

75.41 - Algoritmos y Programación II

Cátedra Ing. Patricia Calvo - 1er cuatrimestre 2018

Trabajo Práctico 2: Granjeros V1.0

Enunciado

Desarrollar una pieza de software que implemente el juego de consola Granjeros con las siguientes especificaciones.

Definición del juego

Granjeros es un juego de estrategia en el que los participantes administran parcelas para distintas variedades de cultivos en sus terrenos.

Objetivo del juego

Maximizar las ganancias obtenidas a partir de la explotación de los recursos disponibles.

Conceptos

Terreno

Conjunto de parcelas con las que cuenta el juego para cultivar.

Parcela

Cada una de las regiones que componen un terreno.

Es la unidad mínima de explotación con un cultivo.

En un momento dado, una parcela puede estar limpia o tener un único cultivo.

Cada parcela se identifica por un par de números: (fila, columna). Por ejemplo, la primer parcela es la (1,1), ubicada en la fila 1 y columna 1.

Cultivo

Recurso desarrollado durante el curso del juego. Existen múltiples tipos de cultivos, con diferentes características.

Tienen origen en las semillas compradas, sembradas, regadas y cosechadas en las parcelas del terreno.

Créditos

Los créditos son las monedas disponibles que tiene un jugador para comprar cosas, y son ganadas al cosechar una parcela de cultivo.

Turnos

Unidad de tiempo en las que se divide una partida del juego.

Condiciones iniciales

Al iniciar el juego los recursos disponibles son los siguientes

- Terreno: 1 terreno de NxM Parcelas por jugador
- Crédito: 2xNxM Monedas (a decidir por el grupo en el nivel de dificultad)
- Tanque de Agua: vacío y capaz de almacenar hasta NxM Unidades de Riego (a decidir por el grupo en el nivel de dificultad)
- Tiempo: L Turnos
- Cultivos: indicados en el archivo cultivos.txt

Cultivos

Los cultivos disponibles vienen dados por un archivo de texto, a saber:

NOMBRE, COSTO, TIEMPO_DE_CRECIMIENTO, RENTABILIDAD,
TIEMPO_DE_RECUPERACION, CONSUMO_DE_AGUA

Ej:

A,1,2,10,1,3

B,3,4,30,2,1

C,2,1,5,0,2

Estos valores en una tabla serían:

Cultivo	Costo semilla <i>[Monedas por Parcela]</i>	Tiempo hasta cosecha <i>[Turnos]</i>	Rentabilidad <i>[Monedas por Parcela]</i>	Tiempo de recuperación <i>[Turnos]</i>	Agua x Truno <i>[U de Riego]</i>
A	1	2	10	1	3
B	3	4	30	2	1
C	2	1	5	0	2

Referencias

Cultivo: Nombre del cultivo

Costo semilla: Cantidad de monedas necesarias para comprar semillas suficientes para sembrar una parcela del cultivo.

Tiempo hasta cosecha: Cantidad de turnos necesarios desde que se siembra las semillas hasta que se puede cosechar una parcela del cultivo.

Rentabilidad: Cantidad de monedas ganadas por cosechar una parcela del cultivo.

Tiempo de recuperación: Cantidad de turnos que se necesitan para volver a utilizar una parcela después de haber cosechado el cultivo.

Turnos

El juego se desarrolla durante una secuencia de turnos.

En cada turno un jugador puede realizar tantas acciones como crea convenientes siguiendo las reglas y ateniéndose a las restricciones establecidas.

Al inicio de cada turno el jugador tira un dado de 6 caras y obtiene el potenciador de agua.

Agua

En cada turno, el jugador recibe agua que almacena en su tanque y que podrá utilizar para regar las parcelas.

La cantidad de unidades de riego recibidas se calcula como 5 por el potenciador de agua.

El tanque de agua tiene una capacidad máxima de unidades de riego. Las unidades de riego en el tanque que no fueron usadas al finalizar el turno quedan disponibles para el próximo. El jugador puede comprar una ampliación del tanque.

Almacén

Las cosechas se guardan en un almacén para ser enviadas. Cada envío tiene un costo por destino. Los destinos se cargan desde un archivo de texto: destinos.txt, a saber:

NOMBRE_DEL_DESTINO,DISTANCIA_EN_KM,PRECIO,CULTIVO

EJEMPLO:

SAN LUIS, 604, A

CORDOBA, 720, B

El almacén tiene una capacidad de $2 \times (N+M)$ (a decidir por el grupo en el nivel de dificultad) cosechas. Se puede extender mediante la compra. Si no hay capacidad al final del turno, la cosecha se pierde.

Acciones

El jugador puede realizar las siguientes acciones durante un turno

- Sembrar semillas en una parcela:
Comprar semillas de un cultivo con las monedas disponibles y utilizarlas para sembrar una parcela libre.
- Cosechar una parcela:
Liberar una parcela tomando el cultivo luego de haber transcurrido la cantidad de turnos necesarios para el cultivo. Esta cosecha va al almacén.
- Regar una parcela:
Utilizar una unidad de riego consumida del tanque de agua para que el cultivo no se seque.
- Enviar: enviar una cosecha a destino y cobrar.
- Comprar un Terreno: comprar un terreno. El valor lo decide el grupo. Debe costar más caro a medida que el jugador tiene mas terrenos.
- Vender un Terreno: se puede vender un terreno. Se cobra el 50% del costo actual.
- Comprar capacidad del tanque de agua.
- Comprar capacidad del almacén.

Reglas

- Si una parcela no es cosechada a tiempo se pudre, perdiéndose en el siguiente turno, pero la parcela reduce su tiempo de recuperación a la mitad.

- Toda parcela cultivada debe ser regada en cada turno, de lo contrario se seca, perdiéndose en el siguiente turno

A definir por el Usuario:

- 1) Cantidad de jugadores
- 2) Parámetro N y M
- 3) L: cantidad de turnos
- 4) Contenido de los archivos de Cultivos y Destinos

A definir por el Grupo:

- 1) Definir los valores del juego con 3 dificultades. Ej: cantidad de turnos, tamaño inicial, valores de las compras, etc.

Interfaz de usuario

El usuario debe interactuar con la juego por medio de la consola, pero el juego debe guardar cada nuevo estado del tablero en un bitmap (distinto cada vez) donde represente el terreno y sus datos.

Normas de entrega

Trabajo práctico individual: 5 persona.

Reglas generales: respetar el apéndice A.

Se deberá subir un archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

Nombre_De_Grupo-TP2.zip

Deberá contener los archivos fuentes (no los binarios), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF).

La fecha de entrega vence el día jueves 24/05/18 a las 23.59hs.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

Apéndice A:

1) Usar las siguientes convenciones para nombrar identificadores.

Clases o Structs: Los nombres de clases o Structs siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.

Métodos y funciones: Deben comenzar con letra minúscula, y si esta compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().

Variables y objetos: las variables siguen la misma convención que los métodos. Por Ejemplo: alumno, padronElectoral.

Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "_". Ejemplo: ANCHO, VACIO, COLOR_BASE.

2) El lenguaje utilizado es C++, esto quiere decir que se debe utilizar siempre C++ y no C, por lo tanto una forma de darse cuenta de esto es no incluir nada que tenga .h, por ejemplo `#include <iostream>`.

3) No usar sentencias 'using namespace' en los .h, solo en los .cpp. Por ejemplo, para referenciar el tipo string en el .h se pone `std::string`.

4) No usar 'and' y 'or', utilizar los operadores '&&' y '||' respectivamente.

5) Compilar en forma ANSI, no todos utilizan el mismo compilador y de esta forma se minimizan los errores. Debe estar compilado con la versión de Virtual Box que damos en la cátedra. Utilizamos el estándar C++98.

6) Chequear memoria antes de entregar. No tener accesos fuera de rango ni memoria colgada.

7) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.

8) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.

9) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios en el .h que los declara.

10) Modularizar el código. No entregar 1 o 2 archivos, separar cada Clase o Struct con sus funcionalidades en un .h y .cpp

11) No inicializar valores dentro del struct o .h.

12) Si cualquier estructura de control tiene 1 linea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    std::cout << i;  
}
```