



One Page **GDD** *(Video Game Design Document)*

24/11/2023

ÍNDICE

INFORMACIÓN DEL PROYECTO INDIVIDUAL	3
OBJETIVOS GENERALES DEL PROYECTO	3
RESULTADOS ESPERADOS	3
APLICACIONES PRÁCTICAS DEL PROYECTO	3
RECURSOS A UTILIZAR	3
DESCRIPCIÓN DEL PROYECTO	4
PLAN DE TRABAJO PREVISTO	4
INFORMACIÓN BÁSICA	5
IDENTIDAD DEL JUEGO	5
PILARES DEL DISEÑO	5
GAME OUTLINE	5
<i>Características</i>	5
<i>Modos de juego</i>	6
<i>Futuras implementaciones</i>	6
GAME SUMMARY	7
SOUND FX / BSO	7
PROPUESTAS DE VALOR	7
ESTILO DE ARTE	8
INTERFAZ	11
SIMILAR PRODUCTS	12
HOJA DE RUTA DEL DESARROLLO	12
HITOS A LOGRAR	12

Stop Pollution

(One Page GDD 1.5)

Información del proyecto individual

Objetivos generales del proyecto

Aprender cómo funciona el *engine open source* Godot y hacerme al motor.
Aprender el lenguaje GDScript; un lenguaje de alto nivel que bebe de Python.
Desarrollar e implementar un *loop jugable* completo.
Desarrollar futuras funcionalidades que encajen en la temática del proyecto y estimar un tiempo el tiempo de implementación necesario.
Implementar un *game feel*¹ que dé una sensación *juicy*².
Implementar *sound FX* y una *bso* sencilla que acompañe cada partida.

Resultados esperados

Obtener el producto mínimo viable.
Aprender el funcionamiento básico de Godot y GDScript.
Aprender los fundamentos de los patrones de diseño enfocados al gamedev.

Aplicaciones prácticas del proyecto

Con el producto mínimo viable, terminar el desarrollo y publicar el resultado en los markets de [Itch.io](https://itch.io), [Steam](https://steamcommunity.com) y [Google Play Store](https://play.google.com/store/apps/details?id=com.stoppollution).

Recursos a utilizar

Como **motor gráfico**: [Godot Engine](https://godotengine.org).
Para el **apartado visual**: [Photopea](https://photopea.com) y [Libresprite](https://libresprite.com).
Para los **modelos 3D**: [Blender](https://www.blender.org)
Para la **bso**: [Bosca Ceoil](https://bosca-ceoil.com)
Para el **SFX**: [JFXR](https://github.com/JFXR) (GitHub)
También, algunos efectos de sonido gratuitos y loops musicales sacados de bibliotecas de sonido para, si fuese necesario, modificarlos y adaptarlos al proyecto con [Audacity](https://audacity.org).
Para las **paletas de colores**: <https://lospec.com/palette-list/>

¹ Corresponde a la sensación subjetiva que se experimenta al interactuar con un videojuego.

² Se dice que [un videojuego es juicy](https://lospec.com/palette-list/) cuando es capaz de transmitir una sensación al jugador mientras juega.

Descripción del proyecto

Stop Pollution es un videojuego que muestra el declive del planeta tierra en función de la cantidad de personas que lo habiten. Haciendo especial hincapié en los efectos que produce el cambio climático, volviendo inhabitable el entorno. Y la desigualdad en las luchas sociales donde [el 1% del planeta que poseen mayores recursos contaminan lo mismo que el 66% más pobre](#).

El apartado gráfico será en 2D, en pixel art; 16bits, con una paleta de colores como es [Pico 8](#).

El apartado sonoro contará con efectos de sonido clásicos.

Las mecánicas jugables consisten en: Arrastrar y soltar.

El objetivo final del juego es evitar la polución el mayor tiempo posible.

Las mecánicas no-jugables tienen que ver con la generación de Pufs; el componente básico, empresas, piquetes, oligopolios y esquiroles. Todos generan polución en mayor o menor medida.

Plan de trabajo previsto

1. **Planificar:** Desarrollar un GDD donde se queden establecidas las bases de trabajo a seguir. Al ser un proyecto en constante cambio, lo ideal es tener una biblia a la que acudir en caso de duda.
2. **Búsqueda de recursos:** Buscar recursos gráficos y sonoros que encajen con la temática del juego y, en caso de necesitarlo, adaptarlos.
3. **Fase inicial:** Programar el código de las clases necesarias para preparar una base inicial y probar las funcionalidades de las mecánicas.
4. **Prototipado:** Con las funcionalidades listas, se implementa el loop jugable.
5. **Implementar recursos:** Con el loop listo, se implementan los recursos sonoros y visuales para mejorar el *game feel* y lograr un aspecto *juicy*.
6. **Testeo:** En base al testeo, se subsanan errores y vuelve a testearse hasta conseguir el producto mínimo viable.

Información básica

Identidad del juego

Cuanto mayor es la población de clase alta, mayor es la polución.

Pilares del diseño

1. Arrastrar y soltar son las mecánicas básicas.
2. Juntar Pufs crea cosas.
3. La polución difumina la pantalla y mata a los Pufs.

Game Outline

Características

- Los Pufs aparecen esporádicamente en el tablero.
- Los Pufs de tipo *clase alta* al juntarse en parejas aumentan las probabilidades de que nazcan más Pufs de *clase alta*.
- Al juntarse seis Pufs de *clase alta*, se crea una empresa. Las empresas aumentan la polución.
- Al juntarse seis Pufs de *clase baja* mientras existan empresas, se crean piquetes.
- Al juntarse dos empresas cercanas entre sí, se crea un grupo empresarial.
- Al juntarse dos grupos empresariales, se crea un oligopolio. Los oligopolios triplican la polución.
- Al cabo de 5 minutos de partida, el paritorio abre sus puertas y te permite crear nuevos Pufs a cambio de dejar dos de ellos en él durante un tiempo: 90 segundos.
- Al juntarse dos piquetes, se crea un sindicato. Los sindicatos rebajan la polución un 10%.
- Los edificios sirven como almacenes de Pufs, albergando una cantidad fija dependiendo de la calidad del edificio. A mayor polución, menor calidad.
- Los Pufs de clase alta se mueven solos y se buscan entre sí.
- Conforme aumenta la polución, la visión del jugador se va emborronando y los Pufs comienzan a morir.
- Las estadísticas a mostrar al jugador son acumulativas:
 - *Población*: El número de Pufs que hay en el tablero en ese momento.
 - *Año*: El año actual.
 - *Polución*: La cantidad de polución, en %, que se ha generado. El porcentaje deriva de la cantidad de población existente, generando la *clase baja* menos polución que la *clase alta*.

Modos de juego

El modo de juego principal consistirá en evitar el aumento en la contaminación a través del control activo de la creación de empresas y oligopolios gracias a su contraparte: los sindicatos y piquetes. Si la contaminación acaba sobrepasando el 66% del límite, la pantalla se emborronará y el humo empezará a quitar visibilidad. Si llega al 100%, se escucharán gritos de auxilio, explosiones y terremotos y será el fin del juego.

Futuras implementaciones

Las siguientes funciones podrían implementarse dependiendo de la viabilidad del proyecto.

- Nuevos personajes temáticos como:
 - *Conspiracionistas*: Pufs de ambas clases que aletargan al resto de Pufs.
 - *Líderes sindicales*: Aumentan la facilidad de sindicalización del pueblo.
 - *Salvadores blancos*: Pufs de clase alta que transforman a Pufs de clase baja en clase media.
 - *Hostigadores*: De vez en cuando nacen hostigadores, trabajadores que llaman al levantamiento en contra de las empresas e inician revoluciones.
 - *Agentes de la paz*: Pufs de cualquier tipo de clase que reprimen a los Pufs de clase baja.
 - *Clase media*: Pufs de clase baja que ven mejorada su vida de algún modo y pueden ascender a clase alta.
 - *Usureros*: Conceden préstamos a los Pufs, pero, si no les pagan con grandes intereses, llaman a los agentes de la paz.
- Estadísticas únicas generadas a través de unos parámetros dados a una IA por cada minion como nombre, edad, altura o sueño.
- Mapa familiar con cada miembro de la familia.
- Característica jugable: Monedas. Lo ganan la clase baja con su sudor y esfuerzo y te permite comprar edificios.
- Nueva funcionalidad del minion: crecimiento. Dependiendo de factores como arte, o trabajo, los Pufs envejecen con mayor o menor rapidez.
- Nuevos aspectos para los Pufs.
- Las características como altura, constitución o salud afecta directamente a la apariencia del minion.
- Eventos climáticos sucedidos a raíz de la contaminación como lluvias, terremotos, huracanes y tornados.
- Nuevo modo de juego de supervivencia donde los Pufs padecen sed, hambre y deben recolectar recursos.
- Nueva dificultad donde debes aguantar el máximo de eras posibles como neandertal, bronce, hierro, acero, comunicación, etc: en cada era, la estética es distinta y la polución se incrementa.
- Cambio estético: posibilidad de cambiar la estética de color por otras paletas
- Traducción amateur al: inglés, euskera, gallego, valenciano y el panocho

Game Summary

El loop jugable es bastante sencillo:

1. Comienza el juego.
2. Un texto blanco aparece sobre una pantalla en negro: "Tienes X Pufs a tu disposición. Descubramos cuánto resistes al sistema"
3. El contador de población y polución comienzan a la vez, entre 5 y 10 segundos después del comienzo del juego.
4. El contador de año se inicia de inmediato, sumando 1 año por cada minuto de juego.
5. Los pufs comenzarán a aparecer cada 10 segundos hasta llegar al límite.
6. Una vez que comienzan a aparecer los Pufs, el jugador puede:
 - a. Arrastrar a los Pufs hacia los laterales del mapa para descartarlos
 - b. Juntarlos para que se creen grupos
7. Si la polución llega al 66%, la pantalla se emborrona y al llegar al 100% se pierde la partida.
8. El objetivo es aguantar el mayor número de años posible.
9. Las rondas no se pueden pausar ni guardar el progreso, cada partida es única.
10. Tras terminar la partida, aparecer un ranking de los jugadores con más años logrados

Sound FX / BSO

Efectos de sonido retro que recuerden a la época de Game Boy.

La BSO podría seguir la misma temática que los SFX, 16 bits. [Ejemplo](#).

Propuestas de valor

Exploración de las disparidades sociales en relación con la contaminación y el cambio climático.

Simulación y gestión socioambiental orientadas hacia la sostenibilidad.

Concienciación socioambiental

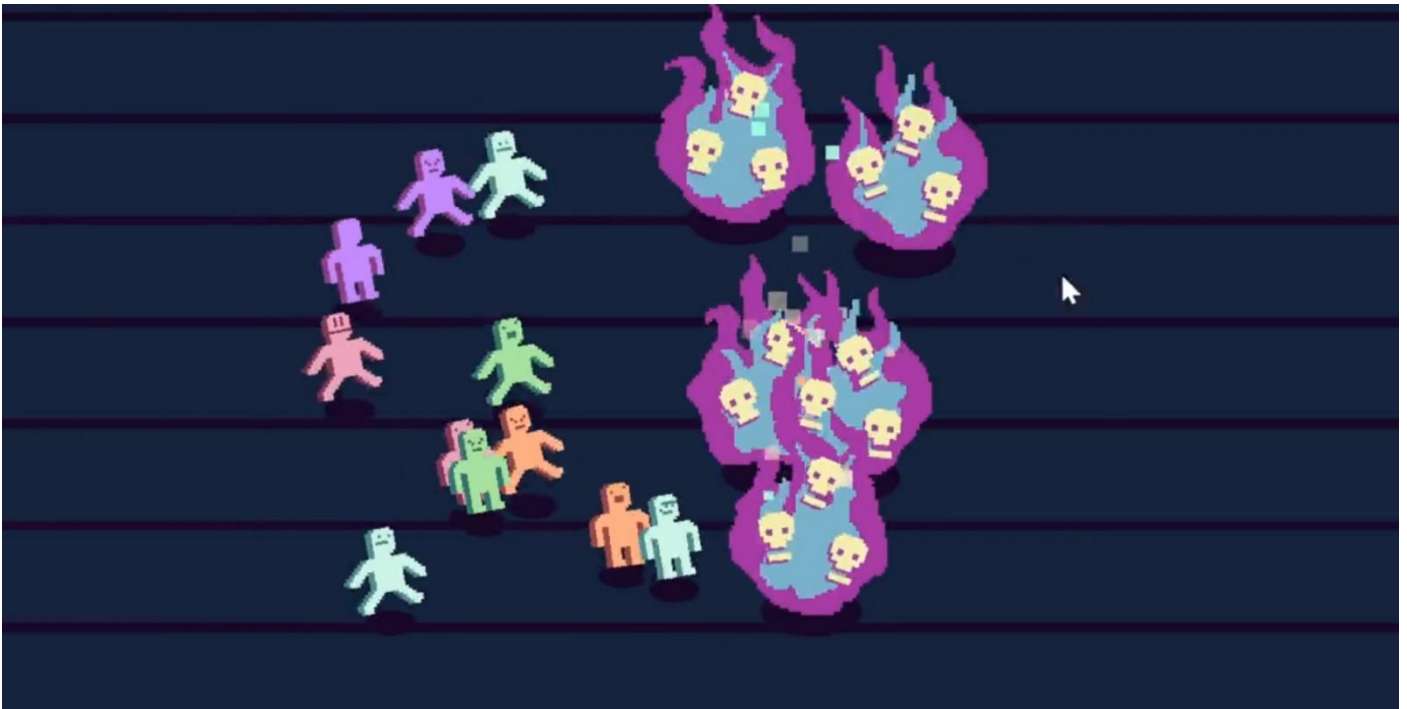
Estilo de arte

El estilo de arte sería una mezcla entre estas dos capturas, pero aplicando shaders de suavizado e iluminación como la tercera.

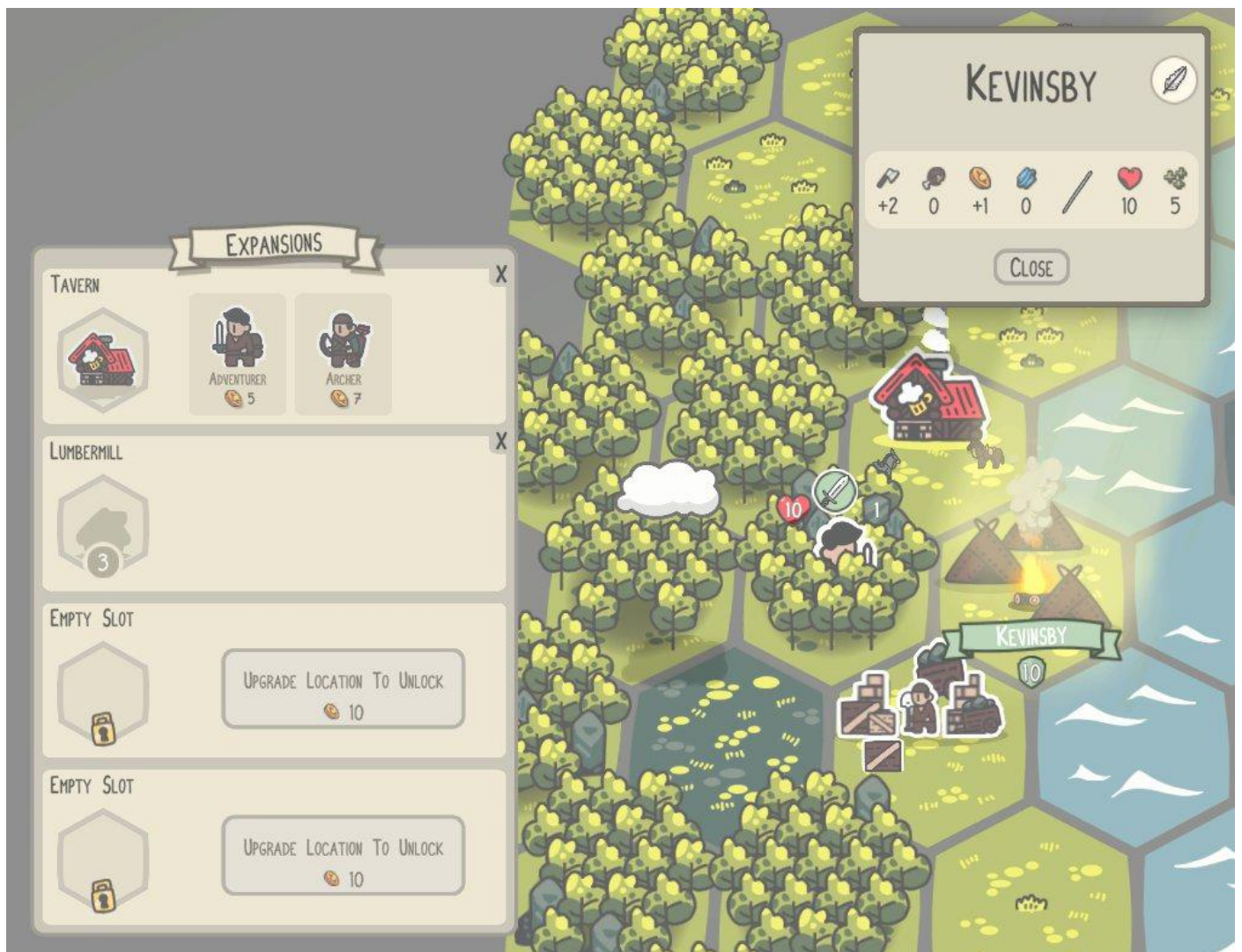
1.



2.



3.



Paleta de colores

- **Gb Noods:** <https://lospec.com/palette-list/gb-noods>



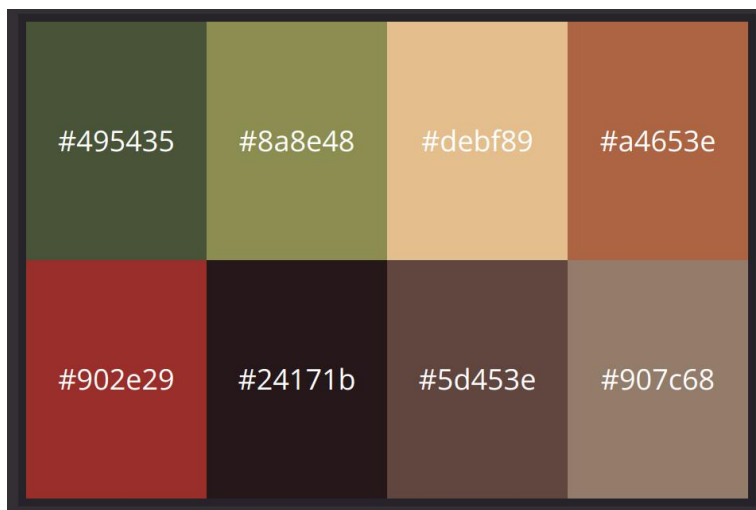
EXAMPLES



- **Retrocal 8:** <https://lospec.com/palette-list/retrocal-8>

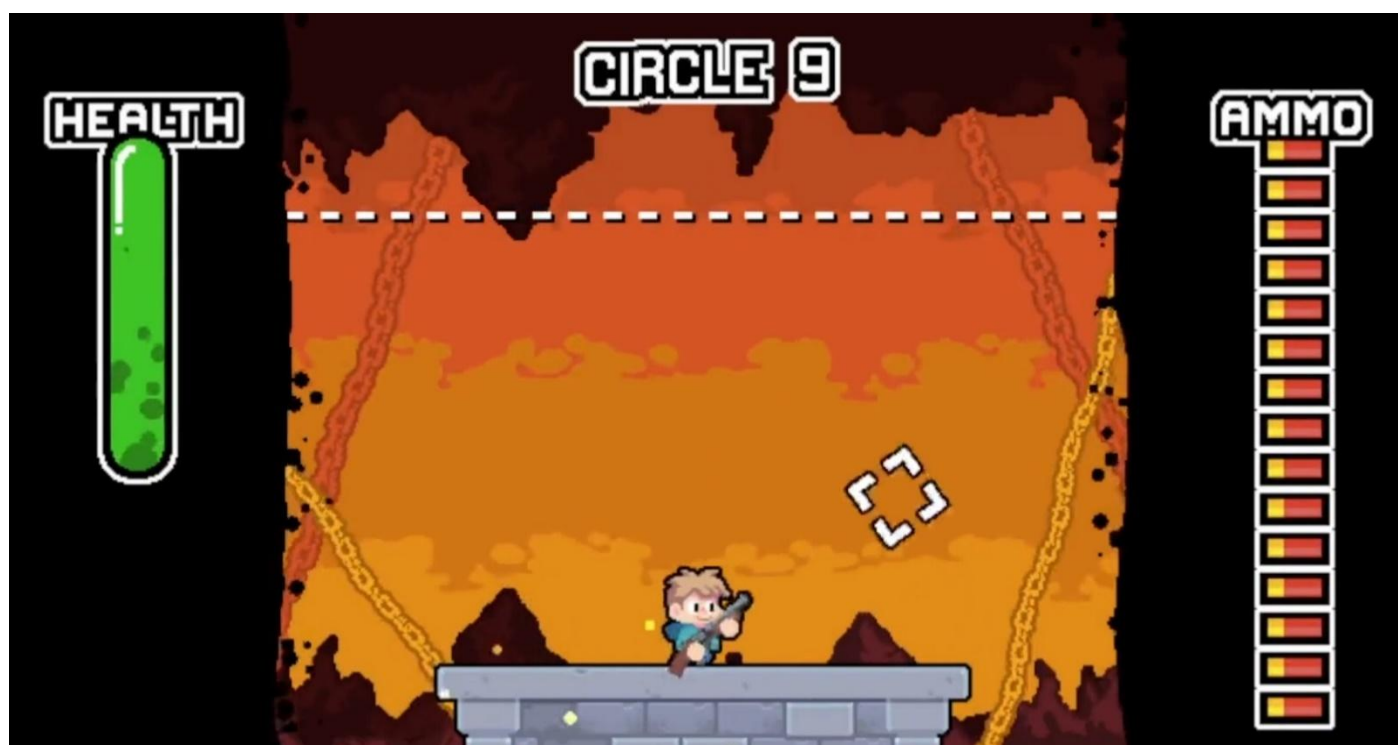


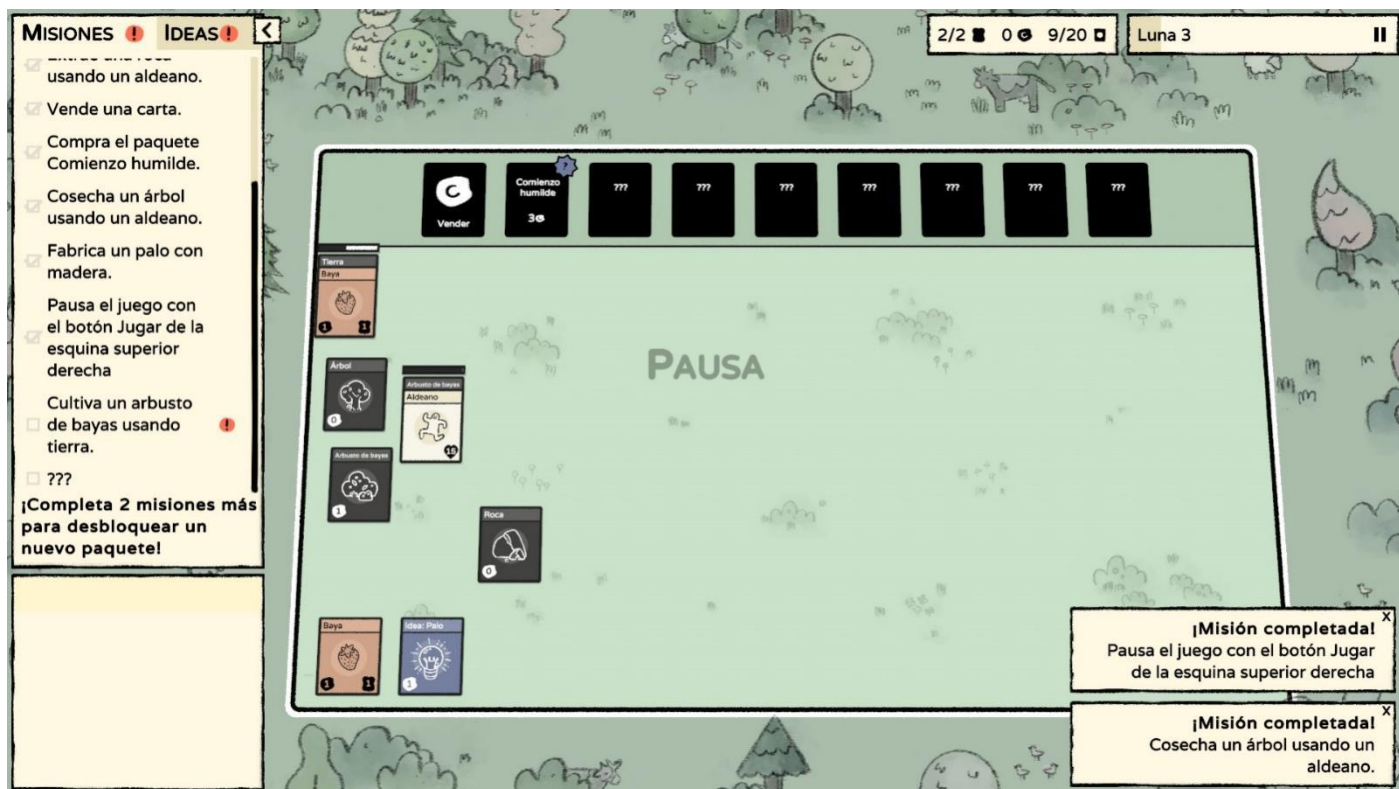
- **Dead Weight-8:** <https://lospec.com/palette-list/dead-weight-8>



Interfaz

Sencilla, en la parte superior de la pantalla de juego. Al estilo de:





Similar Products

Aunque hay evidentes diferencias, algo similar sería:

La dificultad en la gestión de los recursos en [Oxygen Not Included](#).

La estética y simplicidad de [Mistward](#).

Hoja de ruta del desarrollo

Plataformas: Steam/Google Play/Web/ItChio

Audiencia (target): El proyecto está orientado a los nichos de juegos de gestión de recursos, y el nicho de platinar videojuegos que buscan un reto.

Hitos a lograr

Hito	Tipo	Definición	Fecha
1.	Entorno	Implementación del mapa y los límites	
2.	Entorno	Implementación de los Pufs	
3.	Entorno	<i>Animación:</i> Movimiento de los Pufs por el mapa	
4.	Entorno	Implementar cámara isométrica	
5.	Entorno	Implementación de interfaz gráfica	
6.	Mecánica	Implementar una rejilla de posicionamiento donde colocar los pufs y los edificios	
7.	Mecánica	Generación de Pufs cada X segundos	
8.	Mecánica	Alternancia entre clases sociales en los Pufs por porcentajes 50%	
9.	Mecánica	Conteo de estadísticas: Polución, año, población.	
10.	Mecánica	<i>Control:</i> Arrastrar y soltar	

11.	Mecánica	<i>Control:</i> Aplastar
12.	Mecánica	<i>Control:</i> Lanzar
13.	Mecánica	<i>Unión:</i> De los Pufs de clase alta para crear empresas
14.	Mecánica	<i>Unión:</i> De los Pufs de clase baja para crear piquetes
15.	Juicy	Implementación del audio (BSO y SFX)
16.	Juicy	Implementación de las animaciones
17.	Juicy	Mejora en la animación del movimiento de los Pufs
18.	Juicy	Mejorar el movimiento de los objetos de Jittery a Smooth
19.		
20.		

Terminología Git

Ramas

<u>Rama</u>	<u>Definición</u>	<u>Fecha</u>
main	Rama principal. Después de cada commit a main, se elimina la rama con los cambios (a excepción de development)	
development	Rama principal donde se prueban las nuevas feature tras desarrollarlas. Si no hay ningún problema, se mergea a producción	
features/	Rama para desarrollar nuevas funcionalidades	
bugFixes/	Rama para corregir errores	
general/	Rama para todo lo que no esté en las anteriores categorías	

Commits

Después de cada nomenclatura, una breve descripción de lo commiteado.

<u>Nomenclatura</u>	<u>Definición</u>	<u>Fecha</u>
[Feature]	Pequeña descripción de la funcionalidad	
[Fix]	Nombre del bug	
[Refactor]	Con la clase refactorizada	
[Visual]	Con el elemento visual que se ha retocado o añadido. Esto aplica únicamente a la parte artística	
[Doc]	Si se ha actualizado la documentación	
[UI]	Para los cambios relacionados con la UI	
[Test]	Para el testeo de clases, pruebas unitarias, etc	
[Revert]	Por si se ha necesitado eliminar algún commit ya subido a origin	