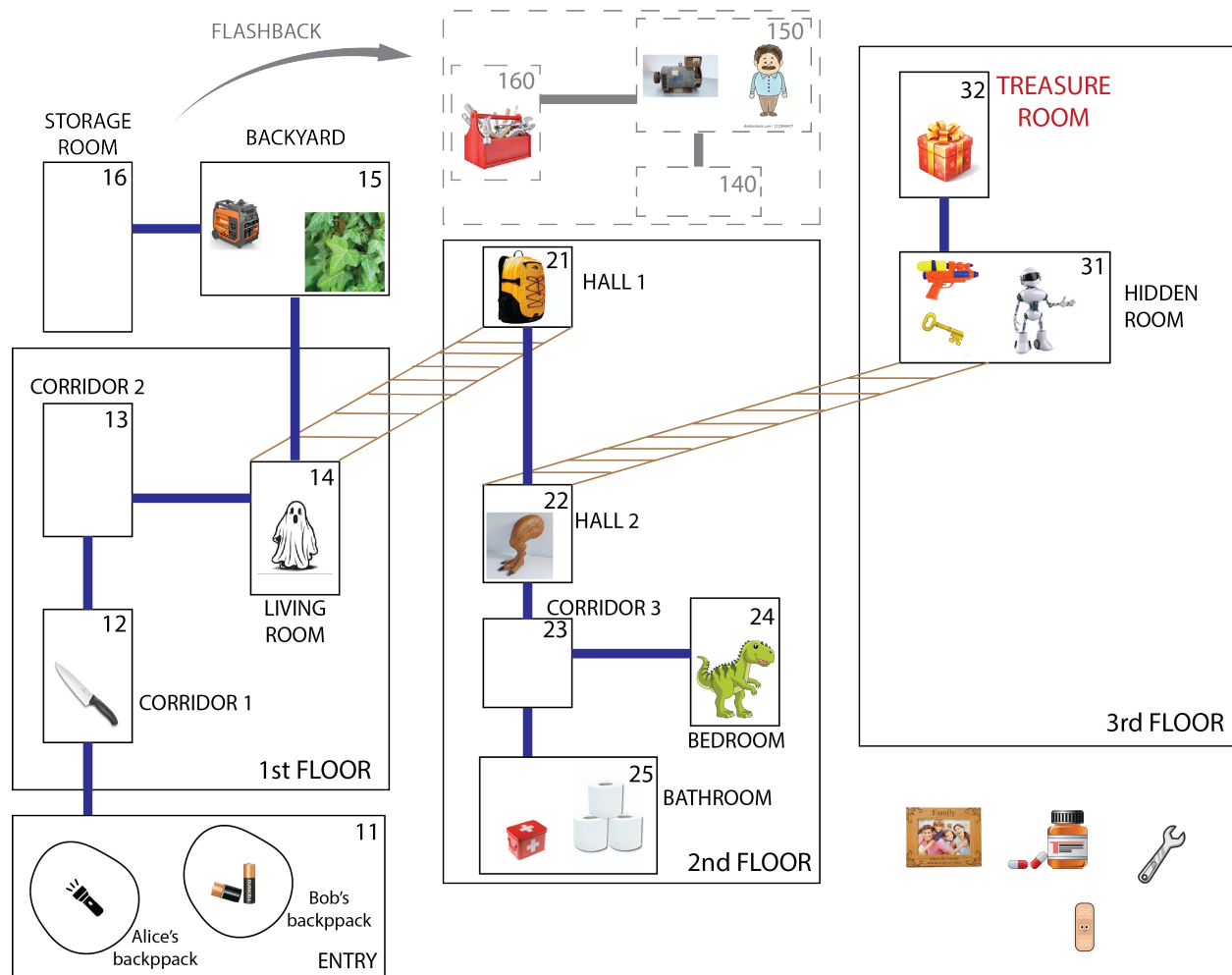


1 - Argumento del Juego



En la antigua casa donde crecieron, los hermanos Alice y Bob deben embarcarse en una aventura llena de desafíos. Hace algunas semanas, su abuela falleció y, estando en su lecho de muerte, les confesó a los dos jóvenes que había dejado algo muy importante para ellos en la casa donde crecieron. Sus padres insistieron en que dicha confesión fue fruto de un delirio, pero Alice y Bob deciden volver a aquella casa de su niñez, que ahora está abandonada, y encontrar el regalo que su abuela les dejó.

A medida que recorren cada rincón de la casa, se van enfrentando a obstáculos inesperados para poder recuperar un misterioso objeto que su abuela les dejó. La aventura les traerá recuerdos de su infancia y les hará reencontrarse con personajes inolvidables del pasado.





2 - Mapa de espacios de la casa



3 - Lista de objetos del juego

	Batteries		Lantern
	Old generator (flashback)		Generator
	StairsTo2ndFloor		StairsTo3rdFloor
	Key		Medkit
	Band-aids (inside the Medkit)		Medicine (inside the Medkit)
	DinosaurLeg		Kitchen knife
	FamilyPicture (inside the WrappedGift)		WrappedGift
	WaterGun		Toolbox (flashback)
	Wrench (inside the Toolbox)		Toilet paper
	Backpack		Ivy

4 - Lista de personajes del juego

	Ghost		Father (flashback)
	Dinosaur		Robocop

5 - Guía de comandos

ABANDON	EXIT	MOVE	TAKE
ATTACK	GIVE	OPEN	TEAM
CHAT	INSPECT	RECRUIT	TURN
DROP	LOAD	SAVE	USE

Nótese que para todos los comandos descritos de aquí en adelante estos no serán sensibles a letras mayúsculas o minúsculas, funcionando en ambos casos. Además, se espera que el usuario no inserte espacios adicionales al final de cada argumento, por lo que entradas como “take key ” devolverán un código de error. No obstante, se aceptan argumentos con más de una palabra.

- Exit:

Exit

Descripción: Comando para terminar la ejecución del programa.

Formato: El comando no espera argumentos.

- Move:

Move <direction>

Descripción: Permite al jugador moverse por los espacios que componen el mapa del juego. El comando comprende seis posibles direcciones, que pueden indicarse con su correspondiente traducción en inglés o con las iniciales de estas.

Formato: Espera un único argumento, que debe ser una de las siguientes opciones: N(orth), E(ast), W(est), S(outh) , U(p), D(own).

Casos de error: Si el argumento pasado al comando move no es uno de los anteriores, la dirección hacia la que se quiere despalazar el jugador tiene un enlace cerrado o inexistente.

- **Take:**

Take <object_name>

Descripción: Permite al jugador coger objetos encontrados en los espacios y portarlos en su inventario.

Formato: el comando espera un único argumento, que puede estar formado por más de una palabra, y que describa el nombre del objeto que se quiere coger. Para ello, el supuesto más importante para su correcta ejecución es que el objeto exista y se encuentre en el mismo espacio que el jugador.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El objeto no existe. El nombre introducido no describe un objeto conocido por el juego.
2. El objeto no se encuentra en el espacio actual del jugador.
3. El objeto no es movable, por lo que es imposible que cualquier jugador lo llegue a portar
4. El objeto es movable, sin embargo, depende de otro para ser cargado por un jugador. Si este no posee al objeto del que depende el que intenta coger, el comando no se ejecutará correctamente.

- **Drop:**

Drop <object_name>

Descripción: Permite al usuario soltar objetos que contenga en su inventario, el opuesto al comando Take.

Formato: Análogamente a take, espera un único argumento que describa el nombre del objeto que se quiere soltar en el espacio actual.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El objeto no existe, o el nombre no se ha introducido correctamente
2. El jugador no lleva dicho objeto en su mochila.
3. El jugador intenta soltar un objeto del que dependen otros objetos de su inventario.

- **Recruit:**

Recruit <character_name>

Descripción: Recruit es el comando que permite reclutar un NPC del juego a su equipo, que contribuirá en las batallas y le seguirá durante la partida.

Formato: El comando espera un único argumento, que puede estar formado por más de una palabra, que describa el nombre del personaje que se quiere reclutar

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. No existe un personaje en el juego con el nombre indicado.
2. El personaje indicado no se encuentra en el mismo espacio que el jugador actual.
3. El personaje no es amigable.
4. El personaje ya está reclutado por otro jugador.

- **Abandon**

Abandon <character_name>

Descripción: Comando que permite abandonar personajes que ya han sido reclutados por un jugador.

Formato: Espera un único argumento que describa el nombre del personaje a ser abandonado. Requisito esencial para la correcta ejecución del comando es que el personaje en efecto haya sido reclutado por el jugador

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. No existe un personaje conocido por el juego con el nombre introducido.
2. El personaje indicado no se encuentra entre los seguidores actuales del jugador

- **Chat:**

Chat <character_name>

Descripción: Comando que permite interactuar con los personajes amigables, mostrando sus mensajes por una zona de la pantalla.

Formato: Espera un único argumento que describa el nombre del personaje a ser abandonado. Requisito esencial para la correcta ejecución del comando es que el personaje en efecto haya sido reclutado por el jugador

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. No existe un personaje conocido por el juego con ese nombre.
2. El personaje indicado no se encuentra en el espacio actual.

- **Attack:**

Attack <character_name>

Descripción: Comando que permite atacar a los personajes enemigos del juego. Los ataques pueden ser conjuntos, si el jugador está formando un equipo con otro(s) y su efectividad varía en función del número de jugadores y personajes que atacan.

Formato: Espera un solo argumento que describe el nombre del personaje enemigo que se quiere atacar.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El personaje no existe, el nombre introducido no describe a un personaje conocido por el juego.
2. El personaje es amigable.
3. El personaje no se encuentra en el mismo espacio que el jugador que ataca
4. El personaje está muerto.

Información adicional:

- En ciertos momentos del juego, este comando requiere que, además, se porte un objeto en concreto para realizar el ataque (como la linterna y el cuchillo cuando Alice y Bob se enfrentan al fantasma del primer piso).
- El comando imprime un mensaje “TEAM ATTACK !” en la caja de diálogos cuando la acción del ataque daña al enemigo (no son los jugadores ni los characters los que resultan dañados) y los jugadores han hecho “team” previamente.

- Inspect:**Inspect <object_name>**

Descripción: Comando que permite inspeccionar los objetos del juego, mostrando una breve descripción de estos por pantalla, en la caja de la parte izquierda.

Formato: El comando espera un único argumento que describe el nombre del objeto que se quiere inspeccionar. Importante para la ejecución que el objeto sea accesible al jugador (se encuentre en su inventario o en el espacio actual).

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El objeto no existe, el nombre introducido no describe un objeto conocido por el juego.
2. El objeto no se sitúa en el inventario del jugador, o bien, en el espacio que se encuentra.

- Save:**Save <file_name>**

Descripción: Comando que permite guardar el estado actual del juego en un fichero que se indica a continuación de la directiva “save”

Formato: El comando espera un único argumento que describe el nombre del fichero en el que se guardará el estado actual del juego.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El archivo introducido ya existe y los permisos del archivo no permiten la escritura.

- **Load:**

Load <file_name>

Descripción: Comando que permite cargar una partida determinada, cuyo estado está guardado en un fichero de datos, cuyo nombre se pasa como argumento del comando.

Formato: El comando espera un único argumento que describe el nombre del fichero en el que se ha guardado el estado del juego a cargar.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El archivo introducido no permite la lectura de los datos por el programa.
2. No hay memoria suficiente en la máquina para almacenar el juego.

- **Team:**

Team <player_name>

Descripción: Comando que permite formar equipos con el otro jugador para colaborar durante la partida.

Formato: El comando espera un único argumento que se corresponde al nombre del jugador con el que se quiere formar equipo.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El jugador cuyo nombre se ha introducido no existe en el juego actual.
2. El jugador cuyo nombre se introduce ya está en el equipo del jugador actual.
3. El jugador con el que se quiere formar equipo no lo acepta.

- **Turn:**

Turn

Descripción: Comando que permite pasar el turno de la jugada a los otros jugadores.

Formato: El comando no espera argumentos.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El jugador actual se encuentra en un flashback.

- **Open:**

Open <connection_name> with <object_name>

Descripción: Comando que permite abrir puertas (enlaces) en el juego haciendo uso de los objetos que porta en su inventario. Una vez se haya ejecutado el comando correctamente, el objeto usado para abrir el enlace se “gastará” y ya no estará disponible para su uso posterior.

Formato: El comando espera dos argumentos, separados por el string “ with ”. El primero de ellos consiste del nombre del enlace que se quiere abrir (cuya sintaxis se puede consultar en las secciones anteriores) y el segundo es el nombre del objeto que se usará para desbloquear dicho enlace.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El Formato de entrada es incorrecto (incluso puede deberse a espacios adicionales al inicio o final de cada argumento, e.g. “puerta ”).
2. Los argumentos introducidos no describen objetos y enlaces conocidos por el juego.
3. El objeto descrito no se encuentra en el inventario del jugador.
4. El enlace descrito no parte del espacio actual del jugador.
5. El objeto descrito no es el adecuado para abrir dicho enlace.
6. El jugador actualmente no porta un objeto del que depende el que se quiere emplear.

- **Use:**

Use <object_name> || Use <object_name> over <character_name>

Descripción: Comando que permite usar objetos del juego llevado por el jugador para alterar los puntos de vida del usuario o personajes del ambiente. Una vez ejecutado, el objeto que se haya utilizado dejará de estar disponible.

Formato: El comando contempla dos modos de uso, con uno y otro con dos argumentos separados por el string “ over “.

Si solamente se introduce un argumento, el comando modificará la vida del jugador que lo ejecute. En caso de que se introduzcan dos argumentos el comando modificará la vida del personaje indicado.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El formato de entrada es incorrecto (al igual que open)
2. Los argumentos introducidos no describen objetos y personajes conocidos por el juego.
3. El objeto introducido no está en el inventario del jugador.
4. El objeto del que depende el objeto indicado no está en el inventario del jugador actual.
5. El personaje introducido no está en el espacio actual.
6. El personaje introducido no es amigable.
7. El objeto introducido no modifica la vida del personaje.

- **Give:**

Give <object_name> to <player_name>

Descripción: Comando que permite intercambiar objetos con los demás jugadores

Formato: El comando espera dos argumentos, que se corresponden al nombre del objeto que se quiere dar y el nombre del jugador que lo recibe, que irán separados obligatoriamente por el separador “ to ”.

Casos de error: El comando no se ejecutará correctamente en cualquiera de los supuestos siguientes.

1. El formato introducido es incorrecto.
2. Los argumentos introducidos no describen objetos y jugadores conocidos por el juego.
3. El objeto introducido no está en el inventario del jugador
4. El objeto introducido depende de otro, que no está en el inventario del receptor.
5. El jugador introducido no se encuentra en el mismo espacio que el jugador actual.
6. La mochila (inventario) del jugador receptor está llena.

6 - Guía de la Interfaz Gráfica

La interfaz de usuario del juego se divide en varias zonas, en las que se muestran informaciones relevantes acerca del juego. A continuación, se detalla cada una de estas zonas.

6.1 - Zona principal (centro) - Descripción del espacio actual

Aquí se incluye la descripción gráfica del espacio actual, una matriz de caracteres ASCII de 151*57 caracteres, encima de la cual también se imprime el nombre del espacio actual.

Esta sección es principalmente estética, con la información relevante del juego mayoritariamente situada en otras zonas de la pantalla.

6.2 - Zona superior - Guía de las misiones

En esta zona se incluyen dos campos orientativos para los jugadores, que dan información acerca de la misión actual y el objetivo que queda pendiente para pasar a su siguiente etapa.

6.3 – Zona central-izquierda – Diálogos

Destinada a la impresión de los diálogos durante las cinemáticas del juego, así como frases individuales que los personajes van diciendo. Esta área imprime, además, la descripción de los objetos inspeccionados.

6.4 – Zona central-derecha – Elementos del juego

Incluye la información más relevante para la interacción de los jugadores con el espacio en el que se encuentran y con los personajes y objetos del juego en general.

OBJECTS IN THE GAME: sección dedicada a una lista de los objetos del juego que se hallan, o bien en la mochila del jugador, o bien en el mismo piso que el jugador. En esta zona se imprimen: el nombre del objeto, su descripción gráfica y, en caso de hallarse en un espacio descubierto o en la mochila, su localización (en el último caso, se imprimiría “backpack”). Además, se imprime si el objeto se puede mover y, en caso de poder abrir algún enlace o curar puntos de vida, estos datos se imprimen también.

CHARACTERS IN THE GAME: sección homóloga a la lista de objetos, pero para personajes, imprimiendo datos específicos como si el character es amistoso (YES/NO), su vida actual y, en caso de que esté siguiendo a algún jugador, el nombre de este.

PLAYER INFORMATION: contiene información relevante sobre el jugador actual, su descripción gráfica y los puntos de vida actuales. También se incluye una lista de objetos que lleva en el inventario, con su descripción gráfica.

CURRENT SPACE INFORMATION: sigue el modelo de los apartados anteriores y expone los elementos del espacio actual, objetos y personajes, sin especificar sus detalles, pues estos ya se hallan en la parte superior.

6.5 – Zona inferior-izquierda – Brújula

Actuando como guía para la orientación del jugador, la brújula muestra en pantalla el nombre de cada uno de los espacios contiguos al actual.

Para los espacios en el mismo nivel que el actual, sus nombres se muestran a continuación de una flecha. Para los espacios conectados verticalmente, si el enlace que los une está abierto, se imprimirán directamente sus nombres; en caso contrario, este será sustituido por “????”.

6.6 – Zona inferior-central – Comandos

Expone los comandos disponibles para el uso del jugador en un momento dado.

6.7 – Zona inferior-derecha – Feedback

A lo largo del juego, esta zona guardará la información de interés acerca de los tres últimos comandos ejecutados por cada jugador: El comando en sí, un código de éxito de su ejecución, y al jugador al que corresponde.

7 – Instrucciones de compilación

El proyecto está comprendido para utilizar el lenguaje de Make para automatizar su compilación. A continuación, se exponen algunas de las instrucciones más útiles para su manejo:

make all	compila el ejecutable principal del juego
make tests	compila todos los ejecutables de prueba
make general	compila el ejecutable del juego y los ejecutables de las pruebas

make clean	elimina archivos objeto y ejecutables generados
make clean docs	eliminate archivos de documentación generados y carpetas vacías
make docs	genera documentación usando <i>Doxygen</i>
make run	ejecuta el juego con el archivo de datos <i>house.dat</i>
make runLog	ejecuta el juego con datos en <i>house.dat</i> y guarda el registro en <i>Logfile</i>
make runLog_read1	ejecuta el juego con <i>house.dat</i> , guarda registro en <i>Logfile</i> y lee comandos de <i>game1.cmd</i> o <i>game2.cmd</i>
make runLog_read2	ejecuta el juego con <i>house.dat</i> , guarda registro en <i>Logfile</i> y lee comandos de <i>game2.cmd</i> . Posteriormente contrasta el registro con la salida esperada
make runV	ejecuta juego con Valgrind para verificar fugas de memoria
make runD	ejecuta el juego en el modo determinista

Ejecutar pruebas individuales:

make character_test_run
make space_test_run
make set_test_run
make inventory_test_run
make link_test_run
make object_test_run
make player_test_run

Ejecutar pruebas individuales con Valgrind:

make character_test_runV
make space_test_runV
make set_test_runV
make inventory_test_runV
make link_test_runV

make object_test_runV

make player_test_runV

make test_run ejecuta todas las pruebas secuencialmente

make test_runV ejecuta todas las pruebas secuencialmente con Valgrind

8 – Walkthrough

En esta sección se expondrá un guion de comandos a seguir por el usuario para llegar al final del juego, así como las funcionalidades que estos exponen.

Para ejecutar el juego, asumiendo que el usuario tiene la herramienta make instalada, es suficiente con ejecutar el comando “make” seguido de “make run” en la terminal.

Nótese que la interfaz del juego normalmente no cabe en la pantalla estándar, por lo que puede ser necesario presionar ‘Ctrl’ + ‘-’ algunas veces.

Para garantizar que estos comandos se ejecuten correctamente, se puede ejecutar el juego en modo determinista, que garantiza la victoria de los jugadores en cada ataque y elimina toda aleatoriedad del juego.

0. Formar equipo / Encender Linterna:

- (Alice) Team Bob - Colaboración entre jugadores.
- (Bob) Y - Confirmación de la creación del equipo
- (Bob) Give batteries to Alice – Intercambio de objetos.

1. Navegar hasta generador antiguo.

- (Alice) Move north
- (Alice) Move north
- (Alice) Move east
- (Alice) Move north
- (Alice) Inspect old generator – Enseña una breve descripción del objeto y transporta al jugador a un flashback de su infancia.

2. Arreglar generador (flashback)

- (Alice) Chat father – Capacidad de hablar con los personajes amigables.
- (Alice) Chat father – Cada personaje puede tener más de un mensaje
- (Alice) Move west
- (Alice) Inspect tool box – Inspeccionar objetos grandes permite coger objetos más pequeños contenidos en ellos.
- (Alice) Take wrench – Permite coger objetos de los espacios y almacenarlos en el inventario del jugador
- (Alice) Move east
- (Alice) Inspect generator – Inspeccionar el generador con el objeto “wrench” en el inventario saca al jugador del flashback de Alice y enciende las luces de la casa.

3. Llegar hasta el tercer piso.

- (Alice) Move south
- (Alice) Turn – Cambiar de turno manualmente
- (Bob) Move north
- (Bob) Move north
- (Bob) Move east
- (Bob) Attack ghost – Un ataque conjunto no es permitido a no ser que cada jugador del equipo tenga los objetos oportunos para realizarlo
- (Bob) Move west
- (Bob) Move south
- (Bob) Take KitchenKnife
- (Bob) Move north
- (Bob) Move east
- (Bob) Attack Ghost – Una vez Bob tiene el cuchillo y Alice la Linterna el ataque es exitoso (el comando se repite hasta matar al fantasma).
- (Bob) Move up – Una vez Ghost se haya muerto se desbloquea el enlace hacia la segunda planta.
- (Bob) Turn
- (Alice) Move up

4. Curar a Alice

- (Alice) Move south
- (Alice) Move south
- (Alice) Move south

- (Alice) Inspect MedKit
- (Alice) Drop batteries – El juego no permite cargar más objetos que los que quepan en el inventario del jugador.
- (Alice) Take medicine
- (Alice) Use medicine – Use sin segundo argumento aumenta la vida del jugador según lo estipulado en la descripción del objeto.

5. Reclutar dinosaurio

- (Alice) Turn
- (Bob) Move south
- (Bob) Move south
- (Bob) Move east
- (Bob) Chat dinosaur
- (Bob) Move west
- (Bob) Move north
- (Bob) Take DinosaurLeg
- (Bob) Move south
- (Bob) Move east
- (Bob) Recruit dinosaur – Solo es posible si el jugador tiene la pata de dinosaurio en su inventario.

6. Llegar hasta la 3ª planta

- (Bob) Move west
- (Bob) Move south
- (Bob) Take Band-Aids
- (Bob) Use Band-Aids over Dinosaur – Use con dos argumentos aumenta la vida del personaje cuyo nombre es indicado según la cantidad estipulada en la descripción del objeto.
- (Bob) Move north
- (Bob) Move north
- (Bob) Move up
- (Bob) Turn
- (Alice) Move north
- (Alice) Move north
- (Alice) Move up

7. Matar el Boss final

- (Alice) Turn

- (Bob) Take WaterGun
- (Bob) Attack Robocop 5 Times – Solo es posible con el objeto WaterGun

8. Final del juego

- (Bob) Take key
- (Bob) Open HiddenRoomToTreasureRoom with key – Abrir enlaces determinados manualente usando objetos determinados
- (Bob) Move north
- (Bob) Turn
- (Alice) Move north
- (Alice) Turn
- (Bob) Inspect WrappedGift
- (Bob) Take FamilyPicture
- (Bob) Inspect FamilyPicture
- (Bob) Exit – Fin