Fase 1 – Actividad práctica



**Análisis del Problema**

**Presentado por:**

Juan Diego Mejía Otálvaro

**Asignatura:**

Programación Orientada a Objetos

**Semestre:**

2018 – 01

Ingeniería de Sistemas

Universidad de Medellín

2018

|  |  |  |
| --- | --- | --- |
|  |  | **Control de cambios** |
|  |  |  |
| **Versión** | **Fecha** | **Descripción de cambios** |
| 1.0 | 12/02/2018 | Se crean los requisitos funcionales del juego |
| 1.1 | 14/02/2018 | Sustantivos y relaciones |
| 1.2 | 20/02/2018 | Corrección N-1 de Requisitos funcionales |
| 1.3 | 21/02/2018 | Corrección N-1 de Diagrama de clases |
| 1.4 | 26/02/2018 | Se describen Clases, atributos y se hace el diseño |
|  |  | del juego en GreenFoot |
| 2.0 | 17/04/2018 | Descomposición de requisitos funcionales |
| 2.1 | 1/05/2018 | Corrección completa de los requisitos funcionales |
| 2.2 | 4/05/2018 | Implementación completa del juego en Greenfoot |

Programación Orientada a Objetos

Ingeniería de Sistemas

Universidad de Medellín

Tabla de contenido



**1** [**DESCRIPCIÓN DEL PROBLEMA**](#1fob9te)[**3**](#1fob9te)

**2** [**REQUISITOS FUNCIONALES**](#3znysh7)[**4**](#3znysh7)

**3** [**MODELO DEL MUNDO DEL PROBLEMA**](#3znysh7)[**4**](#3znysh7)

[**3.1**](#35nkun2)[**Diagrama de clases**](#35nkun2)[**7**](#35nkun2)

[**3.2**](#26in1rg)[**Descripción de clases y atributos**](#26in1rg)[**8**](#26in1rg)

**4 IMPLEMENTACIÓN INICIAL DE CLASES Y DISEÑO DEL JUEGO**

Programación Orientada a Objetos

Ingeniería de Sistemas

Universidad de Medellín

**1 Descripción del problema**

Juego: GoRun Para este proyecto se busca construir un juego llamado GoRun en Greenfoot. La idea del juego es de un personaje llamado (Runner) que va a correr a lo largo de la pantalla (horizontalmente), tratando de cumplir el objetivo de recuperar una serie de tesoros y evitando enemigos durante un tiempo determinado. El personaje Runner siempre se verá ubicado en el centro de la pantalla. El efecto de correr estará dado por la animación del personaje y el desplazamiento de los elementos del fondo de la pantalla. El usuario puede controlar a Runner haciendo que éste salte o se agache para evitar obstáculos o enemigos. Las acciones de salto y de agacharse deben tener su animación correspondiente. Los obstáculos son bloques que van apareciendo por la derecha de la pantalla de forma aleatoria. Runner puede saltar sobre los bloques y pararse sobre ellos, pero si los choca de frente perderá una vida. En el juego también hay enemigos (deben existir al menos tres tipos de enemigos con dificultades diferentes) que aparecen cuando pasa un lapso de tiempo (dicho tiempo se debe determinar aleatoriamente). Si Runner se encuentra con un enemigo lo debe esquivar, si no lo esquiva y es tocado por éste, perderá una vida. Igualmente, el juego tiene unos tesoros que van apareciendo aleatoriamente por la derecha de la pantalla. Runner debe intentar capturar los tesoros. Cada que se captura un tesoro, se debe incrementar la cuenta de tesoros capturados. El juego debe tener una pantalla de inicio donde se tengan opciones para ver las instrucciones e iniciar el juego. El juego debe tener al menos 3 niveles. En cada nivel se debe incrementar la dificultad del juego, la cual dependerá del tiempo con el que se cuenta para terminar el nivel, el número de tesoros que debe capturar Runner y de la cantidad y el tipo de enemigos que aparecen. En cada nivel, Runner debe capturar cierta cantidad de tesoros antes de que se acabe el tiempo. Si Runner logra capturar todos los tesoros en el tiempo establecido, avanzará de nivel. Si no logra capturar los tesoros antes de que se acabe el tiempo, perderá una vida. Cuando Runner pierda su última vida, se terminará el juego mostrando un mensaje de “game over”. En cada nivel, se debe tener un contador de tiempo restante, un contador de vidas restantes y un contador de tesoros recogidos. Estos contadores deben estar visibles en algún lugar de la pantalla sin que obstruyan el juego. El juego se finaliza cuando Runner haya culminado todos los niveles. En ese punto, se debe mostrar un mensaje de felicitación. Cada nivel deberá tener un sonido de fondo. Igualmente, cada acción del juego deberá tener un sonido representativo (saltar, agacharse, chocar con un bloque, capturar un tesoro, ser tocado por un enemigo, perder una vida, pasar de nivel, game over, finalizar el juego, etc…).

**1 Requisitos funcionales**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Nombre** |  | RF-01 Iniciar Juego |
|  |  | **Resumen** |  | El usuario podrá iniciar el juego cuando este lo desee dentro del |
|  |  |  |  | tablero principal |
|  |  | **Entradas** |  | Selección de opción “Inicio de juego” |
|  |  | **Resultado** |  | * Se detiene el sonido de fondo del TableroPrincipal * Se agrega el nivel 1 al Mundo * Se agregan los bloques de plataforma y se posicionan en la parte inferior del mapa. * Se agrega a Runner al mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-02 Caer |
| **Resumen** | Runner caerá mientras que no se encuentre en una plataforma |
| **Entradas** | ninguna |
| **Resultado** | Runner cae hasta que se tope con la plataforma del suelo |
|  |  |
|  |  |
| **Nombre** | RF-03 Saltar |
| **Resumen** | El usuario podrá hacer que Runner salte, cuando se encuentre sobre plataforma y no haya orden de deslice |
| **Entradas** | Botón de salto (Flecha arriba) |
| **Resultado** | * Se le da a Runner la imagen de salto |
|  | * Se reproduce el sonido correspondiente a “saltar” * Se establece “imagen de salto”. |
|  |  |
| **Nombre** | RF-04 Deslizar (Agachar) |
| **Resumen** | El usuario podrá hacer que Runner se deslice sobre el suelo del mapa haciendo que este se agache un poco siempre y cuando este se encuentre sobre una plataforma |
| **Entradas** | Botón de agachar (Flecha abajo) |
| **Resultado** | * Se cambia a “imagen de deslice” * Se hace deslizar a Runner |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Nombre** |  | RF-05 Perder vida |
|  |  | **Resumen** |  | Runner pierde una vida cuando este es impactado por un enemigo |
|  |  |  |  | O un obstáculo. |
|  |  | **Entradas** |  | Ninguna |
|  |  | **Resultado** |  | -Runner pierde una vida que se verá reflejada en el contador que |
|  |  |  |  | Se muestra en la parte superior. |
|  |  |  |  | -Reproducir el sonido correspondiente a “pérdida de vida”.  - Se elimina el obstáculo o enemigo interceptado por Runner |
|  |  |  |  | - Si esta es la última vida de Runner, se ejecutará seguidamente el |
|  |  |  |  | RF-12(Perder juego) |

|  |  |
| --- | --- |
|  |  |
| **Nombre** | RF-06 Correr |
| **Resumen** | Runner tendrá la animación de correr la cual estará basada en 10 imágenes que cambiaran cada 2 iteraciones |
| **Entradas** | Ninguna |
| **Resultado** | * Runner correrá automáticamente mientras se encuentre en plataforma y no haya orden de agache |

|  |  |
| --- | --- |
| **Nombre** | RF-07 Capturar Tesoros |
| **Resumen** | Runner podrá capturar tesoros durante el transcurso del juego. |
| **Entradas** | Ninguna |
| **Resultado** | * Se suma + 1 al contador de tesoros * Se elimina el tesoro del mapa * Se refleja en el contador de tesoros del mapa * Se reproduce el sonido correspondiente a “Capturar tesoro” |

|  |  |
| --- | --- |
| **Nombre** | RF-08 Capturar vidas |
| **Resumen** | Runner podrá capturar las vidas que se generan durante el transcurso del juego |
| **Entradas** | Ninguna |
| **Resultado** | * Se suma +1 vida al contador de vidas * Se reproduce el sonido correspondiente a “Capturar vida” * Se elimina la vida del mapa * Se refleja el contador de vidas en el mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-09 Pasar de nivel |
| **Resumen** | Runner pasa al siguiente nivel cuando tiene los tesoros necesarios y |
|  | En el tiempo establecido. |
| **Entradas** | Ninguna |
| **Resultado** | * Se crea el nivel correspondiente y se agrega al mundo * Únicamente si es el nivel: 3 se ejecuta el requisito: Ganar Juego |

|  |  |
| --- | --- |
| **Nombre** | RF-10 Generar Enemigos |
| **Resumen** | Los enemigos se generan en el mundo en base a una cantidad de iteraciones y una probabilidad dada por el requisito: Establecer dificultad |
| **Entradas** | Ninguna |
| **Resultado** | * Se genera un enemigo y se posiciona en la parte inferior derecha del mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-11 Generar Tesoros |
| **Resumen** | Los tesoros se generan en el mundo en base a una cantidad de iteraciones y una probabilidad dada por el requisito: Establecer dificultad |
| **Entradas** | Ninguna |
| **Resultado** | * Se genera un tesoro y se posiciona en la parte inferior derecha del mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-12 Generar obstáculo 1 |
| **Resumen** | El obstáculo 1 se genera en el mundo en base a una cantidad de iteraciones y una probabilidad dada por el requisito: Establecer dificultad |
| **Entradas** | Ninguna |
| **Resultado** | * Se genera un obstáculo 1 y se posiciona en la parte inferior derecha del mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-13 Generar obstáculo 2 |
| **Resumen** | El obstáculo 2 se genera en el mundo en base a una cantidad de iteraciones y una probabilidad dada por el requisito: Establecer dificultad |
| **Entradas** | Ninguna |
| **Resultado** | * Se genera un obstáculo 2 y se posiciona en la parte inferior derecha del mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-14 Generar vidas |
| **Resumen** | Las vidas se generan en el mundo en base a una cantidad de iteraciones y una probabilidad dada por el requisito: Establecer dificultad |
| **Entradas** | Ninguna |
| **Resultado** | * Se genera una vida y se posiciona en la parte inferior derecha del mapa |

|  |  |
| --- | --- |
| **Nombre** | RF-15 Ganar juego |
| **Resumen** | Runner gana el juego cuando se hayan culminado todos los niveles |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se detiene el sonido de fondo * El juego genera un mensaje de “Ganaste el Juego |
|  | * Se reproduce el sonido correspondiente a “Ganar juego” |
|  | * Se detiene la ejecución del juego |
|  |  |
| **Nombre** | RF-12 Perder juego |
| **Resumen** | Runner pierde el juego cuando se queda sin vidas, o se le acaba el tiempo |
|  |  |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se detiene el sonido de fondo * El juego genera un mensaje de Game Over * Se reproduce el sonido correspondiente a “Perder juego” * Se detiene la ejecución del juego |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
| **Nombre** | RF-18 Reproducir sonido de fondo |
| **Resumen** | El mundo reproduce un sonido de fondo el cual dependerá del nivel en que Runner se encuentre |
|  |  |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se identifica el nivel y se procede a reproducir el sonido de fondo |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
| **Nombre** | RF-19 Establecer Dificultad |
| **Resumen** | A cada nivel se le establecerá una dificultad la cual estará basada en iteraciones y probabilidad que varían según el nivel en que se encuentre Runner haciendo que el juego sea más difícil o más fácil. Entre mayor sea el nivel mayor será la dificultad |
|  |  |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se establece la dificultad en el nivel correspondiente |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **re** | RF- 20 Mover Oponentes |
| **Resumen** | Los oponentes (Enemigos, obstaculo1 y 2) se mueven a una velocidad que depende de que adversario sea, cuando el oponente llegue al borde del mapa durante su movimiento este se eliminará |
|  |  |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se hace mover el oponente en dirección horizontal izquierda |
|  | * Se elimina si está al borde |
|  |  |

|  |  |
| --- | --- |
| **Nombre** | RF- 21 Mover Beneficios |
| **Resumen** | Los Beneficios (Tesoros y vidas) se mueven a igual velocidad, Cuando el Beneficio llegue al borde durante su movimiento este se eliminará |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se hace mover el Beneficio en dirección horizontal izquierda |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
| **Nombre** | RF-22 Mover Texto en esquinas |
| **Resumen** | Si uno de los textos se encuentra en las esquinas estos se moverán en dirección contraria al borde donde se encuentren y así sucesivamente cuando llegue a la otra esquina |
|  |  |
| **Entradas** | Ninguna |
|  |  |
| **Resultado** | * Se hace mover el texto * Se cambia el sentido hacia donde se dirige si llega a una esquina |
|  |  |
|  |  |

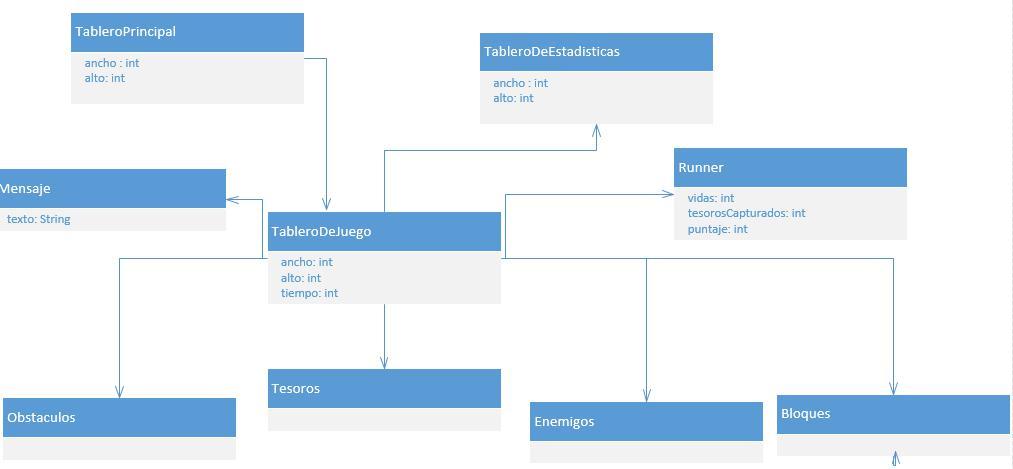
|  |  |
| --- | --- |
|  |  |
| **Nombre** | RF-23 Ver instrucciones |
| **Resumen** | El usuario podrá ver las instrucciones si así lo desea, además cuando se encuentre en la pantalla de instrucciones podrá volver a la pantalla principal con un botón que se le ofrece en la parte inferior |
|  |  |
| **Entradas** | Selección de “Ver instrucciones” |
|  |  |
| **Resultado** | * Se identifica muestran las instrucciones * Si el usuario selecciona volver se mostrará nuevamente la pantalla principal |
|  |  |
|  |  |

**2 Modelo del mundo del problema**

**2.1 Diagrama de clases**

* Runner
* Pantalla
* Tesoro
* Enemigo
* ~~Desplazamiento~~
* Obstáculos
* Enemigos
* Vida
* Tiempo
* ~~Cuenta~~
* Nivel
* Bloques
* Mensaje
* ~~Contador~~
* Texto

**PRIMER DIAGRAMA DE CLASES**



**2.2** **Descripción de clases y atributos**

**Clases**

TableroPrincipal: Es la interfaz principal donde el usuario tendrá la posibilidad de seleccionar la opción de “Iniciar juego” y también la de “Leer instrucciones de juego”.

TableroDeJuego: Es la interfaz a donde se redirige el usuario al seleccionar la opción “Iniciar juego” y ahora este podrá controlar a Runner dentro del mapa que ofrece este tablero.

Runner: Es la clase que representa al personaje controlado por el usuario (haciendo que este salte o se agache), el cual se encuentra en el tablero de juego y que cuenta con un total de 3 vidas en un principio para su supervivencia dentro del nivel que ofrece el tablero de juego.

Enemigos: Esta clase representa a los enemigos que se encuentran dentro del tablero de juego, los cuales son generados aleatoriamente y en una cantidad determinada por cada nivel, además estos se mueven alrededor de un espacio predefinido, dificultando al usuario el control de Runner durante la extensión de

mapa, ya que si hacen contacto con Runner, provocan que este pierda una vida.

Obstáculos: Esta clase representa a los obstáculos que se encuentran dentro del tablero de juego, los cuales son generados aleatoriamente y en una cantidad determinada por cada nivel, dificultando al usuario el control de Runner durante la extensión del mapa, ya que si Runner impacta contra estos, provocan que este pierda una vida.

Tesoros: Esta clase representa a los tesoros que se encuentran dentro del tablero de juego, los cuales son generados aleatoriamente y en una cantidad determinada por cada nivel, además estos pueden ser capturados por Runner con la finalidad de poder incrementar su puntaje y pasar al siguiente nivel.

Mensaje: Es el mensaje el cual se le remite al usuario mediante texto cuando este gana o pierde el juego.

**Atributos.**

altoDeTablero: Es el valor constante de tipo entero que se le da a la altura del tablero principal y de juego.

anchoDetablero: Es el valor constante de tipo entero que se le da al ancho del tablero principal y de juego.

vidasRunner: Es el valor contante de tipo entero que se le da a la cantidad de vidas que tiene Runner al iniciar el juego y el cual es mostrado en la parte superior derecha del tablero de juego y en el tablero de 9stadísticas.

puntajeRunner: Es la variable de tipo entero que va cambiando según el número de tesoros que Runner capture durante el nivel de juego el juego(cuando Runner capture un tesoros su puntaje se incrementará 10), y el cual es mostrado en la parte superior derecha del tablero de juego y en el tablero de estadísticas.

Texto: Es la variable de tipo cadena que se remite mediante un mensaje cual

puede ser: “Game Over”, si el usuario pierde, o “Felicitaciones, ¡Ganaste!”, si el usuario gana.

**3 Descomposición de requisitos funcionales**

Descomponga cada uno de los requisitos funcionales, indicando los pasos, la clase a la que pertenecería cada método asociado al paso y la justificación.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requisito 1: Iniciar juego** | |  |  |  |  |
| **Pasos** |  | **Clase a la que pertenece** | | **Justificación** | |
| 1.Identificar si hay orden de inicio de Juego |  |  | TableroPrincipal |  | El tablero principal es la clase quien recibe la orden de inicio |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 1.1 Detener sonido de fondo del Tablero Principal |  |  | TableroPrincipal |  | El objeto que contiene el sonido está definido en esta clase |
| 1.2 Agregar nivel número 1 (uno) |  |  | Tablero Principal |  | En esta clase se define si se agrega o no el nivel ya que es quien recibe la orden de inicio |
| 1.2.1 Agregar los bloques de plataforma |  |  | Mundos |  | Los bloques aparecerán en los Nivel\_1,2,3 que heredan de mundos |
| 1.2.2Agregar a Runner al mapa |  |  | Mundos |  | Runner aparecerá en las clases nivel\_1,2,3 que heredan de mundos |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Requisito 2: Caer** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si no se está | Runner | Runner contiene las coordenadas del objeto que se identifica |
| sobre una plataforma |  |  |
| 2. Hacer caer a Runner | Runner | Runner contiene las coordenadas del objeto que se hace caer |
|  |  |  |
| **Requisito 3: Saltar** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay orden de | Runner | A Runner es quien va dirigida la orden |
| salto |  |  |
| 1.1 Identificar que no haya orden de deslice | Runner | Runner es quien identifica si se está deslizando o no |
| 1.2 Identificar si se está sobre | Runner | Runner contiene las coordenadas del objeto que se identifica |
| Una plataforma |  |  |
| 1.2.1 Darle a Runner la | Runner | A Runner es a quien se le da la imagen de salto |
| imagen de salto |  |  |
| 1.2.2 Reproducir sonido de | Runner | Runner es quien identifica si hay salto |
| salto |  |  |
| 1.1.3 Hacer saltar a Runner | Runner | Runner contiene las coordenadas del objeto al que se hace saltar |
|  |  |  |
| **Requisito 4: Deslizar (agachar)** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay orden de | Runner | A Runner es quien va dirigida la orden |
| agache |  |  |
| 1.1 Identificar si se está sobre | Runner | Runner contiene las coordenadas del objeto que se identifica |
| plataforma |  |  |
| 1.1.1 Darle a Runner la | Runner | A Runner es a quien se le da la imagen de agache |
| imagen de agache |  |  |
| 1.1.2 Hacer agachar a Runner | Runner | Runner contiene las coordenadas del objeto al que se hace agachar |

|  |  |  |
| --- | --- | --- |
| **Requisito 5: Perder vida** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay choque | Runner | Runner contiene las coordenadas del objeto al que se identifica el impacto |
| con algún obstáculo o |  |  |
| enemigo |  |  |
| 1.1 Reproducir sonido de perder vida | Runner | Runner es quien identifica si hay choque |
| 1.2 Identificar si es la última | Runner | Runner es quien almacena las vidas |
| Vida |  |  |
| 1.2.1 Ejecutar requisito: | Mundos | Mundos es quien contiene el método de perder juego |
| Perder Juego |  |  |
| 1.3 Eliminar obstáculo o enemigo interceptado | Runner | Runner es quien identifica si hay choque |
| 1.4 Restar una vida | Runner | Runner es quien almacena las vidas |
| 1.4.1 Reflejar en el contador la vida restada | Mundos | El contador se refleja en el mundo |

|  |  |  |
| --- | --- | --- |
| **Requisito 6: Correr** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si se está sobre plataforma | Runner | Runner tiene las coordenadas del objeto que se identifica |
|  |  |  |
|  |  |  |
| 1.1 Identificar si no hay orden de deslice | Runner | A Runner es a quien se le da la orden de deslice |
| 1.1.1 Identificar si se cumplen las iteraciones para establecer imagen i de la animación | Runner | Runner es quien tiene el contador de iteraciones |
|  |  |  |
| 1.1.1.1 Establecer imagen de animación | Runner | Las imágenes se guardan en el arreglo que está contenido en esta clase |
|  |  |  |
|  |  |  |
| **Requisito 7: Capturar Tesoros** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay | Runner | Runner contiene las coordenadas del objeto al que se identifica la intersección |
| intersección con algún |  |  |
| tesoro |  |  |
| 1.1 Sumar +1 tesoro al contador de tesoros | Runner |  |
|  |  | Runner es quien almacena el contador de tesoros |
| 1.2 Eliminar objeto Tesoro del | Runner | Runner identifica si se captura el tesoro para proceder a eliminar el objeto de esta clase |
| mundo |  |  |
| 1.3 Reproducir sonido: Capturar Tesoro | Runner | Runner es quien identifica si se captura un tesoro para después proceder a reproducir el sonido |
| 1.4 Reflejar contador de tesoros capturados | Runner | Runner es quien contiene el contador de tesoros capturados |
|  |  |  |
|  |  |  |
|  |  |  |
| **Requisito 8: Capturar Vidas** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay | Runner | Runner contiene las coordenadas del objeto al que se identifica la intersección |
| intersección con alguna vida |  |  |
|  |  |  |
| 1.1 Sumar +1 vida al contador de vidas | Runner |  |
|  |  | Runner es quien almacena el contador de vidas |
| 1.2 Eliminar objeto vida del | Runner | Runner identifica si se captura la vida para proceder a eliminar el objeto de esta clase |
| mundo |  |  |
| 1.3 Reproducir sonido: Capturar vida | Runner | Runner es quien identifica si se captura una vida para después proceder a reproducir el sonido |
| 1.4 Reflejar contador de vidas | Runner | Runner es quien contiene el contador de vidas |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Requisito 9: Pasar de nivel** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si Runner se encuentra en el nivel 1 | Mundos | Mundos es quien almacena el nivel de juego en el que se encuentra Runner |
|  |  |  |
|  |  |  |
| 1.1 Identificar si Runner tiene los tesoros necesarios para pasar el nivel 1 | Runner | Runner es quien almacena la cantidad de tesoros |
|  |  |  |
|  |  |  |
| 1.1.1 Crear nivel 2 y agregarlo al mundo | Mundo | El mundo es el que cambia |
| 2. Identificar si Runner se encuentra en el nivel 2 | Mundos | Mundos es quien almacena el nivel de juego en el que se encuentra Runner |
| 2.1 Identificar si Runner tiene los tesoros necesarios para pasar el nivel 2 | Runner | Runner es quien almacena la cantidad de tesoros |
|  |  |  |
| 2.1.1 Crear nivel 3 y agregarlo al mundo | Mundo | El mundo es el que cambia |
|  |  |  |
| 3. Identificar si Runner se encuentra en el nivel 3 | Mundos | Mundos es quien almacena el nivel de juego en el que se encuentra Runner |
| 3.1 Identificar si Runner tiene los tesoros necesarios para pasar el nivel 3 | Runner | Runner es quien almacena la cantidad de tesoros |
| 3.1.1 Ejecutar Requisito: Ganar Juego | Mundos | Es la clase quien contiene el método de ganar juego |

|  |  |  |
| --- | --- | --- |
|  | | |
| **Requisito 10: Generar enemigos** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de iteraciones de enemigos es igual a las iteraciones por enemigo. | Mundos | Mundos contiene el contador de iteraciones de enemigo |
|  |  |  |
| 1.1 Identificar si la probabilidad generada es menor o igual a la establecida para generar el enemigo | Mundos | Mundos es la clase la cual genera la probabilidad para generar enemigo |
|  |  |  |
|  |  |  |
| 1.1.1. Generar enemigo y agregarlo al mundo | Mundos | El enemigo aparece en los respectivos niveles que hereda de mundos |

|  |  |  |
| --- | --- | --- |
| **Requisito 11: Generar Tesoros** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de iteraciones de tesoros es igual a las iteraciones por enemigo. | Mundos | Mundos contiene el contador de iteraciones de tesoro |
|  |  |  |
| 1.1 Identificar si la probabilidad generada es menor o igual a la establecida para generar el tesoro | Mundos | Mundos es la clase la cual genera la probabilidad para generar tesoro |
|  |  |  |
|  |  |  |
| 1.1.1. Generar tesoro y agregarlo al mundo | Mundos | El tesoro aparece en los respectivos niveles que hereda de mundos |

|  |  |  |
| --- | --- | --- |
| **Requisito 12: Generar Obstáculo N.1** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de iteraciones de obstaculo1 es igual a las iteraciones por obstaculo1. | Mundos | Mundos contiene el contador de iteraciones de obstaculo1 |
|  |  |  |
| 1.1 Identificar si la probabilidad generada es menor o igual a la establecida para generar el obstaculo1 | Mundos | Mundos es la clase la cual genera la probabilidad para generar el obstaculo1 |
|  |  |  |
|  |  |  |
| 1.1.1. Generar obstaculo1 y agregarlo al mundo | Mundos | El obstaculo2 aparece en los respectivos niveles que hereda de mundos |

|  |  |  |
| --- | --- | --- |
| **Requisito 13: Generar Obstáculo N.2** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de iteraciones de enemigos es igual a las iteraciones por obstaculo2. | Mundos | Mundos contiene el contador de iteraciones de obstaculo2 |
|  |  |  |
| 1.1 Identificar si la probabilidad generada es menor o igual a la establecida para generar el obstaculo2 | Mundos | Mundos es la clase la cual genera la probabilidad para generar obstaculo2 |
|  |  |  |
|  |  |  |
| 1.1.1. Generar obstaculo2 y agregarlo al mundo | Mundos | El obstaculo2 aparece en los respectivos niveles que hereda de mundos |

|  |  |  |
| --- | --- | --- |
| **Requisito 14: Generar vidas** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de iteraciones de vidas es igual a las iteraciones por vidas. | Mundos | Mundos contiene el contador de iteraciones de vidas |
|  |  |  |
| 1.1 Identificar si la probabilidad generada es menor o igual a la establecida para generar la vida | Mundos | Mundos es la clase la cual genera la probabilidad para generar vidas |
|  |  |  |
|  |  |  |
| 1.1.1. Generar vida y agregarla al mundo | Mundos | La vida aparece en los respectivos niveles que hereda de mundos |

|  |  |  |
| --- | --- | --- |
| **Requisito 15: Ganar juego** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Detener sonido de fondo | Mundos | Desde mundos es donde se reproduce el sonido de fondo |
| 2. Reproducir sonido de: Ganar Juego | Mundos | En mundos se identifica si se gana el juego |
|  |  |  |
|  |  |  |
| 3. Mostar texto de: Ganar juego | Mundos | En Mundos aparece el texto |
| 4. Parar ejecución del juego | Mundo | El mundo es quien se deja de ejecutar |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Requisito 16: Perder Juego** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Detener sonido de fondo | Mundos | Desde mundos es donde se reproduce el sonido de fondo |
| 2. Reproducir sonido de: Perder juego | Mundos | En mundos se identifica si se gana el juego |
|  |  |  |
|  |  |  |
| 3. Mostar texto de: Perder juego | Mundos | En Mundos aparece el texto |
| 4. Parar ejecución del juego | Mundo | El mundo es quien se deja de ejecutar |

|  |  |  |
| --- | --- | --- |
| **Requisito 17 : Reproducir sonido de Fondo** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si en nivel de juego es 1 | Mundos | Mundos almacena el nivel de juego |
|  |  |  |
| 1.1 Reproducir sonido de nivel 1 | Mundos | En mundos se encuentra el objeto de sonido |
|  |  |  |
| 2. Identificar si en nivel de juego es 2 | Mundos | Mundos almacena el nivel de juego |
|  |  |  |
| 2.1 Reproducir sonido de nivel 2 | Mundos | En mundos se encuentra el objeto de sonido |
| 3. Identificar si en nivel de juego es 3 | Mundos | Mundos almacena el nivel de juego |
| 1.1 Reproducir sonido de nivel 3 | Mundos | En mundos se encuentra el objeto de sonido |

|  |  |  |
| --- | --- | --- |
| **Requisito 18: Contar Tiempo** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el contador de tiempo es igual a 60 | Mundos | El contador de tiempo se encuentra en Mundos |
|  |  |  |
| 1.1 Restar un segundo | Mundos | Los segundos se encuentran en mundos |
|  |  |  |
| 2. Mostrar el tiempo restante en el mapa | Mundos | El tiempo restante está en mundos |
| 3. Identificar si los segundos han llegado a 0 | Mundos | Los segundos se encuentran en Mundos |
| 4. Ejecutar requisito: Perder Juego | Mundos | El método de perder juego está en Mundos |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Requisito 19: Establecer dificultad** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Establecer dificultad nivel 1 | Nivel\_1 | Los aspectos que se establecen se encuentran en esta clase |
| 2. Establecer dificultad nivel 2 | Nivel\_2 | Los aspectos que se establecen se encuentran en esta clase |
|  |  | Los aspectos que se establecen se encuentran en esta clase |
| 3. Establecer dificultad nivel 3 | Nivel\_3 |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Requisito 20: Mover Oponentes** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Mover oponentes | Oponentes | De esta clase heredan todos Enemigos, obstáculo1 y 2 |
| 1.1 Mover Enemigos | Enemigo | Allí se encuentra la velocidad en la que se mueve este enemigo |
|  |  |  |
|  |  |  |
| 1.2 Mover a Obstaculo1 | Obstaculo1 | Allí se encuentra la velocidad en la que se mueve este obstáculo |
| 1.3 Mover obstáculo2 | Obstaculo2 | Allí se encuentra la velocidad en la que se mueve este obstáculo |
| 2. Identificar si llega al borde | Oponentes | Esta clase contiene las coordenadas de los Enemigos y obstáculos |
| 2.1 Eliminar del mapa |  | Estos objetos se encuentran en el mundo |
|  | Mundo |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Requisito 21: Mover Beneficios** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Mover vidas y tesoros | Beneficios | Allí se encuentra la velocidad en la que se mueven estos objetos |
|  |  |  |
|  |  |  |
| 2. Identificar si llega al borde | Beneficios | Esta clase contiene las coordenadas de vidas y tesoros |
| 2.1 Eliminar del mapa |  | Estos objetos se encuentran en el mundo |
|  | Mundo |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Requisito 22: Mover texto en esquinas** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si el texto se encuentra en una esquina | Texto | Esta clase posee las coordenadas de todos los textos que hay en el juego |
| 1.1 Mover texto | Texto | Allí se encuentra la velocidad en la que se mueve el texto |
|  |  |  |
|  |  |  |
| 1.1.1. Identificar si está en el borde | Texto | Esta clase contiene las coordenadas de todos los textos |
| 1.1.1.1 Cambiarle el sentido de movimiento |  | Allí se encuentra la velocidad en la que se mueve el texto |
|  | Texto |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Requisito 23: Ver instrucciones** |  |  |
| **Pasos** | **Clase a la que** | **Justificación** |
|  | **pertenece** |  |
| 1. Identificar si hay orden de ver instrucciones | Pantalla Principal | Es donde está contenido el botón de instrucciones |
| 1.1 Establecer pantalla de instrucciones | Pantalla principal | Es donde se identifica si hay orden |
|  |  |  |
|  |  |  |
| 1.1.1 Identificar si hay orden de volver | InstruccionesDelJuego | Es donde está contenido el botón de instrucciones |
| 1.1.1.1 Establecer pantalla principal | InstruccionesDelJuego | Es donde se identifica si hay orden |
|  |  |  |
|  |  |  |

**4 Definición de los contratos de los métodos**

Defina los contratos de los métodos que surgieron de la descomposición de requisitos funcionales. Esto se debe hacer directamente en el código.

**5 Versión final del modelo de clases**

**EL DIAGRAMA DE CLASES FINAL ESTÁ EN LA CARPETA EN UN ARCHIVO VISIO**

Programación Orientada a Objetos

Ingeniería de Sistemas

Universidad de Medellín

Programación Orientada a Objetos

Ingeniería de Sistemas

Universidad de Medellín

15

Programación Orientada a Objetos

Ingeniería de Sistemas

Universidad de Medellín