Texto

Descripción generada automáticamente       

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Querétaro**

**Programación orientada a objetos**

**(Gpo 1)**

**Profesor:**

Pedro Oscar Pérez Murueta

**Alumno:**

Oscar Eduardo Nieto Espitia - A01705090

**Actividad:**

Entrega del primer avance del proyecto final

**Programación orientada a objetos**

**Proyecto “Octo-π”**



El profesor Octo-π tiene como objetivo enseñar a todos los niños de primaria, pero hay una complicación, el profesor Octo-π no puede acceder a una computadora ya que todavía no se ha desarrollado un equipo capaz de funcionar el cien por ciento del tiempo debajo de agua salada, así que tienes dos opciones, adaptar una computadora para que Octo-π pueda trabajar o también puedes ayudarlo a automatizar lo que pide el profesor. Como buen estudiante de ingeniería en sistemas ayudas a programar las especificaciones que quiere el Octo-π para que sus alumnos se sientan atraídos al objetivo del profesor.

Ya que el profesor piensa enfocarse en reforzar los conocimientos de todos los niños de primaria decide crear una historia por la que puedas progresar conforme vayas subiendo de nivel enfrentando diversos problemas matemáticos, Octo-π nos narra sobre una rivalidad que ha existido entre él y Shark-τ (Shark-tau), debido a que el piensa que el apellido -τ es más importante que el -π, esto se debe a que en muchos problemas matemáticos es más fácil hacer cálculos con τ (6.28…) que con π (3.14…). Shark-τ afirma que la relación de la circunferencia al diámetro es incorrecta, él dice que los círculos no tienen que ver con los diámetros, sino con radios, por lo que si defines la constante del circulo como la relación de la circunferencia al diámetro, lo que estás haciendo realmente es la relación entre la circunferencia con el doble del radio.

Octo-π

Shark-τ

Por lo tanto, Octo-π quiere recrear las aventuras que vivió con su discípulo enfrentando a la familia τ (tau) y sus hermanos. El jugador tomara el papel del discípulo e intentara resolver los problemas que le lancen los hermanos τ (tau), conforme vayas derrotando a los jefes finales de cada piso (cada respuesta correcta le bajas vida al boss, cada respuesta incorrecta te baja vida a ti) vas a ir avanzando en la historia, ganaras recompensas y experiencia conforme avances de piso.

Cada piso tendrá su pool de preguntas. Todas las preguntas son de opción múltiple y tienen un puntaje, dicho puntaje es la vida que le puedes bajar al boss, el daño lo puedes ir aumentando conforme ganes recompensas y dinero.

Temas según la dificultad:

* Piso 1: Suma y resta de números de dos cifras, multiplicación y división simple, longitud peso y volumen
* Piso 2: Suma y resta de números de cuatro cifras, numeración romana, lectura y escritura de números hasta siete cifras
* Piso 3: Operaciones con fracciones, operaciones con decimales
* Piso 4: Multiplicaciones y divisiones avanzadas, potencias y raíces cuadradas

Como incentivo, el profesor Octo-π desea implementar un sistema de puntos para que el usuario pueda canjear ciertas recompensas, conforme vayas enfrentando y derrotando a los bosses de cada piso vas a conseguir experiencia y Mora (dinero), con el que podrás comprar equipamiento para regenerar vida, la experiencia servirá para mejorar tus stats de vida así como de daño. Al terminar el juego, dependiendo de cuanta Mora (dinero) te haya sobrado, podrás comprar canjes especiales para la materia

|  |  |
| --- | --- |
| **Canje de Puntos** | **Coste de Puntos** |
| Un dulce | 10 |
| Entregar tarea atrasada | 15 |
| Exentar tarea | 20 |
| Exentar proyecto | 30 |
| Exentar examen | 40 |
| Exentar la materia | 100 |

**Sinopsis:**

Es el año 1877, once años despues de la era Meiji, reencarnas en el cuerpo de un aprendiz samurai del profesor Octo-π, ahora se encuentra en una revolucion de samurais de la provincia de Satsuma, dicha revolucion se encuentra liderada por Octo-π. La revolucion llevaba varias batallas por derrocar al actual gobierno de la familia τ(tau), esta es la ultima y mas importante de incontables series de levantamientos contra el gobierno que llevaba rigiendo 628 años.

La razon por la que ambas casas estaban en guerra es por una situacion de ideologias, la casa τ (tau) cree que el apellido de su casa deberia de predominar en los archivos de historia debido a su relevancia en ciertas ecuaciones matematicas. Esto de debe a que en muchos problemas matemáticos es más fácil hacer cálculos con τ (6.28…) que con π (3.14…). Shark-τ afirma que la relación de la circunferencia al diámetro es incorrecta, él dice que los círculos no tienen que ver con los diámetros, sino con radios, por lo que si defines la constante del circulo como la relación de la circunferencia al diámetro, lo que estás haciendo realmente es la relación entre la circunferencia con el doble del radio.

Te sitúas en la última batalla que probablemente definirá el destino de ambas casas, debes derrotar a todos los familiares de la casa τ(tau).

En la era Meiji se practica un tipo de combate muy específico, el combate pregRes. Dicho tipo de combate media los conocimientos de cada individuo y ganaba aquel que respondía correctamente las preguntas del contrincante. Se ha establecido que aquel que quiera desafiar a alguien es el que tiene que responder a las preguntas del contrincante.

Actualmente te situas en la primera region de la casa de la familia τ (tau) y debes de ir bajando en profundidad dentro del abismo para encontrarte con tu objetivo, shark-τ, por cada piso que vayas bajando tus encuentros se haran mas dificiles.

Octo-π queda extremadamente fatigado debido a una enboscada, ahora quedas tu, su aprendiz, para derrotar a shark-τ.

**Instrucciones**

Tu personaje inicia con habilidades basicas, conforme vayas avanzando en la historia podras mejorar tus habilidades mediante experiencia, dichas habilidades te proporcionaran mayor daño al momento de responer las preguntas de tu contrincante.

Tu personaje inicia con una pocion de vida, con ella podras regenerar un poco de tu vida actual. Ganaras Mora y experiencia conforme vayas enfrentando a los bosses de cada piso

**Descripcion de los pisos:**

* **El borde del abismo:** Es el lindero entre el territorio de shark-τ y el mar de Ojotsk, este se le considera como el limbo ya que es un punto medio entre ambos territorios. En este terriotrio no existen muchos peligros ni seres importantes, aunque depredadores de capas inferiores suelen venir a buscar alimento. La vegetacion es la misma que en el mar de Ojotsk y no hay muchas diferencias geograficas entre ambas regiones. Se dice que cualquier aventurero pude ir a esta area con seguridad
* **El arrecife de la tentacion:** En esta capa es donde el ambiente se torna peligroso. Los animales se vuelven mas agresivos y territoriales. A los 2000 metros aproximadamente el arrecife se voldea y se convierte en lo que se le conoce como “el arrecife inverido”. En este, el ambiente se vuelve frio e inestable y depredadores muy intrepidos aparecen. Solo los aventureros experimentados puden venir a esta capa. En esta area se creo el campamento de observacion para aquellos que necesitan un descanso.
* **La gran falla:** La tercera capa es un largo decenso vertical, de forma tubular, dicha area se vuelve sumamente dificil de explorar. En las paredes del tubo existe un sistema de cuevas entrelazadas que funcionan como medida de descenso para aquellos que osen explorar este territorio. En el el centro de la tercera capa viven criaturas extremadamente rapidas debido a que se han acostumbrado a las grandes areas que hay que recorrer para conseguir comida. Solo aventureros que esten seguros de sus capacidades puden venir a explorar esta capa.
* **Caliz de los gigantes:** La cuarta y ultima capa es un extraña y oscuro ecosistema lleno de enredaderas que se entrelazan para formar discos concavos, se cree que en los discos concavos viven criaturas con piel extremadamente dura, esto se debe a lo profundo que se encuentra esta capa. El peligro de las bestias que habitan aqui tienen un nivel de peligro extremadamente alto. Solo aquellos viajeros los mas fuertes pueden entrar aqui.

**Descripcion del Boss:**

* **Nikikabane:** Nikikabane es un gran pez que habita en la primera capa. Su aspecto se caracteriza por tener grandes bigotes junto con colores blanco y negro dificiles de olvidar. Nikikabane es uno de los mas recientes secuaces de shark-τ. Nikikabane no es un contrincante especialmente fuerte.
* **Ottobasu:** Ottobasu es un delfin bastante inteligente. Este destaca mas que nada por su particular color rosa, que no te engañe su aspecto, Ottobasu es alguien que se especializa en manipular a su presa de diferentes maneras, a veces las hace dudar de sus capacidades, otras veces simplemente no tiene ganas de perder el tiempo y pueden acabar las cosas muy mal. Ottobasu es un antiguo ser marino casi extinto debido a su particular piel, el pasó a ser vasallo de shark-τ debido a que queria conservar su especie, ahora vive bajo la protección de shark-τ.
* **Mandibula Escarlata:** Mandibula Escarlata o tambien conocido como Splitjaw es una especie de anguila roja gigante que vive principalmente en los acantiados de la tercera capa. Mandibula escarlata como su nombre lo indica se caracteriza por tener la mandibula color carmesi o escarlata, el resto del cuerpo es color gris. Es un depredador feroz que atacan a cualquier presa a la vista. El es la mano derecha de Shark-τ, cuida que nadie entre a la cuarta capa y tambien es un gran combatiente, este ha ayudado a ganar gran parte de las guerras de shark-τ.
* **Shark-τ:** Shark-τ es un tiburon blanco que se caracteriza por su cuerpo fusiforme y gran robustesz, tiene una boca muy grande y cientos de dientes tan filosos como cualquier katana. Es ferozmente temperamental y territorial. Actualmente es el gobernante del mar de Ojotsk.

Diagrama

Descripción generada automáticamente

**Explicacion UML**

1. **Clase item**

**Imagen que contiene Diagrama

Descripción generada automáticamente**

Para esta parte del UML están definidos atributos básicos de un item: itemId (servirá como diferenciador de los objetos item, también se agregó este atributo a pesar de que no nos lo pedían en las especificaciones ya que pensé que podría ser más fácil de identificar los ítems por un id y no por un nombre), nombreItem, descripcionItem (Contendrá una pequeña descripción del item así como una pequeña referencia con respecto al lore) y precioItem. A parte de tener los getters y setters también contiene una función toString que juntara todos los atributos de la clase. Decidí implementar herencia en las subclases Poción, Pista y Especial ya que son tipos de ítems, cada uno tendrá su propia pool de ítems y harán diferentes cosas.

|  |  |  |
| --- | --- | --- |
| Poción | Skip | Especial |
| Las pociones modificarán tanto la vida actual del jugador (atributo vida, clase Jugador) como el daño (atributo respuestasCorrectas, clase Jugador). | El objeto skip saltara la pregunta, además proporcionara un +1 a respuestas correctas | El item especial no tendrá ningún uso dentro del juego, pero como Octo-π mencionó, él quería que su juego tuviera un sistema en donde pudieras comprar cosas fuera del juego, por lo tanto esto servirá como canje ante el profesor. |
| Poción - Minecraft Wiki | Hint Svg Png Icon Free Download (#386617) - OnlineWebFonts.COM |  |

**Objeto tipo Poción**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Ejemplo** |
| Item Id | Id del item, ayudara a asignar el item al inventario del jugador | 1 |
| Nombre Item | Nombre del item | Poción de Vida |
| Descripción  Item | descripción del item | La poción de vida restaura un punto a tu vida actual. |
| Precio Item | Precio del item, se descuenta de la Mora del jugador | 15 |
| Stat Up | Stat Up que otorga la pocin, la vida del jugador | 1 |

**Objeto tipo Skip**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Ejemplo** |
| Item Id | Id del item, ayudara a asignar el item al inventario del jugador | 2 |
| Nombre Item | Nombre del item | Skip |
| Descripción  Item | descripción del item | Skip te permite saltar de pregunta, además proporciona +1 a tus respuestas correctas |
| Precio Item | Precio del item, se descuenta de la Mora del jugador | 30 |
| Skip | Skip saltara la pregunta, además proporcionara un +1 a respuestas correctas | 1 |

**Objeto tipo Especial**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Ejemplo** |
| Habilidad Id | Id del item, ayudara a asignar el item al inventario del jugador | 1 |
| Habilidad Nombre | Nombre del item | Entrega tarea atrasada |
| Habilidad descripción | descripción del item | Este canje te permite entregar una tarea atrasada al profesor Octo-π |
| Habilidad Precio | Precio del item, se descuenta de la Mora del jugador | 15 |
| Item Especial | El item especial no tendrá ningún uso dentro del juego | Puedes entregar una tarea atrasada |

1. **Clase Habilidad**

Diagrama

Descripción generada automáticamente

La clase Habilidad es Composición de la clase Jugador a pesar de que la clase Item tiene atributos muy parecidos a la clase habilidad ya que considero que debe de tener otra clase debido a lo que hace referencia, la clase es composición de la clase Jugador ya que la clase habilidad no es dependiente de la clase Jugador, la vida de la clase Habilidad depende de la clase Jugador.

La clase habilidad tiene atributos similares a los de la clase item, por lo tanto, no es necesario describir cada uno de ellos, el atributo en el que se diferencia es powerUp, tiene como fin modificar stats como maxVida que representa las veces que te puedes equivocar al responder a la pregunta, por lo tanto, podrás equivocarte más veces, también modifica las el maxRespuestasCorrectas que representa la cantidad de respuestas correctas que tienes que responder para pasar al siguiente piso, por lo tanto es como si aumentaras tu daño.

Como métodos tiene sus respectivos getters y setters, constructor y una función toString, dicha función se repetirá en casi todas las clases, lo ideal sería definirla como función y así no tener que estar repitiendo código, el problema es que todas las clases tienen diferentes distribuciones de atributos y con mis conocimientos no encuentro la manera de implementarlo a pesar de que usted menciono que se podía hacer.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Ejemplo** |
| Habilidad Id | Id de la habilidad, ayudara a asignar la habilidad al inventario del jugador | 1 |
| Habilidad Nombre | Nombre de la habilidad | Corte de Superficie |
| Habilidad descripción | descripción de la habilidad | El usuario genera un corte poderoso formando una ola por el movimiento de su arma potenciando su ataque capaz de decapitar fácilmente a cualquier objetivo |
| Habilidad Precio | Precio de la habilidad, se descuenta de la experiencia del jugador | 15 |
| Power Up | Power Up que otorga la habilidad, modifica max Respuestas correctas | 0.2 |

1. **Clase Jugador**

Texto

Descripción generada automáticamente

La clase Habilidad es composición de la clase Jugador, Un objeto Jugador esta a su vez compuesto por uno o varios objetos de tipo habilidad, el tiempo de vida de los objetos habilidad depende de Jugador, si no existe un jugador no pueden existir sus habilidades, esta fue la lógica que seguí a la hora de asignar composición a esta clase. La clase Item es composición de la clase jugador ya que el jugador puede contener dichos item sin la necesidad de que dichos ítems estén atados a la vida de la clase Jugador.

|  |  |  |
| --- | --- | --- |
| **Atributos clase Jugador** | | |
| **Item** | Es un arreglo de apuntadores que tienen la intención de mostrar todos los atributos del item | Minecraft Potion Chart and Guide - Apex Hosting |
| **Habilidades** | Es un arreglo de apuntadores que tiene la intención de mostrar todos los atributos de la habilidad | Placeholder Image - Skill Icon Png Clipart (#5409408) - PinClipart |
| **Nombre** | Da nombre al jugador, hace más fácil encontrar el objeto | Chunchun |
| **Jugador Id** | Da un id al jugador | 1 |
| **Vida** | Vida actual del Jugador | 5 |
| **Max Vida** | Vida máxima que se puede almacenar | 6 |
| **Experiencia** | Experiencia del jugador, se consigue derrotando al Boss | 23 |
| **Mora** | Moneda del Juego, se consigue derrotando al Boss | 35 |
| **Respuestas Correctas** | Veces que has respondido correctamente a la pregunta del Boss | 3 |
| **Max Respuestas correctas** | Representa el numero de respuestas correctas que debes tener para derrotar el boss, se puede interpretar como tu daño | 6 |

Los métodos de la clase jugador son sus respectivos setters y getters, constructores y el destructor de la clase Jugador, existen funciones especiales como getItems (devuelve el item mediante un apuntador), addItem(agrega un item mediante un apuntador) y removeItem (que remueve el item del jugador), lo mismo pasa para las habilidades.

1. **Clase Piso**

**Tabla

Descripción generada automáticamente**

|  |  |  |
| --- | --- | --- |
| **Atributos clase Piso** | | |
| **Nombre Piso** | Nombre del Piso | El Borde del Abismo / El bosque de la tentación / La Gran Falla / Cáliz de los Gigantes |
| **Numero Piso** | Número del Piso | 1 / 2 / 3 / 4 |
| **Nombre Boss** | Nombre del Boss | Nikikabane / Ottobasu / Mandíbula Escarlata / Shark-τ |
| **Descripción Piso** | Descripción del Piso | --- |
| **Pregunta** | Array de preguntas | [“¿Cuánto es 12 + 49?”, “¿Cuánto es 45 - 32?”, “¿Cuánto es 28 / 4”, “¿Cuánto es 12\*14?”, “¿Carlos tiene 38.5 litros de aceite y ha llenado 8 botellas de tres cuartos de litro cada una. El resto lo pone en frascos de 125 ml cada una. ¿Cuántas frascas llenas?”,] |
| **Respuesta** | Array de respuestas, esta en formato string ya que varias respuestas no son numéricas | [“61”, “13”, “7”, “148”, “268”] |

1. **Clase System**

**Imagen de la pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza baja**

Las clases Piso y Jugador son agregación de la clase System, necesitan una forma de interactuar entre sí, todavía no poseo el conocimiento de definir sus funciones ni atributos.

**Cambios realizados (Cuarta entrega)**

Código:

-Se definieron todas las descripciones tipo descripcionPiso y descripcionBoss

-Se agrego función para que la consola escriba lento

-Se agrego la función encuentro definida en la clase Piso

-Se agrego función para tomar poción

-Se agrego función para atacar con habilidades

-Se implemento función para abrir archivos de texto

-Todos los aspectos estéticos fueron pulidos a detalle

-Se modificaron la mayoría de las funciones de la clase piso

-Valor respuestasCorrectas cambio a double para que pueda interactuar con las habilidades

Documento (Subrayado en amarillo):

-Se agrego sinopsis de la historia

-Se agrego las instrucciones del juego

-Se agrego la descripción de todos los pisos.

-Se agrego la descripción de todos los bosses

Notas: El primer encuentro ya esta listo, solo falta definir los 3 encuentros siguientes, esto no es gran problema ya que solo es darle los objetos piso y jugador para que realice los 3 encuentros siguientes, no resulta problema porque todas las descripciones ya están completas solo hace falta mandarlos a llamar. Lo anteriormente mencionado no se agrego por falta de tiempo.

Al usar las habilidades me sigue imprimiendo la dirección de memoria, no creo tardar mucho para que imprima el contenido, sin embargo, si se pueden usar habilidades.

El ultimo encuentro va a abrir una tienda especial donde puedas comprar los objetos especiales.

El proyecto creo que va por un 90% ya que lo más difícil fue hacer el encuentro como quería y ya está implementado

Falta actualizar UML