



E0: Diseño del juego y Mockup

Elías Ayaach, Jorge Jacque y Mario Pedemonte - SmilingFriends

Saboteur

1. Las características básicas del juego que implementará, incluyendo la definición del objetivo de su juego, lo que hará.

El saboteur-online consiste en un juego de estrategia e intriga donde se dividen los jugadores en 2 bandos: mineros y saboteadores. La meta de los mineros consiste en llegar al oro antes de que se acaben las cartas disponibles en el mazo, mientras que la de los saboteadores es, como es de esperar, evitar que estos logren ese cometido. Para desarrollar este sistema ambos equipos cuentan con un set de herramientas que permitirán, ya sea para encontrar el oro, destruir caminos o incluso bloquear a otros jugadores.

El juego va a permitir un máximo de 5 jugadores. Si es de a 5 o 4 van a haber 2 saboteurs y el resto serán mineros. El juego como mínimo debe tener 3 jugadores por partida, y en este caso habrá un solo saboteur. Durante el desarrollo de este se van lanzando cartas para ir formando caminos hasta la meta. Existen 5 tipos de cartas en el juego: Cartas de camino, cartas de derrumbe, cartas de mapa, cartas de bloqueo y de desbloqueo. El funcionamiento de estas será detallado más adelante, junto al funcionamiento general del juego.

2. El comportamiento de su aplicación de manera detallada, describiendo las funcionalidades que se le presentarán a los **diversos tipos usuarios** (casos de uso) y lo que debe realizar el usuario para interactuar y obtener los resultados esperados (comprometidos) de su funcionalidad en específico.

- El usuario accede a la página por primera vez sin tener una cuenta en ella. En este momento el usuario solo puede ver las reglas y el “acerca de” de la página. Se insta al jugador a que se registre para jugar
- El usuario se registra en la página, añadiendo correo electrónico, nombre de usuario y contraseña. Terminado el registro, ahora tiene la posibilidad de presionar el botón jugar y acceder al registro de partidas en espera.
- El usuario registrado puede solicitar unirse a alguna de las partidas. También tiene la posibilidad de crear su propia sala.
- Si el usuario decide unirse a una partida creada, debe esperar a que el creador lo acepte.
- Si es aceptado, la partida empezará cuando el creador lo decida.
- Si el usuario decide crear su propia partida, este tiene la capacidad de aceptar o rechazar solicitudes de unirse de otros usuarios y puede iniciar la partida.
- Una vez la partida inicia se deciden los roles de cada jugador al azar, y también se decide el orden de juego de forma aleatoria
- Mientras el jugador no esté en su turno no puede realizar ninguna acción.
- Una vez el jugador está en su turno, éste puede seleccionar una carta y utilizarla, dependiendo de su tipo, este podrá seleccionar una casilla de juego o a un jugador.



- Una vez el juego termina, por sus propias, cada jugador recibirá una puntuación en relación a el resultado, y tendrá la opción de volver al registro de partidas.

3. Las reglas que gestionará el servidor para tratar cada interacción (descripción de la mecánica de las reglas, *nada de programación*)

-A cada jugador se le asigna una clase al azar. Esta clase puede ser “saboteur” o “minero”.

-Los jugadores jugarán por turnos, los cuales serán decididos al azar.

-El tablero está compuesto por una matriz de 5x7 casillas. En la casilla (5,4) estará ubicado el inicio de los jugadores, mientras que en las casillas (1, 1), (1, 3) (1, 5) serán los 3 puntos donde puede estar el oro.

-Cada jugador recibe 4 cartas al azar y siempre tiene que mantener este número de cartas, por lo que cada turno debe botar o usar una carta, y luego obtener otra del mazo.

-La ubicación del oro en las coordenadas finales es decidida al principio del juego al azar.

-Existen 5 tipos de cartas de juego:

- Carta de bloqueo.
- Carta de camino.
- Carta de mapa.
- Carta de derrumbe
- Carta de desbloqueo

-La carta de bloqueo permite bloquear a cualquier jugador si este no estaba bloqueado previamente. Existen 3 tipos de cartas de bloqueo:

- Bloqueo de l interna
- Bloqueo de pica
- Bloqueo de carreta

-La carta de desbloqueo permite desbloquear a cualquier jugador si este no estaba bloqueado previamente. Existen 3 tipos de cartas de bloqueo:

- Desbloqueo de l interna
- Desbloqueo de pica
- Desbloqueo de carreta
- Desbloqueo doble

-En las casillas libres se ubicarán 5 objetos sorpresa al azar. Estos entregarán alguno de los siguientes bufos al azar, al jugador que construya un camino sobre esta.

- Objeto de turno doble
- Objeto de inmunidad a bloqueo /tipo/ (puede ser l interna, pica o carreta)

-El objeto de turno doble permite al jugador utilizar una carta adicional.



-El objeto de inmunidad a bloqueo impide de manera indefinida que un jugador sea bloqueado por una carta de bloqueo de un tipo en particular.

-Si un jugador intenta bloquear a un jugador con inmunidad a bloqueo |tipo| con una carta de bloqueo |tipo|, este falla al bloquear y pierde su turno. (|tipo| puede ser linternas, pica o carreta)

-Solo es posible bloquear a un jugador si es que no estaba bloqueado previamente.

-Solo se puede desbloquear a un jugador si este está bloqueado.

-Solo se puede desbloquear a un jugador con una carta de desbloqueo cuyo tipo coincida con el de la carta de bloqueo utilizada para tal.

-Un jugador bloqueado solo puede botar cartas y desbloquearse a sí mismo.

-Las cartas de camino solo se pueden ubicar en casillas libres, siempre y cuando exista alguna casilla aledaña con un camino compatible.

-2 caminos son compatibles si alguna de las entradas del primero se puede conectar con las entradas del segundo.

- Los caminos tienen 4 entradas, una para cada dirección: arriba, abajo, izquierda y derecha.

-La carta de mapa permite a un jugador revisar a elección alguna de las cartas finales, y anunciar a los demás si está o no el oro en ella.

- La carta de derrumbe solo puede utilizarse sobre casillas que contengan caminos construidos por los jugadores.

- El juego termina cuando los jugadores logran construir un camino hacia el oro o cuando el mazo se queda sin cartas.

- Los “saboteurs” ganan si el mazo se queda sin cartas.

- Los “mineros” ganan si se llega al oro.

- Los jugadores no pueden realizar ninguna acción si no están en su turno.



4. Un primer borrador de su protocolo de comunicación. Esto es, una definición del objeto y la información se debe mandar para que el servidor procese la jugada

El protocolo que se utilizará durante el desarrollo del juego facilitará el procesamiento de las jugadas y asegurará la comunicación eficiente entre cliente y servidor. También se considerarán las validaciones y el manejo de errores.

Protocolo cliente

Cada movimiento que haga un jugador enviará al servidor un objeto con la siguiente información:

```
{  
    "id_jugador": int,           // identificador único del jugador  
    "tipo_carta": string,       // tipo de carta que se está jugando  
    "estado_jugador": string,   // estado del jugador ("bloqueado" o "no bloqueado")  
    "accion": string,           // acción realizada ("mover", "atacar", "usar carta", "bloquear",  
    "desbloquear")  
  
    "coordenadas": {  
        "x": int,                // coordenada x de la acción  
        "y": int                 // coordenada y de la acción  
    },  
    "jugadores_afectados": [int], // lista de IDs de jugadores afectados (si aplica)  
    "cartas_en_mano": [string],  // lista de cartas en mano después de la acción  
    "timestamp": string,         // marca de tiempo de la acción  
    "turno_actual": int          // número de turno actual en el juego  
}
```

Descripción de los campos - cliente

- 1) id_jugador:

Tipo: int

Descripción: Identificador único asignado a cada jugador al iniciar sesión. Este ID se mantiene constante a lo largo de la sesión.

- 2) tipo_carta:

Tipo: string

Descripción: Especifica el tipo de carta que el jugador está utilizando. Puede ser uno de los siguientes valores:

"bloqueo"
"camino"
"mapa"
"derrumbe"
"desbloqueo"

- 3) estado_jugador:



Tipo: string

Descripción: Indica si el jugador está actualmente bloqueado o no. Los valores posibles son "bloqueado" o "no bloqueado".

4) accion:

Tipo: string

Descripción: Define la acción que el jugador está realizando. Ejemplos de acciones incluyen:

"mover": Moverse a otra casilla.

"atacar": Atacar a otro jugador.

"usar carta": Usar una carta de su mano.

"bloquear": Usar una carta de bloqueo.

"desbloquear": Usar una carta de desbloqueo.

5) coordenadas:

Tipo: object

Descripción: Contiene las coordenadas (x, y) donde se realiza la acción.

Campos:

x: int (coordenada horizontal).

y: int (coordenada vertical).

6) jugadores_afectados:

Tipo: array of int

Descripción: Una lista de IDs de jugadores que se ven afectados por la acción (por ejemplo, si se utiliza una carta de bloqueo).

7) cartas_en_mano:

Tipo: array of string

Descripción: Una lista de los tipos de cartas que el jugador tiene en su mano después de realizar la acción. Esto ayuda al servidor a actualizar el estado de las cartas.

8) timestamp:

Tipo: string

Descripción: Marca de tiempo en algún formato específico, por ejemplo, ISO 8601 (ej. "2024-09-2T15:00:00Z") que indica cuándo se realizó la acción. Esto es útil para el registro y la auditoría de las acciones.

9) turno_actual:

Tipo: int

Descripción: Número de turno actual en el juego. Permite al servidor verificar que el jugador está actuando en su turno correcto.

Protocolo servidor

```
{  
  "status": string,           // estado de la respuesta ("exito", "error")  
  "mensaje": string,         // mensaje descriptivo del resultado
```



```
"id_jugador": int,           // ID del jugador que realizó la acción
"estado_juego": {             // estado actual del juego
    "turno_actual": int,      // número de turno actual
    "jugadores": [            // lista de jugadores con su estado
        {
            "id_jugador": int, // ID del jugador
            "estado": string, // estado del jugador ("bloqueado", "no bloqueado", "ganador", etc.)
            "cartas_en_mano": [string] // cartas en mano del jugador
            "num_cartas_mazo_juego": [int] // representa el número de cartas que quedan en la partida
        }
    ],
    "tablero": [                // estado del tablero
        [                      // Fila 1
            {
                "tipo": string, // tipo de carta o "vacío" en la casilla
                "jugador_ocupante": int // ID del jugador que ocupa la casilla (si aplica)
            }
        ],
        [
            "objetos_sorpresa": [ // lista de objetos sorpresa en el tablero
                {
                    "coordenadas": { // coordenadas del objeto
                        "x": int,
                        "y": int
                    },
                    "tipo": string // tipo del objeto sorpresa
                }
            ]
        ]
    ]
}
```

Descripción de los campos - servidor

1) status:

Tipo: string

Descripción: Indica el resultado de la acción. Puede ser "éxito" o "error".

2) mensaje:

Tipo: string

Descripción: Mensaje descriptivo que explica el resultado de la acción. Proporciona información adicional para el jugador.

3) id_jugador:

Tipo: int

Descripción: ID del jugador que realizó la acción. Permite identificar quién realizó la jugada.

4) estado_juego:



Tipo: object

Descripción: Contiene la información actualizada sobre el estado del juego, incluyendo el turno, el estado de los jugadores y el tablero.

5) turno_actual

Tipo: : int

Descripción: Número de turno actual en el juego.

6) jugadores

Tipo: : array of object

Descripción: Lista de jugadores con su estado actual

id_jugador: int - ID del jugador.

estado: string - Estado del jugador ("bloqueado", "no bloqueado", "ganador", etc.).

cartas_en_mano: array of string - Lista de cartas que el jugador tiene en mano.

tablero: array of array of object - Representación del estado del tablero:

Cada objeto representa una casilla en el tablero, con:

tipo: string - Tipo de carta en la casilla (ej. "camino", "bloqueo", "vacío").

jugador_ocupante: int - ID del jugador que ocupa la casilla (si aplica).

objetos_sorpresa: array of object - Lista de objetos sorpresa en el tablero:

7) coordenadas: object - Coordenadas del objeto.

x: int - Coordenada x.

y: int - Coordenada y.

tipo: string - Tipo del objeto sorpresa (ej. "turno doble", "inmunidad").

Validaciones

El servidor debe llevar a cabo las siguientes validaciones al recibir un mensaje:

- Turno: Verificar que el jugador esté actuando en su turno.
- Estado: Comprobar que el estado del jugador permite realizar la acción (por ejemplo, no puede jugar si está bloqueado).
- Acción Legal: Confirmar que la acción solicitada es legal según las reglas del juego.
- Disponibilidad de la Carta: Asegurarse de que el jugador tiene la carta que intenta usar.

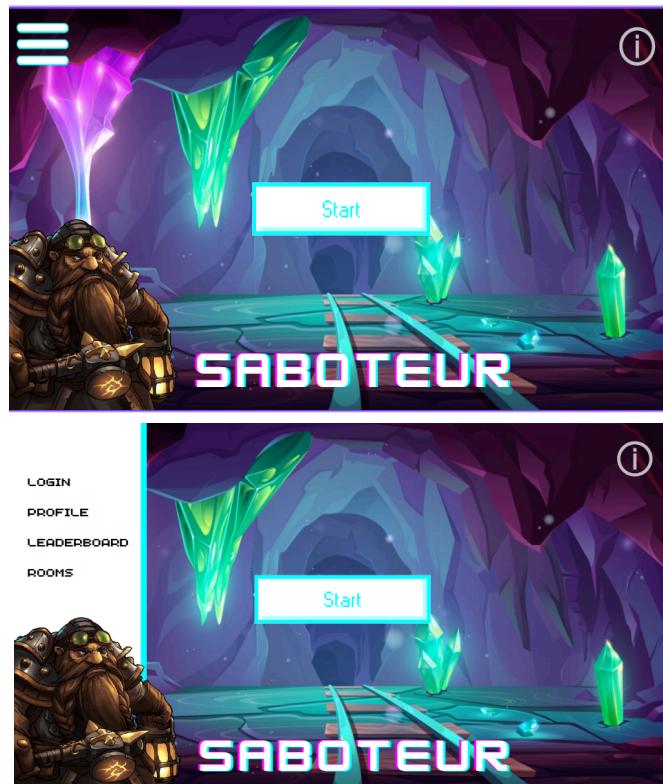
La estructura del mensaje de error es la siguiente:

```
{  
  "error": {  
    "codigo": int,           // código de error (ej. 400 para bad request)  
    "mensaje": string,      // descripción del error  
    "id_jugador": int       // ID del jugador que intentó realizar la acción  
  }  
}
```

Conexión



- Persistencia de Sesión: Es importante que los jugadores mantengan una conexión estable para evitar pérdidas de información durante el juego. En caso de desconexión, el jugador debe tener la opción de reconectarse y recuperar su estado.
 - Timeout: Se debe implementar un tiempo de espera en el servidor para evitar que los jugadores permanezcan inactivos indefinidamente.
5. El diseño de la página “landing page”, esta es la página de inicio (el index.html) que se llega por defecto cuando se va a la dirección WEB de su aplicación. Este tipo de páginas tienen típicamente un “acerca de...”, las reglas de operación o juego, y otra información de carácter general.



6. Deberán considerar el comportamiento para un usuario registrado y para otro no registrado.

Jugadores No registrados: son capaces de ver un leaderboard y ver cuántas están en proceso. También tiene la capacidad de ver el acerca de, las reglas y registrarse si es que lo desean.

Jugadores Registrados: Pueden acceder a partidas públicas y/o privadas. Pueden crear partidas. Van a tener un proceso personal con logros que permitirán modificar su ícono de jugador, y otras atribuciones (diseño de sus cartas etc). Dependiendo de su rango/puntos aparecerá en el leaderboard.

7. Para los usuarios registrados deberán considerar **varios perfiles y roles diferentes**, por Para el juego consideramos los siguientes tipos de perfiles:



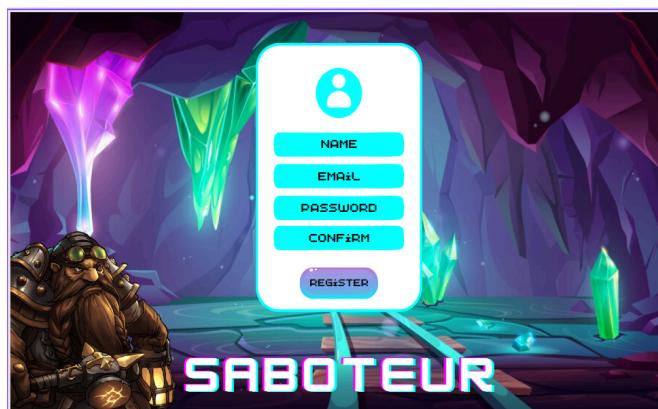
Usuario administrador: es capaz de eliminar otros usuarios o eliminar partidas en curso así como también tiene acceso a todos los beneficios por recompensas.

Creador de partida: este usuario tiene la capacidad de pausar o terminar la partida que creó y en la que se encuentra jugando.

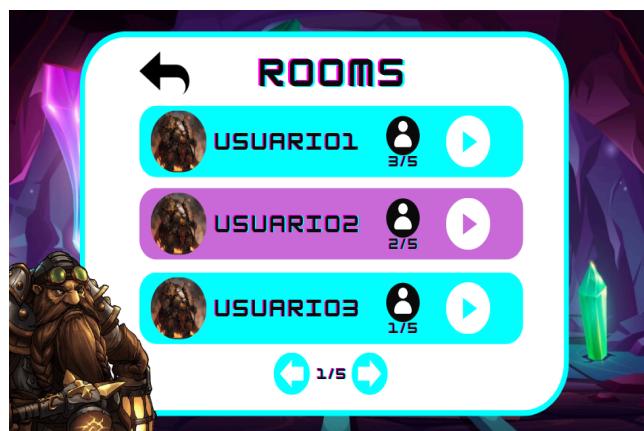
Usuario invitado: este jugador solo es capaz de jugar dentro de la partida y como máximo puede solicitar que el creador pausé la partida.

Beneficios por puntaje: las atribuciones de los perfiles van aumentando con el puntaje por ganar partidas. Ganar partidas suma una cantidad de puntos a tu perfil y perder resta. A medida que un usuario va ganando puntos puede desbloquear skins o tableros personalizados que todos lo verán.

8. Un mockup para el registro de un usuario



9. El registro para comenzar o integrarse a un juego. En este caso la lógica que seguirá el registro de usuario/jugador a un juego en particular (no es el registro de un usuario como “persona” en el sitio) y el saber cómo se unirán a otros participantes ¿cómo se “juntan” varios usuarios para el uso de la aplicación? ¿Hay algún timeout?



Cada partida cuenta con una sala de espera. Desde la perspectiva del creador de la partida, este puede ver una lista de los usuarios que desean unirse, junto con botones para aceptar o rechazar



cada solicitud. Por otro lado, los usuarios que intentan unirse verán el mensaje en pantalla que dice “esperando aceptación del anfitrión”.

Una vez que están dentro de la partida, los usuarios podrán visualizar la lista de jugadores aceptados. Además, el creador tiene la capacidad de eliminar a cualquier usuario de la sala antes de que la partida comience. No hay un límite de tiempo establecido, ya que el anfitrión puede iniciar la partida en cualquier momento, siempre y cuando se haya alcanzado el número mínimo de jugadores.

10. La navegación (es decir la interacción con la pantalla de usuario en el browser) ¿Selección de menús? ¿Drag and drop? ¿combinaciones de lo anterior? ¿Hot-keys? (teclas especiales que al presionarlas hacen algo). En este caso se requiere un mock-up (es decir, pantallas dibujadas a modo de borrador, utilizando para ello un programa gratuito que les permita hacer mockups, incluso puede ser en una ppt que grafique el uso de la pantalla). Junto al mockup se requiere, por supuesto, explicaciones y descripciones de las pantallas.



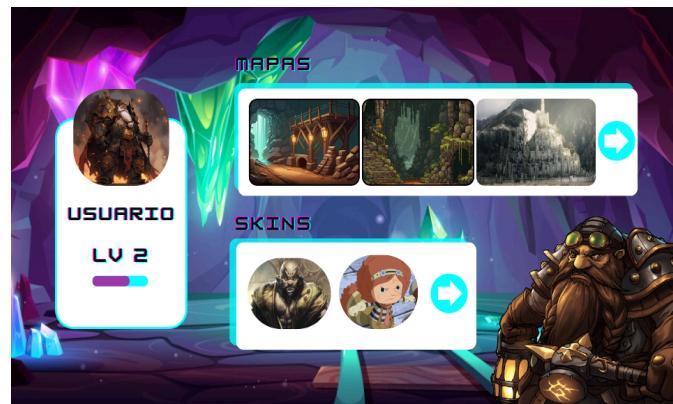
Se puede apreciar que va a tener un display simple. Hay un menú a la izquierda, el cual al clickear se despliega (esperamos lograr que sea animado). En el menú vamos a tener la opción de loguear cuyo diseño es muy similar al del mockup de registrarse que está en el punto 9, también contará con la opción de ver tu perfil donde podrás visualizar tu puntaje, tus skins (fotos de perfil) y tus tableros los cuales vas ganando a medida que subes de nivel/ ranking. El leaderboard, es una página que permitirá ver a los mejores jugadores (según su nivel). Rooms mostrará aquellas partidas que aún no comienzan, en otras palabras son aquellas partidas públicas a las que puedes unirte o privadas que están esperando que lleguen los jugadores que les faltan.

Para el menú principal tendrá combinaciones de teclas especiales (por definir durante la implementación) para poder acceder de forma rápida al perfil del jugador, a las salas disponibles



(rooms) y a la información (“acerca de” que se despliega al clickear el ícono de información que se ve en la esquina derecha).

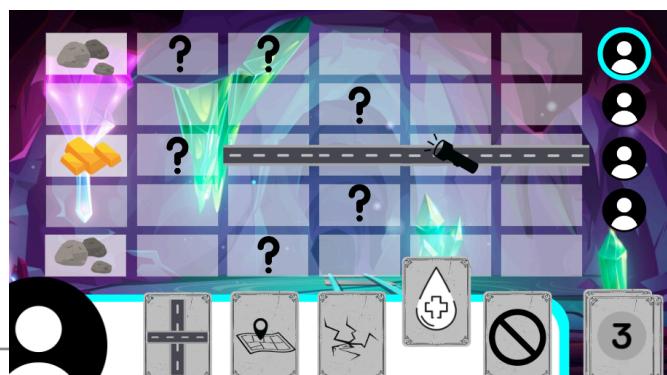
11. La interfaz de usuario permite visualizar los tableros, gráficos y/o mapas correspondientes que usted defina.



Podemos ver una visión generalizada del perfil de un jugador a grandes rasgos se ve que esté tendrá un nivel, mapas desbloqueados y sus skins (fotos de perfil) que podrá ir ganando a medida que suba de nivel de forma aleatoria. La asignación de una imagen a cada jugador a medida que sube de nivel va a ser realizada por el servidor. Para subir de nivel un jugador debe ganar partidas y cuando pierdes se le dará un descuento en la experiencia que tiene hasta ese punto.



12. Mock-ups y explicación de los mismos para los tableros de control, por ejemplo en el juego ¿habrá marcación de puntaje? ¿Habrá información pública (para todos los participantes) y privada (para un grupo o un sólo participante?). Por ejemplo, en un juego de cartas, yo puedo ver las que están en la mesa pero nadie puede ver “mi mano”.



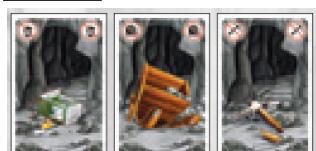


Para la jugabilidad se puede apreciar que va a ser de tipo tablero. El jugador en su turno deberá seleccionar una carta a jugar. Un jugador puede estar o no bloqueado en su turno. Si en su turno se encuentra bloqueado puede desbloquearse con la carta adecuada o puede botar cartas para que le llegue una nueva asignada aleatoriamente por parte del servidor. En caso de que no esté bloqueado este podrá seleccionar la carta a jugar. En caso de ser una carta de tablero (carta que afecta el tablero como un camino o un derrumbe) debe seleccionar la casilla dónde la quiere insertar y el servidor se encargará de que esto suceda. En caso de querer jugar una carta que afecte a otro jugador deberá seleccionar al jugador y esperar a ver si es que puede, ya que hay efectos especiales que hacen que ciertos jugadores no sean afectados por bloqueos. En caso de que ocurra lo anterior el jugador que jugó el ataque pierde el turno y se le cambia la carta jugada por otra del mazo.

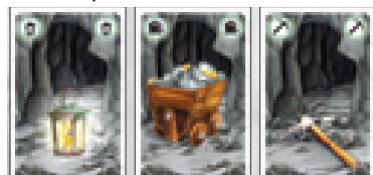
13. Descripción preliminar de los objetos y/o instancias que intervendrán (por ejemplo en un juego de aventuras, los tipos de personajes, las características de los personajes, el modelo de objetos asociado a ello, herencias posibles, diferentes tipos de armas, o de “encantamientos”, niveles que puede obtener cada personaje, etc.)

El juego consiste en un juego de cartas por lo que las instancias principales son las cartas y los jugadores:

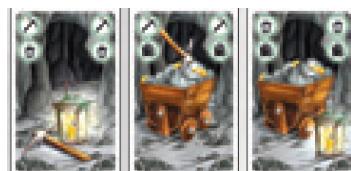
CARTAS:



Estas son las cartas de bloqueo o herramienta rota. Como su nombre lo indica va a permitir a los jugadores bloquear a otros. La única forma de que un jugador se pueda desbloquear es si usa la carta de desbloqueo adecuada, él mismo o otra persona.



Estas son las cartas de reparación, cada una puede reparar un tipo de carta específico. Por ejemplo: el faro solo puede reparar un bloqueo de tipo faro y así con las otras.



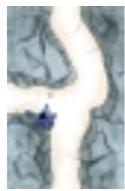
Estas son las cartas de desbloqueo doble. Tienen la capacidad de desbloquear alguno de los dos tipos de bloqueo que indica.



Carta de derrumbe. Permite destruir un camino en una sección elegida. Una vez que esta carta es usada para poder seguir debe ser arreglado el camino de la forma que permita la circulación adecuada del trayecto.



carta mapa. Esta carta permite visualizar alguna de las cartas destino para ver si hay oro o no. De esta forma puedes saber si la dirección en la que están dirigiéndose es la adecuada en el caso de los mineros, o si debes impedir que lleguen en el caso de si ere saboteur



cartas de camino/túnel. Son las que permiten el avance de los mineros a lo largo del mapa. Estas son las que van a permitir llegar o no al oro/piedras.

JUGADORES:



cada uno va a tener un tipo que será asignado al inicio de la partida. El jugador podrá ver el tipo de jugador que está debajo de su foto de perfil escogida. Además, de las cartas que este posee este podrá obtener habilidades especiales recibidas al poner cartas en una casilla especial (signo de pregunta en el mockup). Estas habilidades pueden ser tipo escudo (impide que te bloquen) o también jugar 2 turnos en una ronda.