Evidencia: Situación Problema

Salvador Fernando Camacho Hernández A01634777

Jorge Carrillo Castro A01634630

Programación Orientada a objetos

Fabiola Uribe Peralta

Grupo 1

Viernes 11 de junio de 2021

Situación Problema

En esta ocasión en la situación problema se tendrá que realizar un juego serio (serious game), los cuales tienen como principal objetivo el impartir aprendizaje al igual que entretener. Se le denomina como un juego serio a aquel que es utilizado en distintos sectores como lo son sector educativo, científico, ingeniería, política entre otras. Una de las utilidades de este tipo de juegos es el enseñar habilidades específicas con el uso del entretenimiento para poder facilitar el aprendizaje. Para esta situación problema, creamos el juego *Bloody Pickles* debido a cuestiones de tiempo solo se pudo agregar el componente lúdico y la parte del aprendizaje se dejó a un lado.

Historia

Un pepinillo despierta (personaje principal) en la barra de la cocina. Ve un celular cerca, intenta acercarse girando para hacer una llamada. Como no está acostumbrado a ser un pepinillo no se sabe mover bien. Gira en la dirección contraria y tira una naranja al suelo lo que llama la atención de Zabu. Un perro gigante - al menos para un pepinillo - fuerte, de esos perros que intimidan hasta a las personas pero aún peor hambrienta, comida favorita pepinillos asustados. Nuestro pepinillo asustado y en pánico gira hacia el teléfono para hacer su llamada por ayuda pero por el miedo y ser nuevo a ser un pepinillo gira con demasiada fuerza y termina cayendo en el bote de basura. gira de más y cae en el bote de la basura. llega la hora de recolección de basura. Pepinillo intenta gritar por ayuda pero nadie contesta, llega la hora del camión de la basura y es arrojado al camión. Una vez en el camión de basura el pepinillo encuentra palillos los cuales utiliza para formar su cuerpo (2 piernas y 2 brazos). El pepinillo comienza a explorar el camión de basura, donde se encontrará múltiples objetos de los cuales uno es una caja de cereal con un mapa en la parte de atrás. Este mapa contiene las instrucciones para llegar a la cura (lo cual es el objetivo del jeugo) para poder volver a ser humano. Analiza el mapa y se da cuenta que va por el camino correcto para llegar a la cura, solo debe permanecer en el camión unas cuantas paradas más. Mientras el pepinillo espera, a lo lejos del camión escucha un ruido y decide ir a investigar. Al llegar al ruido se encuentra con una rata comiendo chorizo verde, de Toluca, el pepinillo intenta pelear contra la rata usando un palillo extra como espada, pero al ser muy débil sale derrotado. La rata le arranca un brazo y el pepinillo se ve obligado a saltar del camión, por suerte el pepinillo cae en frente del centro comercial donde se encuentra la farmacia con la cura. Aquí comienza la verdadera aventura Pepinillo, a lo largo de su camino se encuentra con diversos enemigos que se lo quieren comer - principalmente animales, por lo cual tendrá que recolectar objetos para poder defenderse y volverse más fuerte.

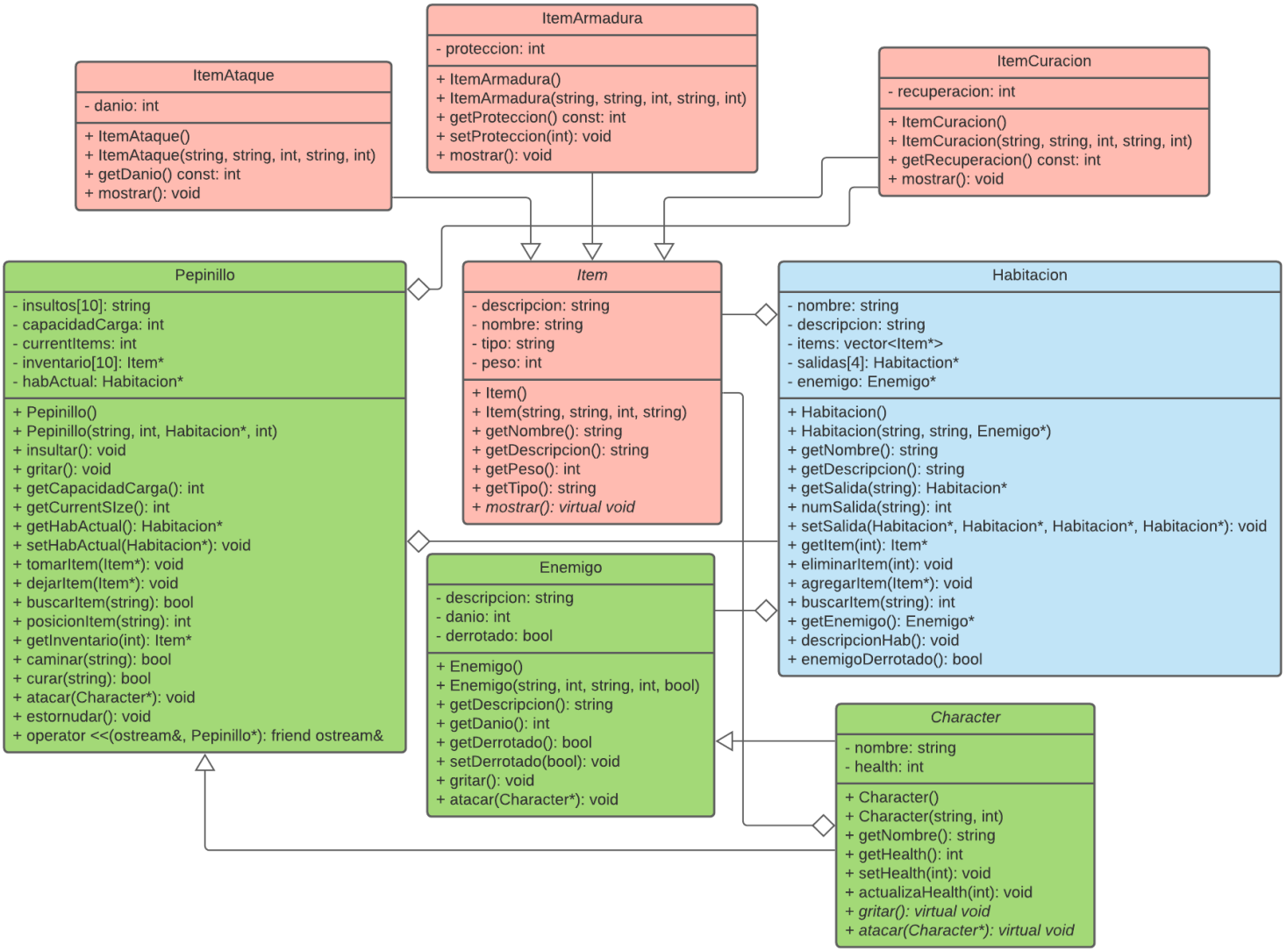
Pepinillo se da cuenta que para poder llegar a la farmacia, y obtener la cura para volver a ser humano, tiene que pasar por BestBuy y PETCO. Pepinillo entra a BestBuy y antes de pasar por la rendija que lo lleva a PETCO decide explorar la tienda para conseguir brazos, piernas, un arma, entre otros ítems. Consigue sus extremidades y un arma. Para poder pasar a donde está la rendija por la cual llegar a PETCO. El pepinillo se encuentra con la opción de explorar un cuarto de conserje o ir directo a la rendija.

> Si decide ir al cuarto, a continuación está la opción de regresar e ir por la rendija o trepar por un trapeador que lleva a un ducto de ventilación.

Si va por el ducto de ventilación se da cuenta que lo lleva directo a la farmacia pero antes de llegar se encuentra con su némesis, rata del bote de basura - que al parecer siguió a Pepinillo.
Lucha contra la rata y al derrotar puede pasar a la farmacia.

En PETCO se encuentra con algo que no esperaba, una mafia de hamsters controla el lugar y para llegar a la farmacia debe vencer al líder de la mafia “Señor hamster Rogelio”. Una vez que vence al líder, tiene el camino despejado para poder llegar a la farmacia. Ya en la farmacia encuentra la cura y puede volver a ser humano.

UML



Herencia

La herencia la aplicamos en 2 instancias con la clase Ítem y Character, cada una de estas clases con su respectivo polimorfismo.
En Ítem utilizamos herencia porque tenemos 3 tipos de ítems diferentes - ítem de armadura, ítem de curación y item de ataque. El polimorfismo en ítem se encuentra en el método mostrar - que es un método virtual y hace la clase abstracta. El método mostrar imprime los

atributos del ítem en un formato específico, se necesita porque un atributo nuevo y diferente se agrega a cada uno de los 3 tipos de ítems.
En Character utilizamos herencia porque la clase pepinillo - personaje principal - y la clase enemigo comparte algunos atributos y métodos, por lo cual las dos clases heredan de Character. El polimorfismo en Character se encuentra en 2 métodos - que son métodos virtuales y hacen a la clase abstracta - el método gritar que imprime y el método atacar que recibe un apuntador a objeto de tipo Character y generando un número random entre 0 a 2 se decide con una probabilidad de ⅔ de acertar el ataque, baja la salud del atacado e imprime si el ataque fue exitoso o fallo. Utilizamos el polimorfismo en estos métodos porque tanto pepinillo como enemigo deben ser capaces de atacar y bajar salud, cada vez que atacan gritan pero el grito es diferente entre pepinillo y enemigo; sobreescribimos el método.

Sobrecarga de operadores
En el programa aplicamos sobrecarga de operadores en la clase de pepinillo para poder imprimir el inventario del pepinillo o "Pepinventario" con el uso de un ciclo for. El operador que utilizamos para que esto fuera posible, es el operador de output (<<).

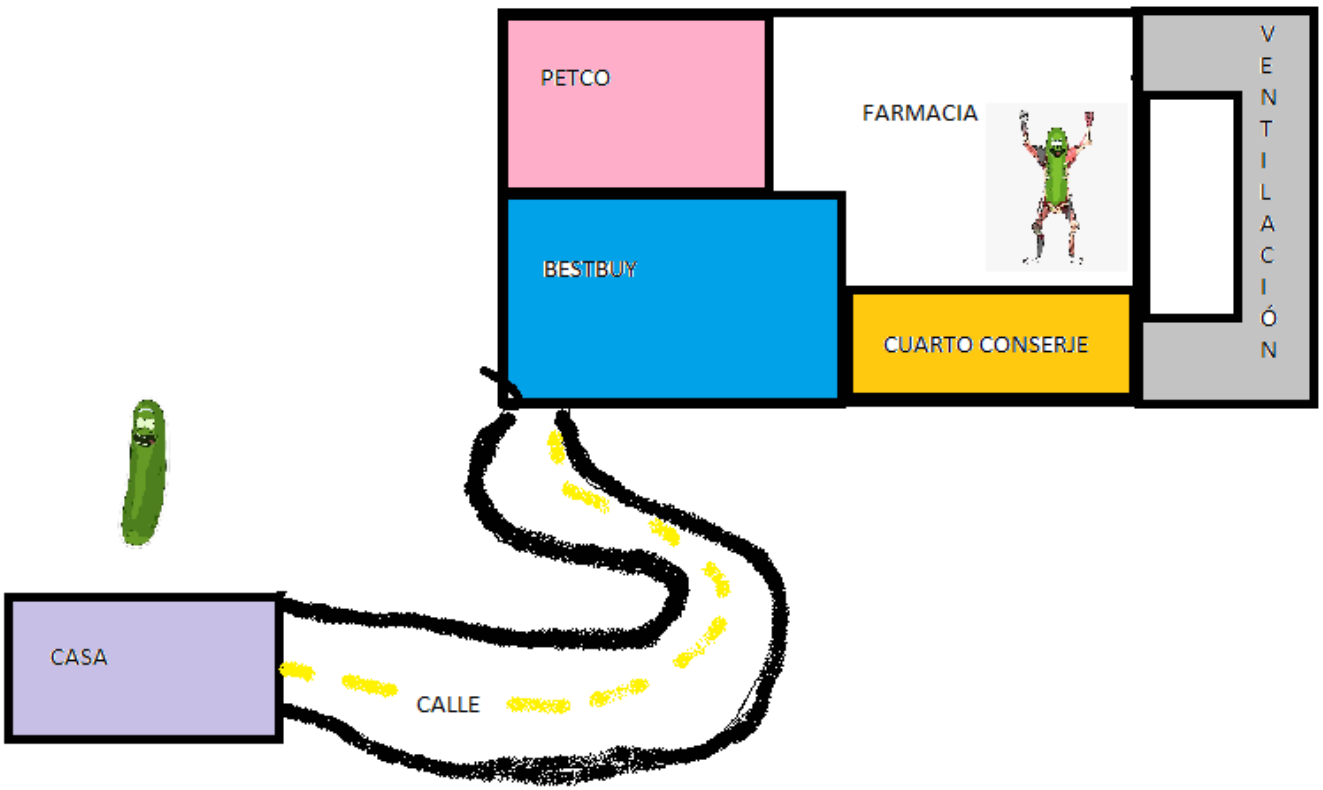
Excepción
La excepción la utilizamos en el método bienvenida de, para que en dado caso que el usuario escriba de manera incorrecta la palabra, el programa no se detenga.

Caso en el que dejaría de funcionar
El programa deja de funcionar cuando se ejecuta el comando ir en la habitaciones que no tienes un enemigo - su enemigo es un nullptr - y la dirección que recibe el método ir es una nullptr. También en el método ir, si el usuario escribe mal o cualquier cosa que no sean las palabras "norte", "sur", "este" u "oeste", tal y como están escritas.

Conclusiones
Jorge: A lo largo de este curso aprendí acerca de polimorfismo y herencia sin embargo con este proyecto pude poner toda la teoría en práctica. Al momento que estás en clase haciendo ejemplos, todo sale fluido y si hay algún error puedes preguntar a la maestra. El verdadero aprendizaje comienza cuando intentas aplicar lo visto en clase en la creación del juego ya que surgen muchos problemas que ocasionan que tengas que investigar por fuera y estar horas pensando. Cuando empecé a trabajar en el proyecto era muy lento al momento de aplicar herencia y polimorfismo. A lo largo de la creación del código he notado que cada vez me hacía más fluido al momento de aplicar herencia y polimorfismo ya que cada vez tenía menos errores, al igual que ya no tengo la necesidad de checar código creado en clase para guiarme. Durante la creación del código presentamos varias dificultades, una de las más importantes es que no podíamos crear la clase "enemigo". Duramos mucho tiempo buscando una solución a este problema sin embargo no la encontramos y tuvimos que asistir a asesoría. En asesoría la maestra nos comentó que habíamos creado una herencia circular en la clase de enemigo y de habitación. Debido a que la clase enemigo incluía una habitación y la clase habitación incluía un enemigo, el programa no sabía cuál de las dos clases iba primero por lo cual se confundía y no se podía crear la clase enemigo.

Fernando:
Pienso que este proyecto fue muy divertido de hacer. La idea de hacer un videojuego al principio parecía algo para lo cual aún no tenía las habilidades para desarrollar. Sin embargo, hay muchos tipos de videojuegos y me gustó aprender de este tipo de juego que ya no está de moda pero que en mi es muy creativo.
Me ayudó aprender y practicar los conceptos básicos de programación orientada a objetos. En clase sentí que entendía todo. Herencia, polimorfismo, sobrecarga, apuntadores al 100 siguiente, pero al momento de hacer las tareas - el proyecto más que nada - las dudas específicas empezaron a salir, resolver esos problemas fue lo que realmente me ayudó a comprender los temas lo mejor que puedo al momento.
Finalmente creo que el proyecto me ayudó a generar un callo para programar durante varios días y horas.
Tuvimos muchas dificultades, la mayoría se resolvieron de manera rápida buscando una explicación en google. Sin embargo, nuestro everest tomó forma en la clase enemigo. Para resolver el problema cambiamos, agregamos y/o borramos partes del código; nada funcionó. Pudimos resolver el problema gracias a la asesoría de la maestra, quien nos dijo que hicimos una llamada circular entre habitación y enemigo; el código no sabía qué hacer, lo que generó la mayoría de problemas que teníamos al momento.

Mapa



Referencias

Función delay() en C++.(2010). *Código C++. Ayuda para tu tarea C++*.Recuperado de https://ccodigo.wordpress.com/2010/09/24/funcion-delay-en-c/

Rand. (s.f.). *cplusplus.com.* Recuperado de https://www.cplusplus.com/reference/cstdlib/rand/