

El team

Franco Kumichel
Gustavo Lopez
Lautaro Castro



grupo 02

Problemática

- Juego
 - Limpieza del caballo
- Objetivo
 - Aprendizaje
- ¿Por qué?
 - Primeras interacciones

Estado inicial

¿Cómo empezamos el desarrollo la aplicación?

- Tecnología a utilizar
- Mecánica y flujo general del juego
- Organización interna

Tecnología a utilizar

- Jetpack Compose
 - Herramienta moderna
 - Interfaces con un diseño moderno y práctico
- Views
 - Tecnología clásica
 - Antigua pero con más herramientas
- Librerías de terceros
 - Utilizadas para problemas específicos (al menos a primera vista)
 - Room database
 - Etc...

Tecnología a utilizar

Jetpack Compose

- Más amigable para el desarrollo
- Juego simple y sin mecánicas complicadas de implementar.
- Aprendizaje
 - Aprender de la documentación en general
 - Cursos de Google
 - Mandarse de lleno a implementar

Mecánica y flujo gral del juego

- Cómo es la dinámica paciente/terapeuta?
- Cómo va a ser la limpieza del caballo?
- Cuántas etapas?
- Cómo presentar las herramientas?
- Puntuación
- Tiempo
- Dificultad, etc...

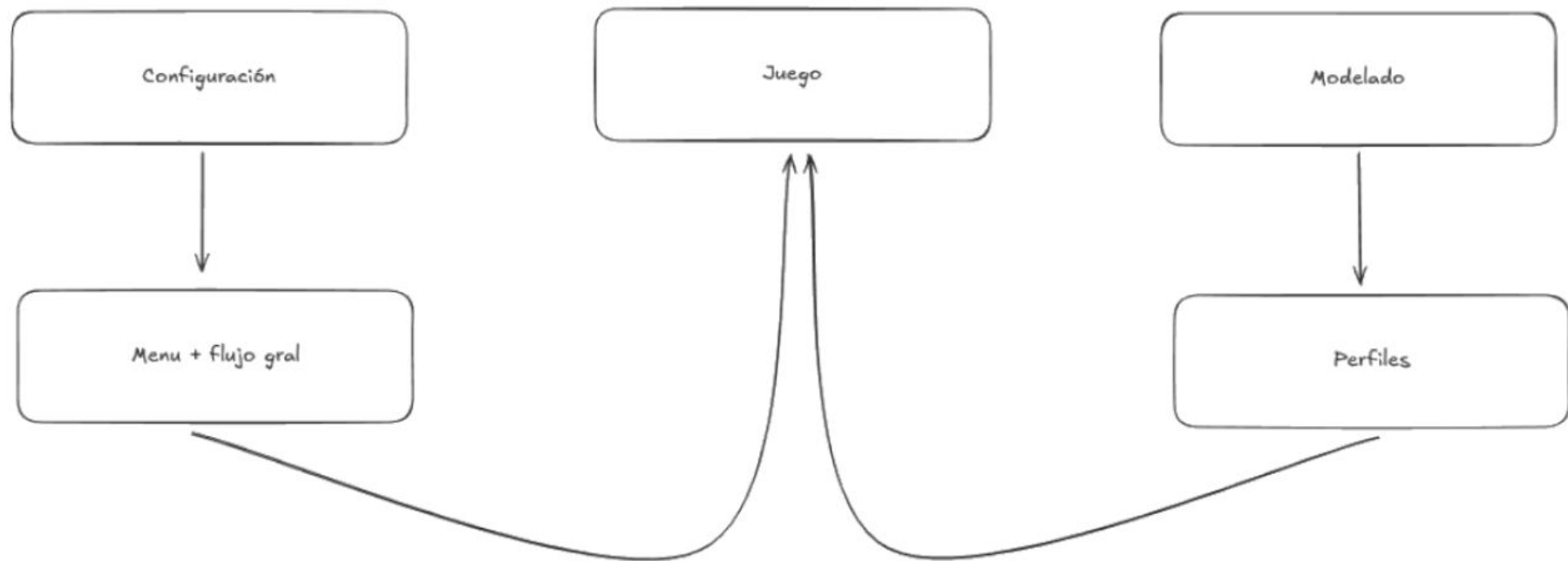
Mecánica y flujo gral del juego

- Cómo es la dinámica paciente/terapeuta?
 - Sin autenticación, el control lo tiene el terapeuta, acompañamiento constante.
- Cómo va a ser la limpieza del caballo?
 - Estilo inspirado en Pou, facilidad de uso e intuitivo.
- Cómo se pasa de etapa?
 - Selección de parte correcta y limpieza.
- Cómo presentar las herramientas?
 - Barra de herramientas no invasiva.

Organización interna

- Componentes principales del juego:
 - Juego en sí
 - Menú
 - Perfiles
 - Configuración
 - Estadísticas
- Problema
 - El juego en sí es el más complejo

Cómo resolvimos



Nuestro enfoque

A los problemas anteriores, tratamos de manejarlos siguiendo estas pautas:

- Simplicidad -> Reducir la complejidad siempre que sea posible, vamos a lo más simple.
- Vista del usuario -> Siempre tomando en cuenta como sería el caso de uso, que sea intuitivo y fácil de usar.

Problemas técnicos

Principales desafíos y cómo resolvimos

- Doble interfaz.
- Selección
- Limpieza

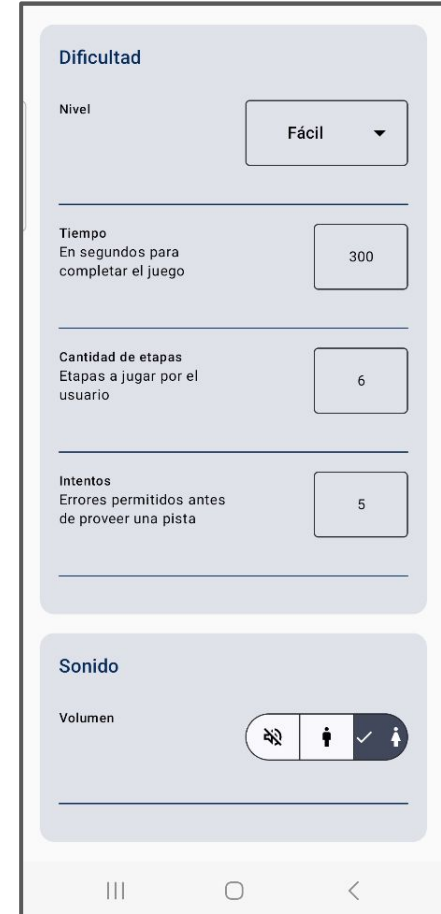
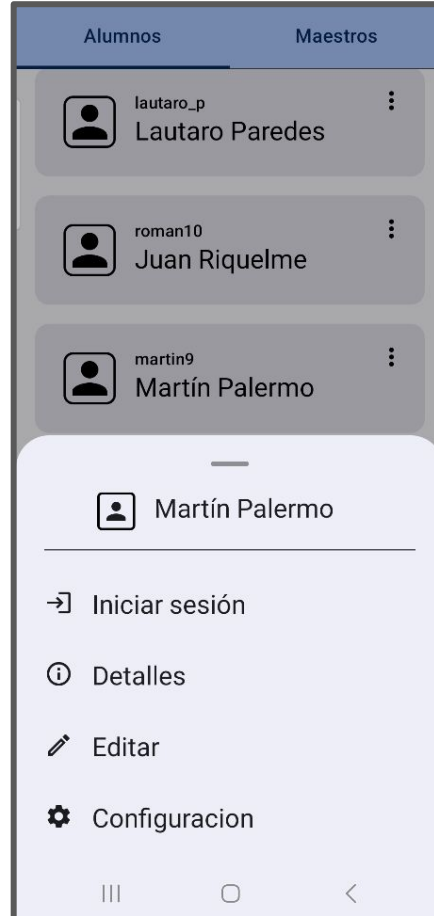
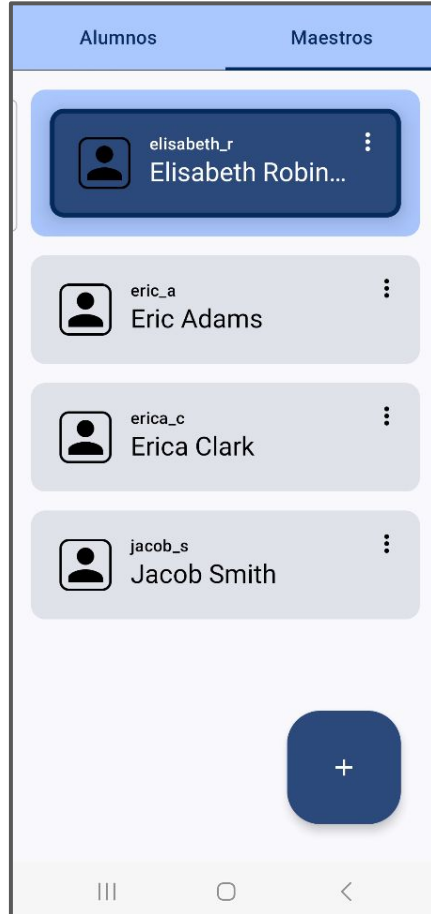


Doble interfaz

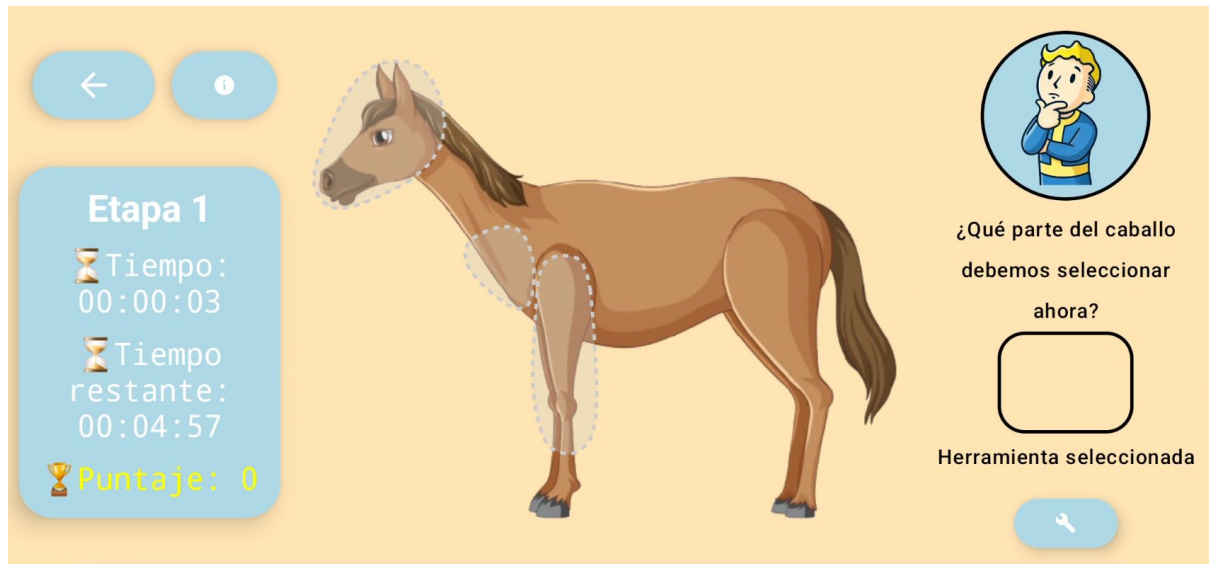
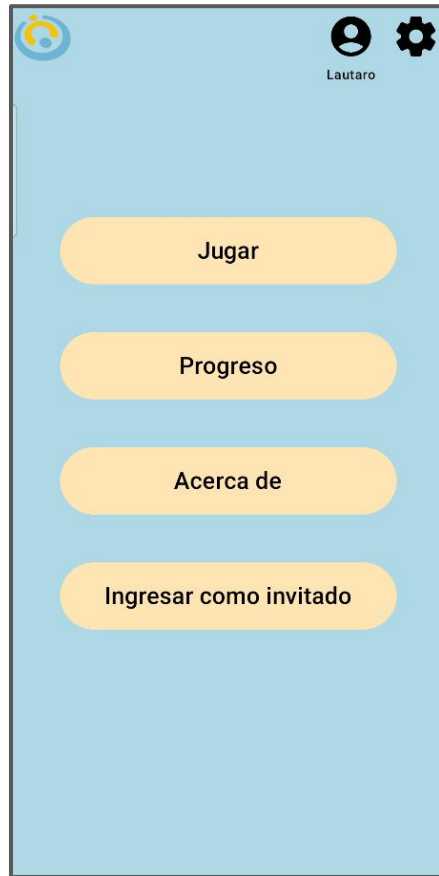
A nivel de diseño la aplicación presenta dos partes

- Gestión de usuarios
 - Creación, borrado, edición
 - Personalización de Configuración
 - etc
- El juego en sí mismo
 - Las mecánicas y su lógica asociada
 - La vista o pantalla de inicio

Interfaz de gestión de usuarios

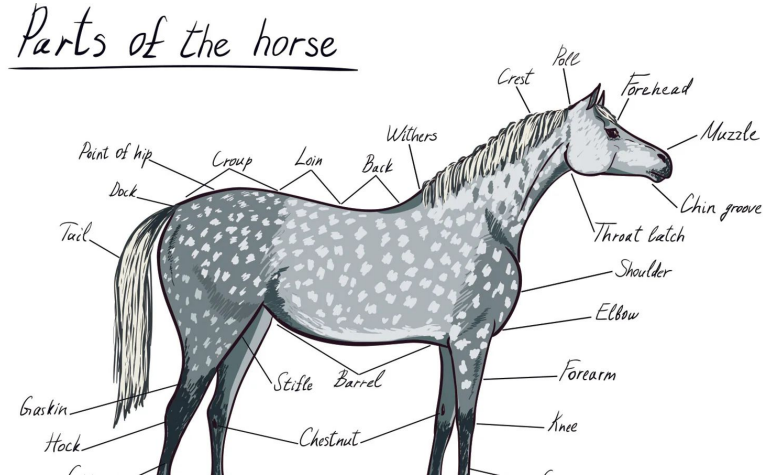


Juego y pantallas asociadas



Partes a limpiar

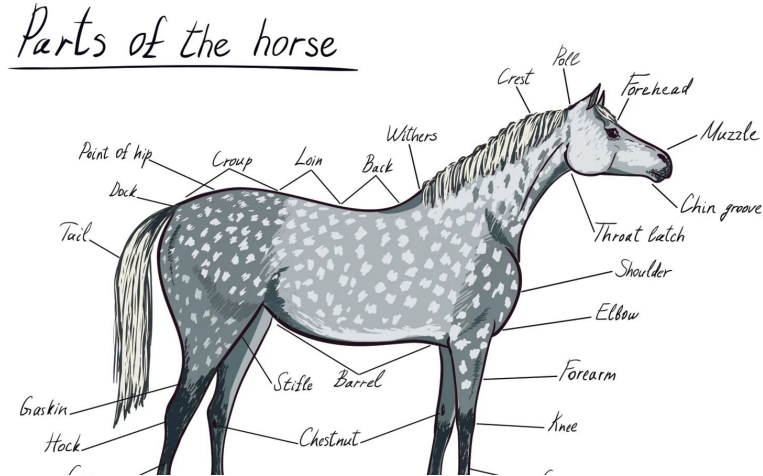
- 13 etapas



- Muchas partes -> confuso, como hacemos?

Partes a limpiar

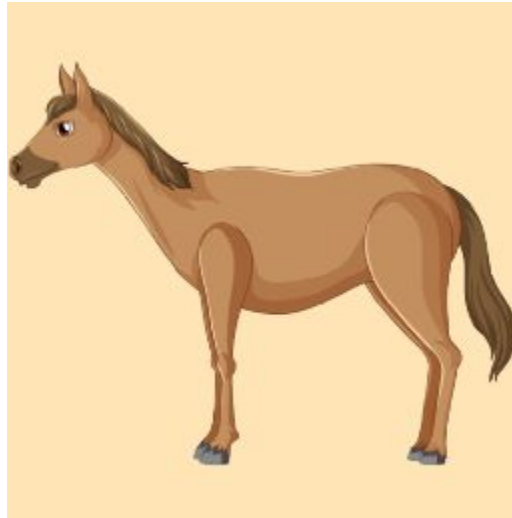
- 13 etapas



- Muchas partes -> confuso, como hacemos?
- Selección de partes

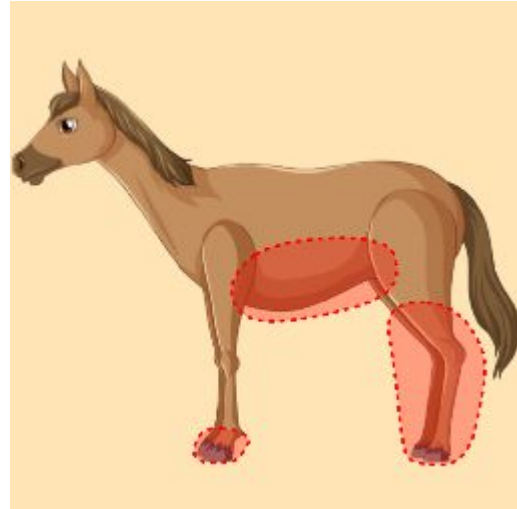
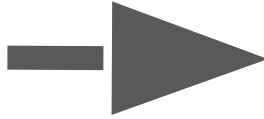
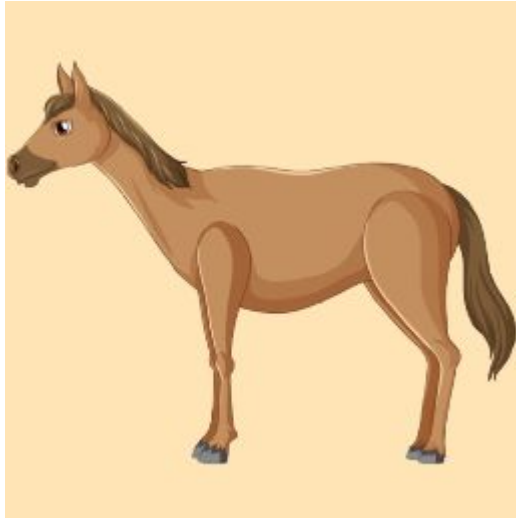
Selección de partes

- Selecciono una parte, limpio, paso de etapa, repito.
- Cómo indicamos?
- Sin dar la correcta, sin ser invasivos.



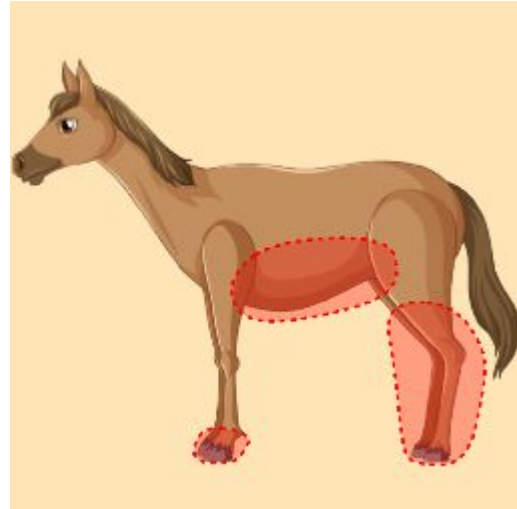
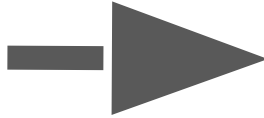
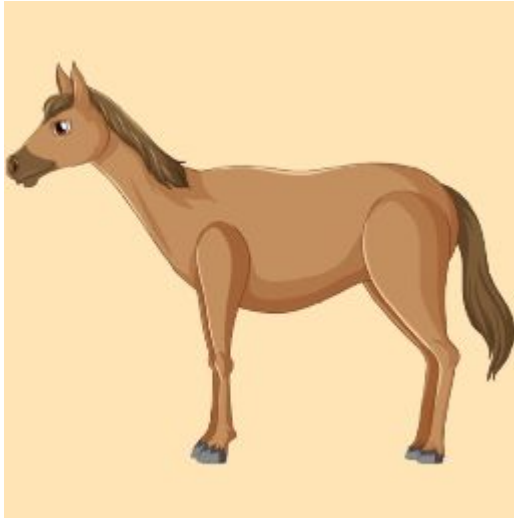
Selección de partes

- Selecciono una parte, limpio, paso de etapa, repito.
- Cómo indicamos?
- Sin dar la correcta, sin ser invasivos.



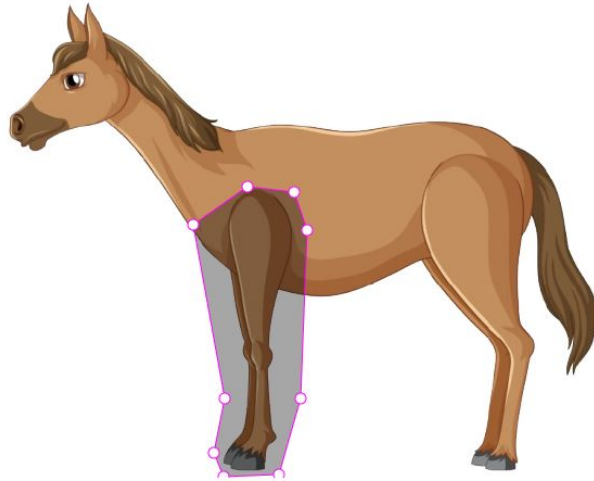
Implementación

- Cómo lo hacemos más llamativo?
- Cómo dibujamos en pantalla esto?



Implementación

- Polígonos -> simple, fácil de ver, intuitivo, llamativo
- Herramienta para crear polígonos
- Forma flexible de agregar, modificar o quitar partes -> Enums y listas.



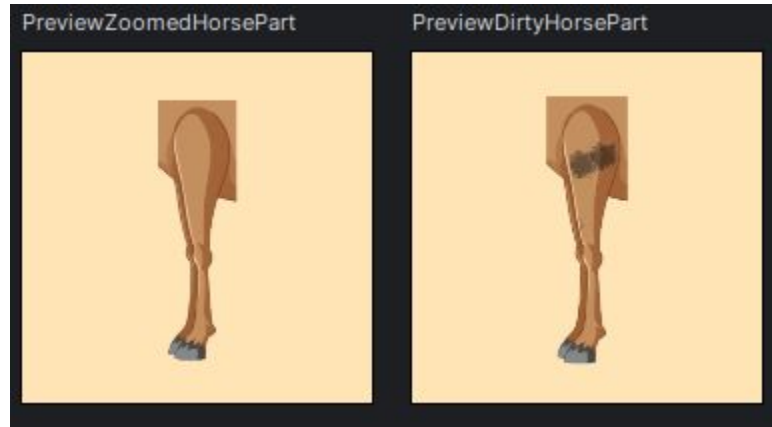
Selección de partes

- Seleccionamos la parte, es viable mostrar el caballo entero para limpiar?
- Teniendo en cuenta bajas dimensiones.



Selección de partes

- Hacemos un “Zoom”
- Mostramos partes enteras específicas en vez del caballo.
- Manipulación más fácil de la herramienta, la suciedad se ve mejor, menos distracciones.



Limpieza del caballo

Dos grandes dilemas:

- ¿Cómo se valida que se está realizando la limpieza?
- ¿Cómo reconoce el usuario que está realizando la limpieza de manera correcta?

Limpieza del caballo

- La limpieza se realiza arrastrando la herramienta hacia la parte ampliada del caballo
- En la parte ampliada del caballo, establecemos un “área de limpieza”
- La validación surge de saber si la herramienta está dentro del área de limpieza.
- Problema: comparar posición de la herramienta con posición del área de limpieza
- Solución: llevar ambos a una situación en la que sus posiciones sean comparables mediante cálculos

Limpieza del caballo

Tenemos la validación, ¿Como sabe el usuario que está realizando de forma correcta la limpieza?

- Refuerzo auditivo: al arrastrar la herramienta sobre la zona de limpieza, se emite un sonido de cepillado
- Refuerzo visual: a medida que el usuario realiza la limpieza, la mancha de suciedad va desapareciendo de a poco

Cerrando

Y a la demo

Conclusiones finales

- Tiempo
- Lo principal está: Mejor diseño del caballo y de las partes zoomeadas
- Mejoras en el flujo general del juego
 - Ampliar aspectos organizativos
 - Ampliar configuraciones personalizables
- Refactoring, eficiencia del juego
- Motor de juego para el juego en sí?