

MANUAL TÉCNICO JUEGO “SURVIVALVILLE”

Por: Leví Isaac Hernández Sapón
Carné: 201930693

Manual Técnico

El software creado es en esencia un juego de supervivencia. La mecánica es sencilla, intenta sobrevivir la mayor cantidad de tiempo posible. Para esto deberás comprar semillas, animales y barcos que te ayudarán a obtener productos con los que se alimentará para recuperar vida o también puede venderlos para generar dinero y así poder comprar más cosas.

Este software fue diseñado con el lenguaje Java, específicamente con el JDK 11 y se utilizó el IDE de Visual Studio Code.

Clase Menú (Interfaz)

Clase que contiene botones para que el usuario se vaya trasladando por cada ventana del juego. En este método se encuentran declarados los animales (Herbívoros y Omnívoros), Semillas (Granos y Frutos) y los productos de cada uno de ellos.

Métodos:

Constructor, Métodos para colocar labels, botones y métodos para sus ActionListener, también se cuenta con métodos para redimensionar animales, plantas y productos.

Métodos para redimensionar: estos métodos son los encargados de aumentar el tamaño de los arreglos e ingresar el nuevo Objeto.

MenuCrear: Este método nos envía a tres distintas ventanas dependiendo de la selección. 1. Crear Animal, 2. Crear Planta y 3. Crear Producto. La opción 4 es para volver al menú principal.

Clase Reporte (Interfaz)

Clase encargada de mostrar los reportes del jugador, animales y semillas.

Métodos:

Contiene todos los métodos necesarios para agregar un JPanel, radioButtons, un Jtable y todos sus respectivos oyentes de acción, así como un botón que nos permitirá volver al menú.

Clases Crear (Interfaz)

Entre estas clases se encuentran: CrearAnimal, CrearSemilla y CrearProducto, sus métodos son todos aquellos para colocar los botones, etiquetas y cajas de texto en el JFrame. Además cuenta con métodos para poder asignar la información necesaria en variables que se utilizarán para crear objetos dependiendo de la ventana en la que se encuentre. Siendo objetos de animal, semilla y producto en las clases CrearAnimal, CrearSemilla y CrearProducto respectivamente.

Clase Inicio (Interfaz)

Interfaz que sirve para poder almacenar la información relevante del usuario, siendo esta el nombre y su nickname.

Métodos:

Contiene los métodos básicos para el uso de etiquetas, cajas de texto y un checkButton usado para aceptar los términos y condiciones del juego. Además contiene un ActionListener para el botonInicio y

uno para el botonVolver, El botonInicio almacena el nombre y nickName en un atributo de tipo Jugador e inicia una ventana que contiene el Juego, el botonVolver regresa al menú.

Clase Juego (Interfaz)

Clase que se encarga de llevar la dinámica del juego.

Métodos:

contiene métodos para iniciar todos los componentes necesarios. Siendo estos ToogleButtons, etiquetas, botones, etc.

Método colocarTerreno: Método que llama a los métodos generarTerreno y generarPosicion dentro de 2 ciclos que envía como parámetros (i, j) que es la posición en la matriz del terreno y objeto que quiere agregar al panel. Además de esto es el encargado de asignarle el respectivo oyente a las etiquetas.

Método generarTerreno: Este método es el encargado de generar las instancias de los terrenos de forma aleatoria.

Método generaPosicion: Este se encarga de colocar las etiquetas de los terrenos y los objetos en un espacio del JFrame y agregarlas al mismo.

Método oyenteTerreno: Método que le agrega un oyente de Mouse a la etiqueta cuando esté seleccionado el toogleButton “verTerreno”, la acción dependerá si la instancia de la etiqueta de Terreno es Grama, Lago o Parcela. También les agrega una acción a los toogleButtons “comprarTerreno” y “crearParcela”.

Método accionesLago: Este método usa enums para ver el estado del Lago para saber si puede colocar un barco para pescar, si ya hay un barco en ese lago, si no tiene peces o si el barco ya terminó de pescar todos los peces. Dependiendo del estado en el que se encuentre va a ser la tarea que realizará.

Método accionesParcela: Este método usa enums para saber el estado de la Parcela. Con esto se sabrá si se puede colocar un animal para criar, si el espacio ya está siendo ocupado por un animal, si el animal ya está listo para destazarlo o si hay algún animal muerto. Dependiendo del estado en el que se encuentre es la tarea que realizará.

Método accionesGrama: Método que usa enums para saber el estado de la Grama. Con esto se sabrá si se encuentra disponible, si ya hay una siembra en proceso, si el Grano/Fruto ya ha generado productos o si hay una siembra muerta. La acción que realice va a depender del estado en el que se encuentre dicho terreno.

Método accionCompra: Método que le permite al usuario comprar nuevos terrenos para aumentar su territorio y poder generar más productos.

Método comprarParcela: Este método es el encargado de convertir Grama a Parcela y se pueda utilizar para criar animales.

Método actualizarCeldasSembradas: Como indica el nombre, este método estático solo aumenta el número de celdas sembradas dependiendo de la semilla que haya sido cultivada.

Método actualizarAnimalesDestazados: Método que incrementa el número de animales destazados dependiendo del animal que haya sido criado.

Clase Animal (Padre)

Clase abstracta que hereda de Thread para los hilos de vida de los animales, contiene los atributos como Vida, nombre, precio, tipo (TipoProducto) y un arreglo de productos iniciales de 0.

Métodos:

Contiene los métodos getters y setters. También cuenta con el método constructor que inicializa el nombre, la vida, precio y tipo. Un método oculto que inicializa cantidadAnimales, criasDestazadas, y productos en 0.

Método agregarProducto: Este método es el encargado de incrementar el número de productos si se le intenta agregar un nuevo producto o actualizar uno que el animal ya posea.

Método elegirTerreno: Este método solo hace referencia al espacio en memoria que ocupa el terreno en el que el animal está siendo colocado así como el objeto (etiqueta) que también se encuentra en esa posición.

Clase Herbivoro (Hija)

Clase que hereda de Animal, como atributos cuenta con objetos tipo ImageIcon que serán utilizados para colocarlos en el objeto(etiqueta).

Métodos:

Contiene el método constructor, el método Run heredado de Animal(que a su vez hereda de Thread) para la vida del animal y claramente todos los métodos heredados de Animal.

Método toString: Método que hereda de Object que retorna el nombre.

Método colocarImagen: Este método le coloca una imagen al objeto(etiqueta) dependiendo del estado en el que se encuentre el hilo.

Clase Omnivoro (Hija)

Clase que hereda de Animal, como atributos cuenta con objetos tipo ImageIcon que serán utilizados para colocarlos en el objeto(etiqueta).

Métodos:

Contiene el método constructor, el método Run heredado de Animal(que a su vez hereda de Thread) para la vida del animal y claramente todos los métodos heredados de Animal.

Método toString: Método que hereda de Object que retorna el nombre.

Método colocarImagen: Este método le coloca una imagen al objeto(etiqueta) dependiendo del estado en el que se encuentre el hilo.

Clase Planta (Padre)

Clase abstracta que hereda de Thread, como atributos tiene nombre, vida, precio, cantidadAdquirida, celdasSembradas, y producto.

Métodos:

Cuenta con un constructor que inicializa nombre, vida y precio, métodos getters y setters. método toString, métodos abstractos run y colocarImagen.

Método elegirTerreno: hace referencia al terreno en el que se va a sembrar la planta y también al espacio en memoria del objeto(etiqueta) que se encuentra en esa misma posición.

Método agregarProducto: le da un valor al producto que va a producir dicha planta.

Método productoAsignado: Verifica que el producto no haya sido asignado, si ya se ha asignado un producto retorna un valor verdadero, de lo contrario retorna falso.

Clase Grano (Hija)

Clase que hereda de Planta, como atributos además de tener los que hereda de la clase padre, tiene tres objetos de tipo ImageIcon que se le asignan al objeto(etiqueta) para poder diferenciar el estado en el que se encuentra el terreno.

Métodos:

Implementa el método run que hereda de su padre que a su vez hereda de Thread.

Método run: crea un hilo en el que se descuenta la vida de la planta cada segundo y al finalizar genera todos los productos a cargo de la misma.

Método colocarImagen: Método heredado de la clase abstracta Planta, este método solo sirve para hacer el cambio de ícono del objeto(etiqueta) dependiendo del estado en el que se encuentre el hilo.

Clase Fruto (Hija)

Clase que hereda de Planta, como atributos además de tener los que hereda de la clase padre, tiene tres objetos de tipo ImageIcon que se le asignan al objeto(etiqueta) para poder diferenciar el estado en el que se encuentra el terreno y un atributo privado tipo entero cantidadProducto.

Métodos:

Implementa el método run que hereda de su padre que a su vez hereda de Thread.

Método run: este método crea un hilo en el que se van a ir sumando el número de productos (cantidadProductos) con forme vaya avanzando y el usuario pueda ir recogiendo dichos productos antes de que muera la planta.

Método colocarImagen: Método heredado de la clase abstracta Planta, este método solo sirve para hacer el cambio de ícono del objeto(etiqueta) dependiendo del estado en el que se encuentre el hilo.

Clase Barco

Clase que hereda de Thread que posee como atributos pecesObtenidos de tipo entero, peces de tipo Alimento, estado de tipo EstadoBarco, y lago de tipo Lago de tipo privado, además de contar con dos objetos tipo ImageIcon para el estado del lago.

Métodos:

Contiene métodos sencillos como el constructor, getters, entre otros.

Método cambiarEstado: este es utilizado para cambiar el estado del barco.

Método run: método que sirve como hilo, cambia el estado del barco cuando comienza a pescar así como el del lago en donde se encuentra pescando en ese momento. La incrementación de peces es cada segundo así como la disminución de peces en el lago, al llegar a cero peces en el lago termina el método generando un objeto de tipo Alimento llamado peces y cambiando el estado del lago así como la imagen del objeto(etiqueta).

Método colocarImagen: al igual que todos los anteriores métodos este método es utilizado para cambiar el ícono del objeto(etiqueta) dependiendo del estado del hilo.

Clase Mercado

Clase diseñada para hacer las transacciones con el usuario, en este apartado podrá obtener animales, semillas o barcos. Contiene como atributos un objeto de tipo Barco y una constante de tipo Double llamada precioBarco con un valor de 1500.

Métodos:

Método estático menuMercado: este método muestra en un JoptionPane un mini Menú en el que el usuario podrá navegar dependiendo de la opción que elija.

Método estático subMenuAnimales: método que por medio de un JoptionPane le muestra al usuario distintas opciones, siendo estas comprar animales herbívoros, comprar animales omnívoros o alimento para estos.

Método estático comprarAnimales: método que por medio de un JoptionPane le muestra al usuario todos los animales disponibles que puede comprar. Dependiendo de la opción que elija va a ser el animal que se le asigne al usuario y el oro que se le disminuirá.

Método presentarAnimales: método que mediante un ciclo almacena la información de cada animal en el juego en una variable de tipo String que será retornada.

Método estático agregarProducto: este método asigna todos los productos que el animal produce dependiendo del animal que el usuario haya comprado ya que se crea el animal desde cero creando un nuevo espacio en memoria, los animales dentro del juego solo funcionan como molde para la compra de animales.

Método estático subMenuSemillas: Método que por medio de un JoptionPane muestra un menú en el que el usuario puede navegar entre diversas opciones.

Método estático comprarSemillas: por medio de un JoptionPane el usuario puede elegir la semilla que desea comprar. Este método crea la semilla desde 0 y le asigna el producto que esta produce. El arreglo de semillas del juego son el molde con el que se crean las semillas que va a plantar el jugador.

Método estático presentarSemillas: método que mediante un ciclo almacena en una variable tipo String toda la información de las semillas del juego. Dicha variable String será retornada.

Método comprarBarco: el usuario mediante un JoptionPane debe ingresar si quiere comprar un nuevo barco. Al confirmar se crea un objeto de tipo barco y le es asignado al usuario.

Clase Bodega

Clase que muestra toda la información de los productos, animales o barcos que posee el jugador en turno. Contiene un único atributo para las transiciones entre el menú.

Métodos:

Método estático bodega: crea un menú por medio de un JoptionPane. El usuario puede moverse a través del menú dependiendo de su elección.

Método estático subMenuAnimales: menú que muestra toda la información relevante de los animales que el jugador posee.

Método estático presentarAnimales: muestra toda la información de los animales que tiene el usuario en posesión.

Método estático subMenuSemillas: menú que muestra toda la información de las semillas que el jugador posee.

Método estático presentarSemillas: método que almacena en una variable String la información de las semillas que el jugador posee y que retorna dicha variable.

Método estático subMenuBarco: menú que muestra toda la información de los barcos que posee el jugador.

Método estático presentarBarco: almacena mediante una variable String todos los datos de los barcos que el jugador posee.

Método estático subMenuProductos: muestra todos los productos que ha generado el jugador a lo largo de la partida.

Método estático subMenuComer: método parecido al subMenuProductos con la única diferencia de que solo muestra los productos que sean alimento.

Método estático presentarProducto: muestra los productos que posee el jugador dependiendo de la elección que haya hecho. Si accede al menú comer solo almacena los productos que sean un alimento. De lo contrario muestra todos.

Método estático elegirCantidad: mediante un JoptionPane se ingresa la cantidad de producto que se desea vender/comer dependiendo del sub menú en el que se encuentre. Dependiendo del menú en el que se encuentre también se realizarán distintas acciones, siendo esta vender en el caso de estar en el subMenuProductos y comer en el caso de que se encuentre en el subMenuComer.

Clase Producto (Padre)

Clase que tiene como atributos nombre, precio, cantidad, y tipoProducto siendo este último de tipo TipoProducto (enum).

Métodos:

métodos getters y toString, también cuenta con dos métodos llamados restarCantidad y aumentarCantidad en los que se sobre entiende su función.

Clase Alimento (Hija)

Clase que hereda de Producto, cuenta con un atributo propio llamado vidaRecuperada.

Métodos:

Además de contar con los que hereda de Producto tiene un getVida.

Clase Materia (Hija)

Clase que hereda de Producto.

Métodos:

Solo cuenta con los métodos heredados de la clase Producto.

Clase Terreno (Padre)

Clase abstracta que hereda de Jlabel.

Métodos:

posee un único método abstracto llamado generarTerreno.

Clase Grama (Hija)

Clase que hereda de Terreno. Como atributos posee un objeto de tipo ImageIcon, un objeto llamado estado de tipo EstadoGrama (enum) y un objeto semilla de tipo Planta.

Métodos:

posee el método que hereda de Terreno y su función es cambiar la imagen del Jlabel. El constructor que solo cambia el estado inicial del terreno, getters y setters y un método cambiarEstado que cambia el estado del terreno.

Clase Lago (Hija)

Clase que hereda de Terreno e implementa Runnable. Como atributos posee un objeto de tipo ImageIcon, un objeto llamado estado de tipo EstadoLago (enum) y y diversos atributos de tipo entero.

Métodos:

Constructor que inicializa componentes y llama a un método llamado generarPeces, getters y setter, un método llamado cambiarEstado que cambia el estado del lago. También posee el método que hereda de Terreno y su función es cambiar la imagen del Jlabel.

Método generarPeces: crea un objeto de tipo Thread con parámetro “este” lago inicia el hilo.

Método run: crea un ciclo que duerme cada segundo, su función es llenar el lago de peces y al finalizar dicha tarea cambia el ToolTipText así como el estado del lago.

Clase Desierto (Hija)

Clase que hereda de Terreno. Como único atributo posee un objeto de tipo ImageIcon.

Métodos:

posee el método que hereda de Terreno y su función es cambiar la imagen del Jlabel.

Clase Parcela (Hija)

Clase que hereda de Terreno. Como atributos posee un objeto de tipo ImageIcon, un objeto llamado estado de tipo EstadoParcela (enum) y un objeto animal de tipo Animal.

Métodos:

posee el método que hereda de Terreno y su función es cambiar la imagen del Jlabel. Además posee métodos getters y setter y un método para cambiar el estado de la parcela llamado cambiarEstado.

Clase Jugador

Clase que contiene muchos atributos como lo son el nombre, nickName, oro, vida, un arreglo de animales, un arreglo de semillas, uno de productos y uno de barcos, tiempoDurado oroGenerado, alimentoGenerado, alimentoConsumido y celdasCompradas.

Métodos:

Constructor que inicializa cada componente, métodos getters y setters, método toString y métodos aumentarOro y disminuirOro. Los métodos agregarAnimal, agregarPlanta y agregarBarco solo ingresan valores a sus respectivos arreglos. Los métodos eliminarSemilla y eliminarAnimal, como indica su nombre solo son los encargados de eliminar la semilla y animal elegidos de sus arreglos respectivamente.

Método comer: aumenta la vida del usuario dependiendo de la comida que haya ingerido, aumenta el contador de alimento consumido y resta la cantidad de comida de la bodega, al hacerlo verifica si la comida ya se ha consumido toda.

Método vender: aumenta el oro del jugador dependiendo del producto que haya vendido, aumenta el contador de oroGenerado y se resta la cantidad de producto de la bodega, al hacerlo verifica si el producto ya se ha vendido todo.

Método agregarProducto: método encargado de ingresar los productos producidos por los animales o semillas al arreglo de productos, en el caso de que ya se haya agregado anteriormente un producto con el mismo nombre solo aumenta el número de productos. Si el producto ingresado es un alimento aumenta el contador de alimento generado.

Método colocarBarco: Este llama a otro llamado verificarBarco.

Método verificarBarco: Este método revisa si el jugador posee barcos disponibles para poder colocar uno en el lago.

Método verificarProductoTerminado: Este revisa si el producto ingresado tiene una cantidad de cero, de serlo lo borra del arreglo de productos.

Método colocarSemilla: Inicia el hilo de la semilla elegida por el usuario y la borra del arreglo de semillas, además de esto actualiza el valor de celdas sembradas llamando al método actualizarCeldasSembradas.

Método colocarAnimal: Inicia el hilo del animal elegido por el usuario y lo borra del arreglo de animales.