Granjeros v1.0

Padron: 100137
Nombre/s y Apellido/s:Jonathan Ezequiel Alejandro Rolon
Materia: Algoritmos y programacion II – Cátedra Calvo

Manual del programador

INDICE

1.Lenguaje utilizado

En este caso se utilizó el lenguaje C++, lenguaje determinado por la cátedra.

2. Tipos de datos y estructuras

#Tipos string para las opciones ingresadas por el usuario.
#Tipos char para matrices de un solo carácter como matriz de cultivos, matriz de tiempo de recuperación, matriz de tiempo hasta cosechar la parcela.
#Tipos booleanos para matriz de estado de riego de una parcela, para ciertas variables que determinaban el curso del programa como por ejemplo la variable bool finJuego que determina si se quiere quitar el juego o no.

#Tipos unsigned int para cantidades o pasar de string a int las opciones de fila y columna ingresadas por el usuario.
#structs para determinar los atributos del jugador como turnos de juego del jugador, cantidad de monedas, cantidad de semillas de un cultivo especifico, también para determinar los atributos de un cultivo.

3.Pasajes a funciones

Pasajes por valor: se utilizaron pasajes por valor en el caso de procedimientos que deben imprimir algo por pantalla por ejemplo.
Pasajes por referencia: se utilizaron para pasar la dirección de booleanos a modificar por ejemplo, o para pasar los vectores o matrices.

4.Archivos

Se utilizó un archivo “cultivos.dat” que contiene registros para captar en el vector de cultivos en el programa los atributos de cada cultivo. En eclipse, este archivo se ubica en el Workspace y de ahí en la carpeta del proyecto. Cabe aclarar que el programa no leerá el archivo si se coloca este en la carpeta src donde están los archivos .cpp y .h . Compilando con el compilador de Linux alcanza con colocar el archivo en la misma carpeta de los demás archivos.

4.1. Archivos del juego:

>“granjeros.cpp”: Solo tiene el main y en él la llamada a la función general que desarrolla el juego.
>”granjerosGame.cpp”: Tiene las funciones más elementales por ejemplo la de regar una parcela, la de

cosechar o la de sembrar, entre otras.
>”granjerosLib.h”: el header file del juego contiene todas las declaraciones de funciones del juego, además de constantes y structs.
>”granjerosIni”:contiene todas las funciones que inicializan las variables, además de contener la función de decremento de matriz de tiempo de recuperación, y la matriz de tiempo de cosechado de una parcela.
>”granjerosVal”:contiene las funciones que validan las entradas del usuario.
>”granjerosUser”:contiene las funciones que muestran al usuario el curso del juego además de mostrar el menú principal de opciones.

5. Funciones

Se utilizan funciones como atoi() predefinida en la biblioteca <cstdlib> que sirve para convertir lo ingresado por el usuario en un entero. Esto porque el usuario podría ingresar una tira de caracteres no válidos, entonces se valida que no lo sea y luego se usa esta función para obtener en un int la opción esperada.
Se utiliza la función void exit(int) que ocasiona la terminación normal del programa al llamarla una vez.
#Funcion ofstream de lectura de archivo.
#Funcion rand y srand para generar números pseudoaleatorios ya que la función rand tomara una tira de resultados y cada vez que ejecute el programa saldrá un numero pero siempre de la misma tira, la función srand toma una tira distinta cada vez pero esto corrompe la aleatoriedad. No encontré nada mucho mejor.
#Procedimientos antes que funciones, no se me ocurrió

otra manera en el tiempo dado. Esto es un punto de mejora y optimización del código, arrastrando menos variables puntero por los procedimientos.

6.Ampliación de la aplicación

Según las reglas el juego cumple lo pedido, aunque hay puntos a mejorar por ejemplo cada vez que se pide regar en la función regar() se podría preguntar si se quiere seguir regando esto implica que la función no retorne al menú de opciones directamente.
Se podría modificar la cantidad de turnos inicial del jugador ya que con 10 turnos queda limitado el tiempo de recuperación de una parcela para plantar otro cultivo por ejemplo el cultivo B además de tener un tiempo de cosecha de 4 turnos. Todo esto implica modificar la parte de la función que inicializa al jugador inicializarJugador() ubicada en granjerosIni.cpp.

MANUAL DE USUARIO

COMO JUGAR

El juego comienza mostrando la bienvenida al mismo, y pide al inicio de cada turno arrojar un dado. Se debe ingresar la letra S equivalente a un SI de lo contrario no se tirara el dado y no se obtendrán unidades de agua para regar el terreno.
Se muestra un menú de opciones para ir moviéndose entre ellas.

Tienda de semillas

El usuario ingresa a la tienda de semillas mediante la opción 1.
En la tienda se pregunta qué cultivo se desea comprar y que cantidad mediante una leyenda en pantalla.
De haber ingresado un cultivo no existente se mostrara la leyenda cultivo inexistente, luego seguirá preguntando si se quiere comprar algo más hasta que el usuario ingrese N(siempre en mayusculas) para volver al menú principal.

Sembrar

Se ingresa mediante la opción 2.
Se pide ingresar el tipo de cultivo a sembrar esto equivale a ingresar por teclado A, B o C.
Si no se ingresó algún cultivo existente se muestra mensaje de ingreso erróneo y se pide si se quiere seguir

sembrando.
Si se sigue sembrando se vuelve a pedir el ingreso del cultivo.
Una vez que se sembró en una parcela valida del terreno.
Se pregunta si se quiere seguir sembrando si es N se vuelve al menú principal.

Regar

Se accede mediante la opción 3.
Si no se tiene agua en tanque de agua se sale de la función al menú principal.
Una vez que se ingresó una opción válida del terreno se riega la parcela y se vuelve al menú principal, si se ingresó una opción no valida se vuelve al menú principal.
Si no se regó el terreno en algún turno, se seca el cultivo perdiéndose en el turno siguiente .

Cosechar

Se accede mediante la opción 4.
Pide posición donde se desea cosechar si la posición esta en recuperación o no es cosechable se pregunta si se desea seguir cosechando, en este momento se puede acceder al menú mediante la opción N.
Si se cosecha una posición se ganan monedas de acuerdo al tipo de cultivo.
El cultivo A o el C da cierta cantidad de monedas, el cultivo B es el que más da pero deja más cantidad de tiempo de recuperación de parcelas. Tenga en cuenta que al no cosechar la parcela en el momento justo al siguiente turno se pudre dejando el tiempo de recuperación de la parcela a la mitad.
Pasar al siguiente turno
En esta opción que es la 5 el jugador pasa al siguiente

turno.

Salir

Es la opción 6 y quita el juego completamente.

OBJETIVO DEL JUEGO

Incrementar la cantidad de monedas iniciales otorgadas por el juego en una cantidad de 10 turnos.

Cuestiones de implementación

Se pidió que si se seca un cultivo al siguiente turno se pierda el mismo por lo que la decisión que tome fue de que cuando termina el turno la función que valida si se regó tal parcela dejara el terreno en # si en tal caso no fue regada perdiendo así el cultivo antes de su cosecha.
Según el tiempo de recuperación pedido en las consignas aparecerá al finalizar el turno la letra R (recuperación) o no, hasta que el mismo (el tiempo) finalice.
También se pedía que que al no cultivar una parcela que esta apta para el cultivo en el tiempo determinado se termina pudriendo, el cultivo , perdiéndose en el siguiente turno.
En este caso lo simbolice con la letra P.

En el caso de que se pudra en el turno actual lo que se modifica es el tiempo de recuperación de la parcela de tierra a la mitad. La decisión que tome fue si es el cultivo C como tiene recuperación 0 no se modifica este parámetro , si el cultivo es A la mitad de 1 es ½ pero no se puede dejar en ½ turno de recuperación. Y finalmente si el cultivo es B la recuperación se reduce de 2 a 1.
Otra decisión fue la de mostrar el terreno al final de cada turno solamente para que el juego tuviera mas “misterio” y requiera de mas memoria de parte del jugador a la hora de acordarse donde planto que, en que momento regar y

demás. Aclaro esto porque al principio había decidido mostrarlo al terreno ante cada acción de regar cosechar o sembrar un cultivo.

Respuestas al cuestionario

Que es un Debug(depurar)?

La depuración de programas es el proceso de identificar y corregir errores de programación. En inglés se conoce como *debugging*, porque se asemeja a la eliminación de *bichos* (*bugs*), manera en que se conoce informalmente a los errores de programación.

Que es un Breakpoint?

Un BreakPoint es un punto de corte, que en programación es una linea especifica en la cual queremos que se detenga el flujo habitual del programa. Sirven para poder examinar el valor de las variables, o ejecutar el programa paso a paso.

¿Que es "Step Into", "Step Over" y "Step Out"?

Son comandos parte de la ejecución paso a paso.
Step Into *:* ejecuta la siguiente línea de código. Si esa línea es una llamada a otra función, el programa entrará en esa función.
Step Over : igual que la anterior, pero si la siguiente línea es una función, la ejecuta sin entrar en ella.
Step Out *:* sale de la función actual.