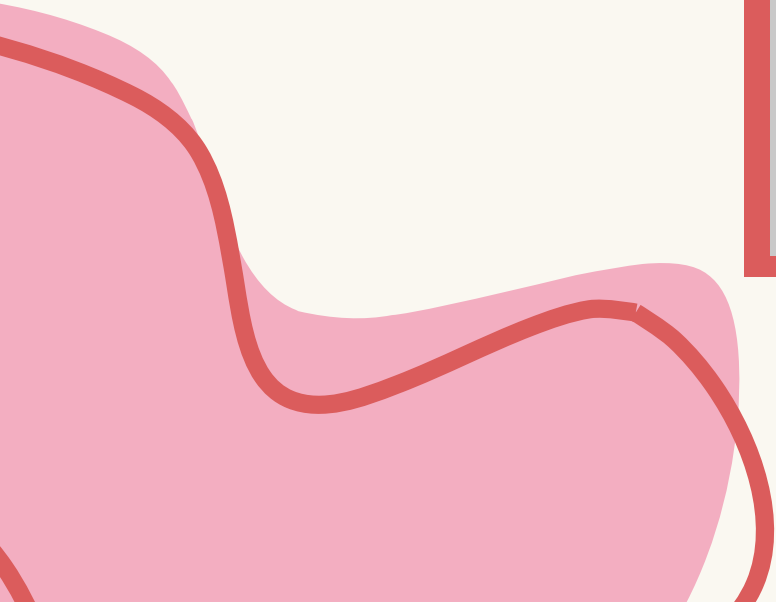


Mecánicas (again)

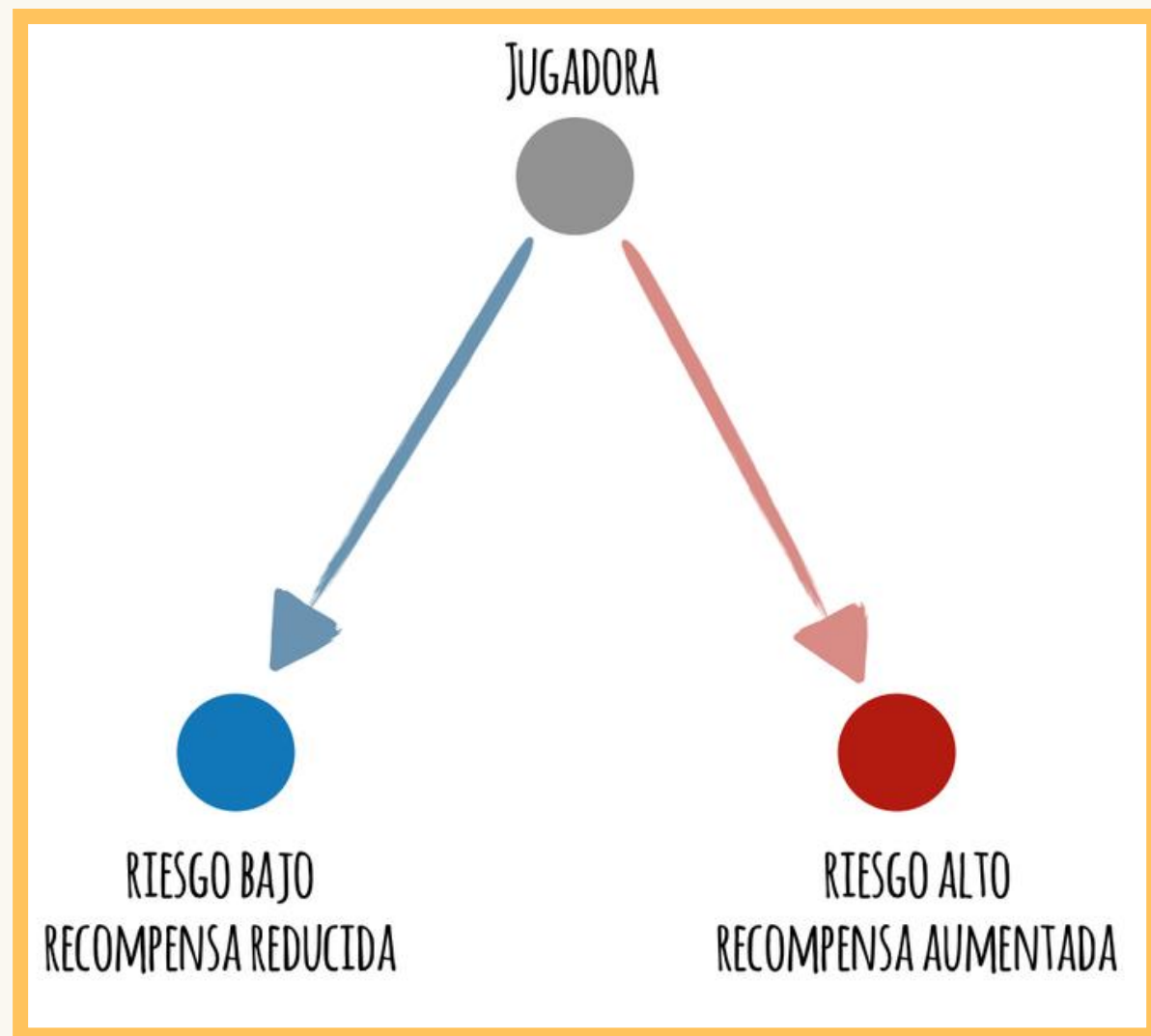


El level design nos ayuda a enseñar y comprobar que los jugadores han entendido las mecánicas. Volviendo al Mario Bros, esto se hace a modo de prueba, enfrentando al jugador con el primer momento donde puede morir. En el primer juego de Mario, un goomba se acercará al jugador. Si este no salta morirá. Esta es la primera prueba que le dejará ver si comprende las mecánicas o no.





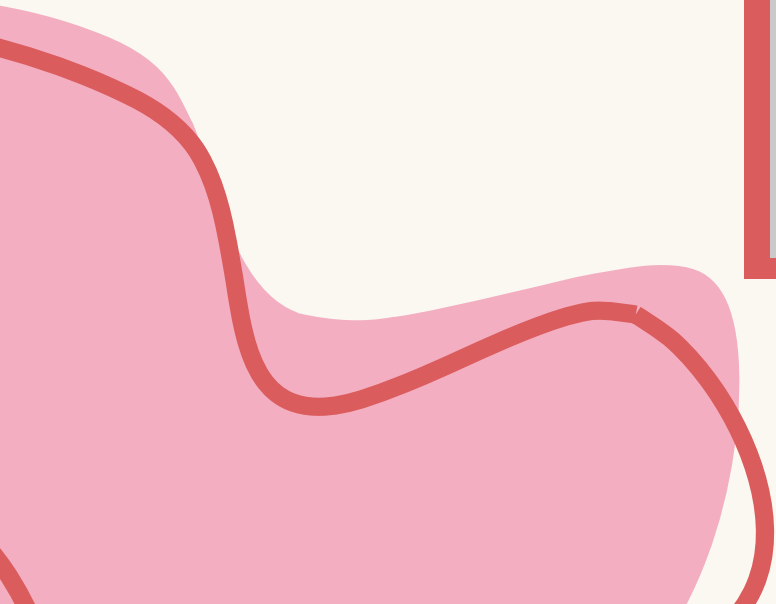
Riesgo-Recompensa (no pain no gain)

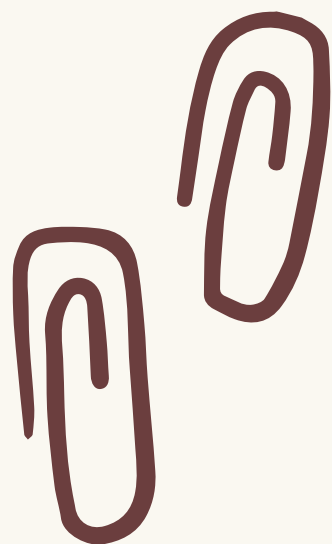


El riesgo recompensa es un método que se utiliza para presentar al jugador mayores retos, a cambio de mayores recompensas, pero de forma opcional. Es decir, pueden elegir asumir el riesgo o llevarse lo seguro y más fácil.

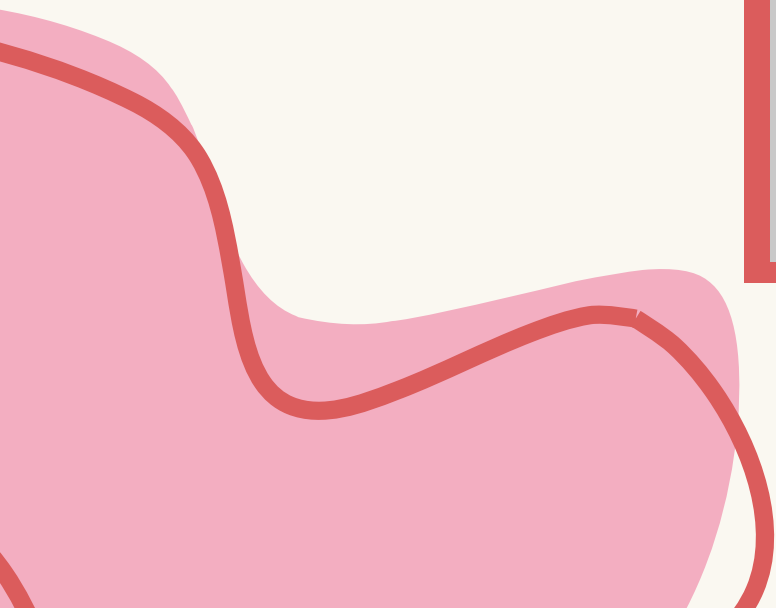
Esto se puede aplicar tanto a mecánicas normales como a level design. Si un camino alternativo es más difícil, ofrecemos una mejor recompensa para que el jugador quiera tomar ese camino.



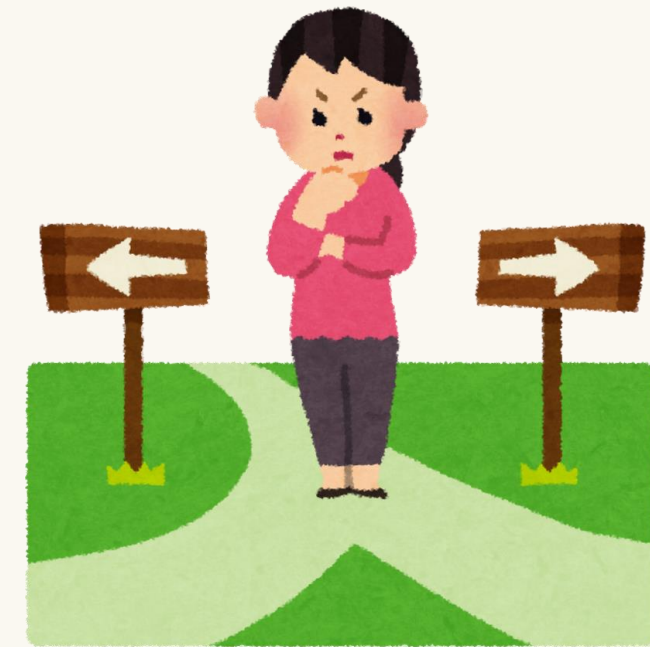




0:55

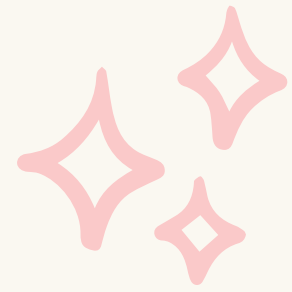


Guiar al jugador



No solo basta con hacer un buen layout, un buen diseño de nivel base, es importante saber como transmitimos al jugador qué camino debe seguir. Para esto, es necesario hacer uso de varias herramientas. Primero, una buena enseñanza al jugador de las mecánicas. Luego, un buen uso del arte y sus elementos y, por último, tener claro el camino que debe seguir el jugador.





Elementos a utilizar

Landmarks

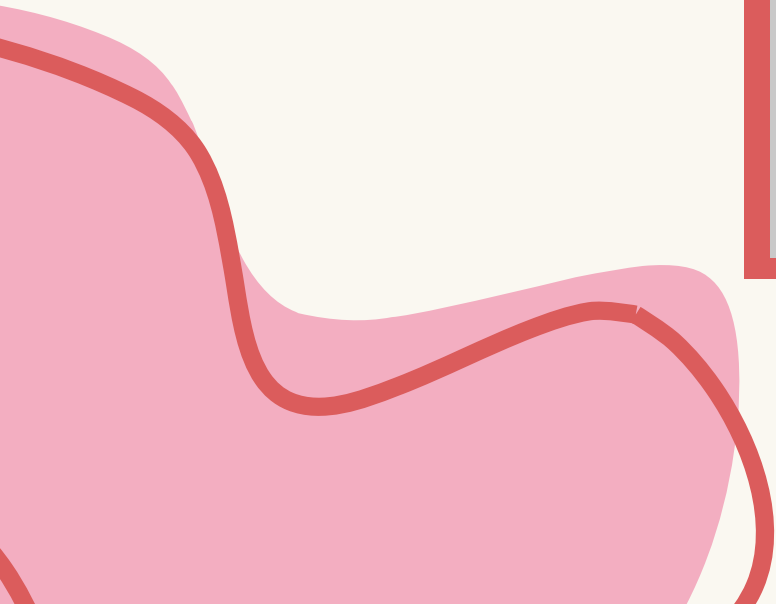
Ayudan a guiarse y a posicionarse en el mapa al jugador. Lo toma como meta y le ayuda a saber por donde seguir.

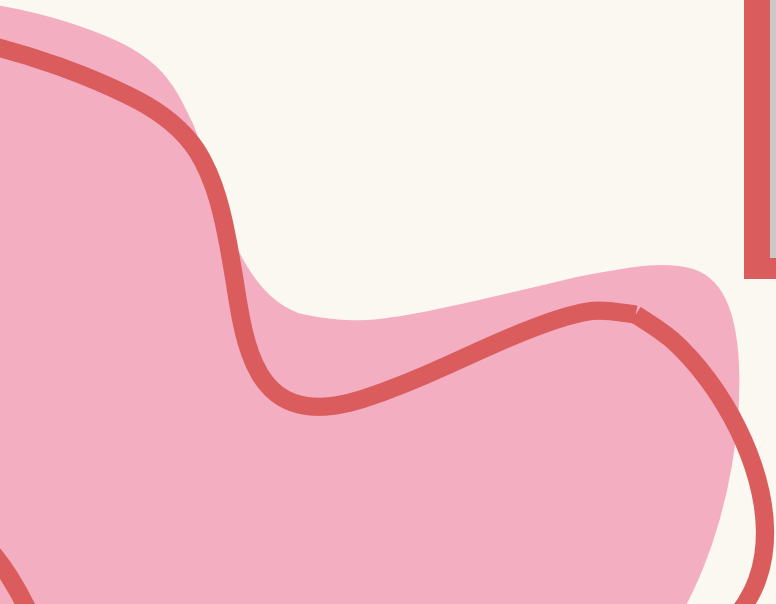
Luces

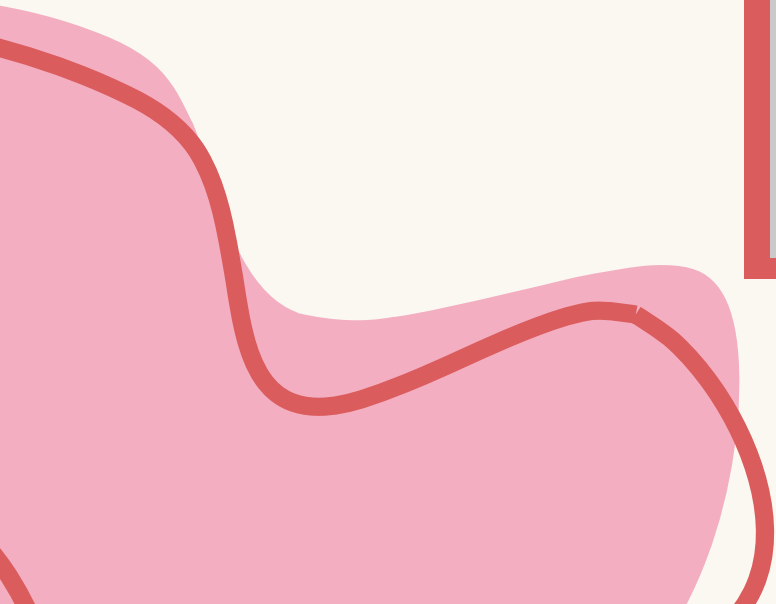
El uso de luces y paletas de colores nos ayuda a mostrar al jugador, no solo el camino, sino también reglas a recordar para saber guiarse mejor.
El uso de rojo en el Mirror's Edge, el uso de luces como en el Stray...

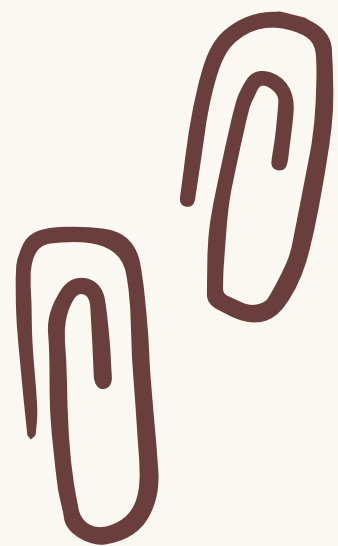
Colores











Recomendado!!





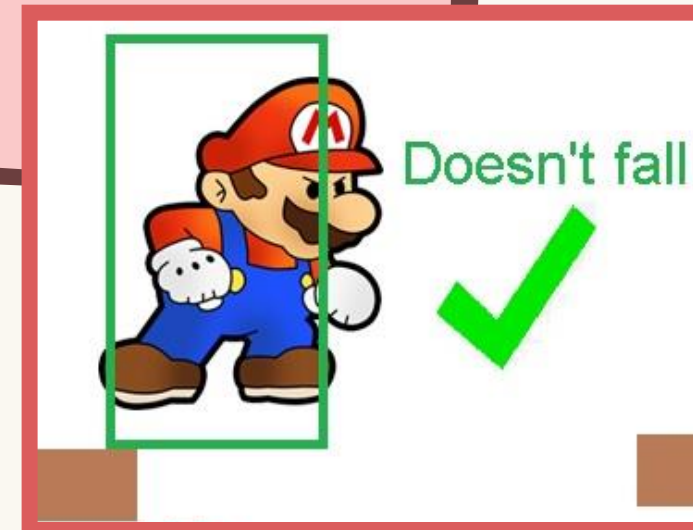
iiOJO!!

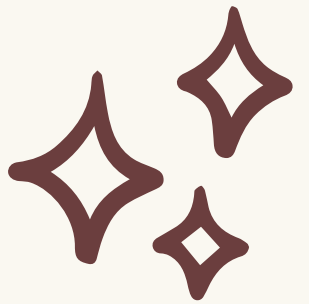
El jugador no es perfecto



No se debe pedir una ejecución perfecta de los controles, como tener que conseguir un salto muy justo en distancia o tiempo.

El jugador no es perfecto y no tiene por qué llegar a ejecutar los controles de la forma más optima posible, pero sin dejar al jugador saberlo. Es por esto que muchos juegos le hacen a los jugadores la vida más fácil poniendo pequeños trucos, como el coyote time.





¡¡A diseñar!!

