TRABAJO PRACTICO GRUPAL DE:

Algoritmos

y

Programacion

2

Cátedra: P. Calvo

Nombre del grupo: Punto y Coma (;)

Tema: Juego de la Vida 2.0

Integrantes:

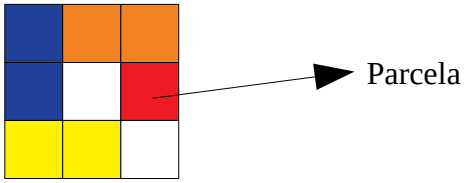
* Altamirano, Giuliana (100012)
* Cañizales, Cristina (102532)
* Couce, Santiago (100318)
* Perez Freire, Sol (98618)
* Valeriano, Juan Gabriel (97099)

Año: 2018

Informe TP2 “El juego de la vida V1.0”.

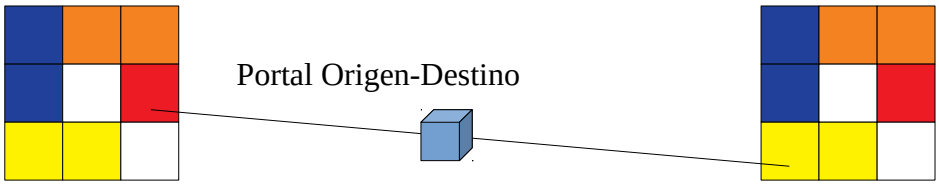
Acción n°1: Extraer los datos de importancia del enunciado.

Para la ejecución del programa se usaran varios tableros conectados por portales que contendrán células vivas o muertas.



Los tableros serán de un tamaño NxM que se indicara en el archivo de texto inicial.
Están compuestas por parcelas que tendrán asignadas un color y tasas de mortalidad y natalidad.

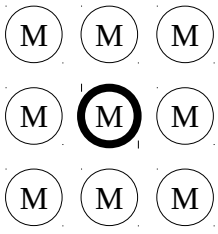
Las células pueden estar vivas o muertas.



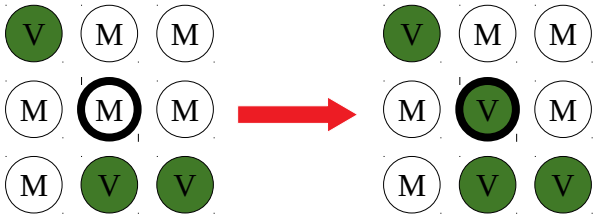
Los portales se encargan de conectar los diferentes tableros.
Hay tres tipos de portales y dependiendo de cual se trate podrá modificar las células que se vean afectadas por el mismo.

<u>Las posibles transiciones de la célula de un turno a otro:</u>

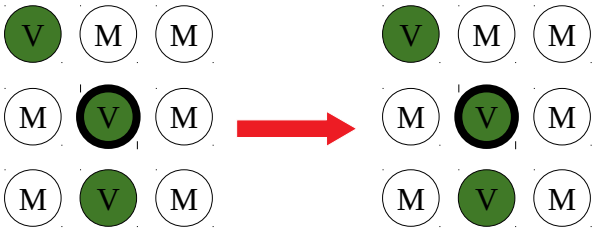
El estado de una célula cambiara por la interacción con sus células vecinas, las tasas de natalidad y mortalidad de la parcela donde se encuentra y los portales que la incluyan.



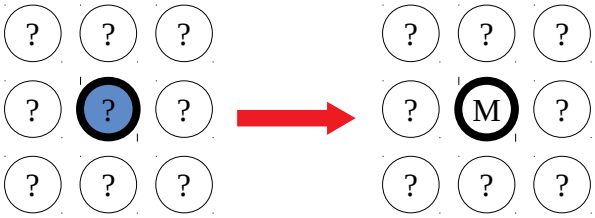
Una célula puede tener hasta 8 células vecinas en un tablero.



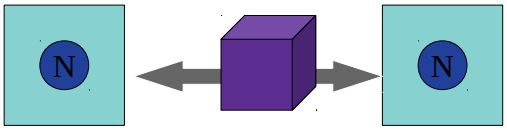
Una célula muerta(M) con exactamente 3 células vecinas vivas nacerá en el próximo turno.



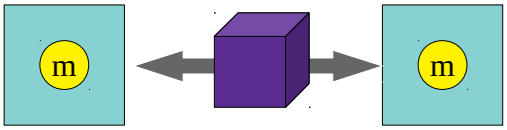
Una célula viva(V) con 2 o 3 células vecinas vivas no modificara su estado para el próximo turno.



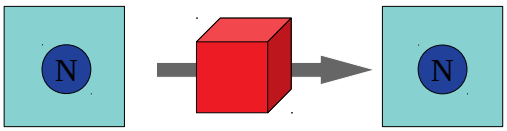
En cualquier(?) otro caso la célula muere o permanece muerta para el turno siguiente.



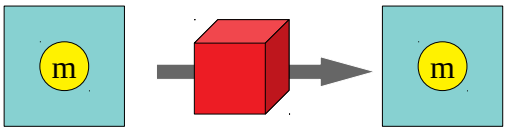
En un portal activo origen si una célula nace(N) entonces también nace en el portal destino y viceversa.



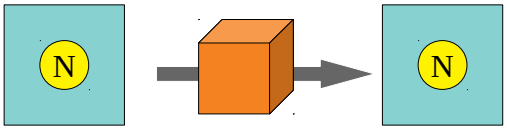
En un portal activo origen si una célula muere(m) entonces también muere en el portal destino y viceversa.



En un portal normal origen si la célula nace entonces **también** nace en el portal destino.



En un portal normal origen si la célula muere entonces **también** muere en el portal destino.



En un portal pasivo origen si la célula nace entonces **también** nace en el portal destino.

Acción n°2: Determinar los datos de entrada y salida.

Datos de entrada:

* La ruta de la ubicación del archivo de texto que tiene los registros de los elementos que componen el juego.

* Un archivo de texto que tendrá la información de los tableros, sus parcelas, los portales y las células que comenzaran como vivas.
Dentro del archivo se pueden tener registro intercalados excepto las células vivas y evitando hacer referencia a elementos que no existen. Por ejemplo querer configurar una parcela de un tablero cuyo registro viene mas adelante o no existe.

* Elección de una tarea a realizar. Mediante un menú que se mostrara el usuario ingresara un entero que corresponderá a una tarea especifica del programa.

Datos de salida por pantalla:

* Cantidad de células vivas:
* Cantidad de nacimientos en el ultimo turno:
* Cantidad de muertes en el ultimo turno: (entero >= 0).
* Promedio de nacimientos en el juego:
* Promedio de muertes en el juego: (float entre 0 y n, con n < 100).
* Si el juego no cambio en 2 turnos consecutivos: (String informando cuando suceda el acontecimiento).

Datos de salida por BMP:

* Cada turno se guardaran archivos BMP para cada estado de todos lo tableros que componen el juego. El tamaño del archivo se definirá en base a las dimensiones de cada tablero.

Acción n°3: Hipótesis consideradas.

* El archivo de texto no contendrá errores que puedan perjudicar el armando de los elementos que componen el juego.

* Las posiciones de un tablero se designan por (numero de columna, numero de fila), en ese orden al contrario de lo habitual (fila, columna).

* La ultima linea del archivo de texto sera la palabra “FINAL” que indica la finalización de los registros dentro del mismo.

* Una célula puede ser modificada por 2 factores: células vecinas y la interacción con un tipo de portal. La prioridad de cambio sera: parcela primero y luego portal.
Ej: “X” célula viva esta ubicada en el origen de un portal y según sus vecinas tiene que seguir viviendo para el turno siguiente pero la célula ubicada en el destino tendría que morir en el próximo turno.

* El usuario del programa no ingresara datos que sean de otro tipo a los pedidos.

* El efecto de los portales prevalece por sobre el efecto de las células vecinas.

* Si se tiene que aplicar un efecto de un portal donde la célula que modifica esta por morir y la célula modificada esta muerta o por morir entonces no se cambia nada en el destino.

* Si se tiene que aplicar un efecto de un portal donde la célula que modifica esta por nacer y la célula modificada esta por nacer entonces no se cambia nada en el destino. Si la célula modificada esta viva entonces se copia el color en la célula destino pero no se considera un nacimiento.

Acción n°5: Pseudocodigo.

* Mostrar mensaje de bienvenida al usuario.
* Pedir la ruta de un archivo de texto.
* Procesar el archivo.
    * Crear elementos del juego reservando memoria.
* Dibujar los tableros iniciales y guardarlos en formato BMP.
* Mostrar menú de juego.
    * Elegir una tarea.
        * Ejecutar una cantidad de turnos a seleccionar.
            * Evaluar tableros del juego.
            * Actualizar tableros del juego.
                * Modificar BMP de los tableros.
            * Contar y guardar los cambios para los informes.
            * Dibujar tableros de juego.
            * Volver al menú de juego.
        * Reiniciar juego.
            * Reiniciar contadores, acumuladores.
            * Eliminar tableros y portales.
            * Pedir la ruta de un archivo de texto
            * Procesar el archivo.
            * Dibujar tableros de juego.
            * Volver a la lista de los tableros.
        * Terminar juego.
            * Mostrar mensaje de cierre del programa.
            * Salir del juego.

Acción n°6: Clases utilizadas.

JuegoDeLaVida: Es una partida del juego que contendrá todos los elementos y métodos necesarios para la ejecución del programa.

Tablero: Estructura de un tamaño predefinido formada por parcelas que contiene una célula. Implementada usando una array bidimensional cuyos elementos son punteros a las distintas parcelas que forman el tablero.

Dibujante: Encargado de generar el lienzo que representara un tablero del juego.

Interfaz: Encargado de mostrar por pantalla los distintos mensajes y datos de la partida durante el uso del programa.

Enfermero: Encargado de evaluar la condición de las células en el tablero. También sera quien modificara el color y la condición de las células haciéndolas nacer o matándolas.

Color: Representa un color formado por la escala RGB.

Informe: Guardara los datos necesarios durante el transcurso del juego para que sean mostrados al usuario.

Parcela: Representa la porción mas pequeña de un tablero que contiene una célula .

Portal: Conectara dos tableros distintos y podrá modificar las células vinculadas al mismo dependiendo del tipo de portal que sea (Activo, Normal, Pasivo).

Célula: Elemento cuyo comportamiento podrá variar durante el juego tanto su color, energía y estado de viva a muerta o viceversa.

Energía: Cantidad de vida que tendrá una célula.

Manual de programador.

El programa comienza pidiendo memoria para la clase **JuegoDeLaVida** que sera la que vinculara el resto de entidades del programa.
La nueva instancia **JuegoDeLaVida** reserva memoria para las clases:
* **Informe** (lleva el registro de los cambios en los tableros).
* **Interfaz** (muestra por pantalla los mensajes de las diferentes situaciones de juego).
* **Dibujante** (dibuja cada tableros de juego creando un archivo .BMP).
* **Enfermero** (analiza los tableros y los portales para determinar el estado próximo de las células).
También se pide memoria para 2 listas:
* Lista de portales (guarda todos los portales del juego. Los mismos tienen acceso a las parcelas que vinculan).
* Lista de tableros (guarda todos los tableros del juego. El **Tablero** es un array bidimensional del tipo puntero a una parcela).
Luego se ejecuta el método jugar() que realiza una partida del juego que podrá ser reiniciada.
Al comienzo del juego la pantalla muestra un mensaje de bienvenida al usuario del programa y luego le pide la ruta de un archivo txt para cargar los datos de los elementos del juego.
Durante el procesamiento de archivo se toma la primera palabra de cada linea para saber que tipo de ingreso se trata. Puede ser “Tablero”, “Parcela”, “Portal” , “”CelulaViva”.
Los archivos para lectura tienen que seguir una estructura donde no se hagan referencias a elementos no procesados todavía (en caso de existir).
Si se lee la palabra “Tablero” el programa toma el resto de los datos y lo crea pidiendo memoria para todos sus elementos que lo componen. El contenido del tablero son punteros a parcelas que contienen una célula muerta cuyo color es blanco. Para finalizar se lo agrega a la lista de tableros.
Si se lee la palabra “Parcela” se toma el resto de los datos que la caracterizan y se la configura en el tablero donde esta ubicada.
Si se lee la palabra “Portal” el programa pide memoria para un portal que contendrá referencias a la **Parcela** origen y destino del **Portal** para tener acceso a las células.
Al final del archivo tienen que estar los registros de las células que comenzaran con vida.
Se toman sus datos y las configura en el tablero que las contiene. Para finalizar se lo agrega a la lista de Portales.
Después de procesar todo el archivo el Dibujante se encarga de generar los archivos BMP que representan a los tableros del juego en su estado inicial.
Como siguiente paso se ingresa en un bucle de tareas donde podremos efectuar las distintas acciones que se pueden realizar con los elementos del juego .De este bucle se saldrá cuando se quiera terminar el juego.
En el bucle la Interfaz primero muestra por pantalla el menú del juego y obtiene la elección por parte del usuario. Podrá elegir entre 3 acciones posibles:
* Ejecutar una cantidad de turnos:
Se muestra por pantalla y se pide la cantidad de turnos que se quieren ejecutar. Los turnos se ejecutaran siempre y cuando el estado de los tableros no hayan entrado en equilibrio (dos turnos seguidos sin cambios).
Por cada turno Nisman el Enfermero del juego primero determinara el estado próximo de cada **Célula** de todos los tableros en base a las células vecinas.

Lo siguiente es determinar el color que tendrán las células que nazcan el próximo turno. Al finalizar esto se controlara la influencia de los portales del juego en las células vinculadas. Se analiza cual de las células producirá cambios en la otra para modificarle el estado y color del próximo turno.
El enfermero actualiza el estado de las células haciéndolas nacer o matándolas mientras va tomando registro para generar los informes del juego.
El dibujante dibuja el nuevo estado de los tableros y genera los archivos BMP del turno correspondiente.
La Interfaz muestra por pantalla los informes del juego y si el juego se congelo en el intento de ejecutar los turnos.
Regresamos al menú de tareas.
* Reiniciar:
Libera la memoria pedida para los tableros y portales del juego. Pide por pantalla la ruta de un archivo txt y lo procesa generando nuevos elementos para jugar.
Se dibujan y generan los archivos correspondientes a los nuevos elementos del juego y se vuelve a mostrar el menú de juego.
* Terminar:
La Interfaz muestra un mensaje de finalización del programa y se sale del bucle de tareas para terminar el programa.

MANUAL DEL USUARIO

El juego de la vida es un autómata celular diseñado por el matemático británico John Horton Conway en 1970. Hizo su primera aparición pública en el número de octubre de 1970 de la revista Scientific American, en la columna de juegos matemáticos de Martin Gardner. Desde un punto de vista teórico, es interesante porque es equivalente a una máquina universal de Turing, es decir, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida.

Este es un juego de cero jugadores. Su evolución tiene lugar a partir de la entrada de datos iniciales y una vez inicia, no hay necesidad de carga de datos posteriores.

En nuestro caso, el campo de juego estará determinado por varias mallas formadas por cuadrados ("células"), cada una de dimensiones independientes. En cada posición del tablero (“parcela”) puede existir un portal, que como regla general inicia la vida en el destino del portal si se inicia la vida en el origen del portal.

Si en el origen del portal nace una célula viva, en el destino del portal se duplica dicha célula. Si en el origen del portal la célula muere, depende del tipo de portal, la célula destino permanece viva a muerte.

Tipos de Portales:

- Activos: si nace en origen, nace en destino. Si muere, muere en destino. Y viceversa.
- Normal: si nace en origen, nace en destino. Si muere en origen, muere en destino. No al revés.
- Pasivo: si nace en origen, nace en destino. Si muere en origen, no muere en destino.

Las células tienen dos estados posibles, pueden estar “vivas” o “muertas”. El estado de la malla va cambiando por turnos y el estado de todas las células se toma en cuenta para definir, y actualizar de manera simultánea, el estado de estas en el turno siguiente.

Estas actualizaciones se llevan a cabo dependiendo del estado de las células vecinas:

- Si una célula está viva: si tiene 2 o 3 células vecinas vivas, se mantiene viva, en otro caso muere por soledad o superpoblación. Se le resta 100% de la energía multiplicado por el factor de mortalidad de la Parcela.
- Si una célula está muertas: si tiene exactamente 3 vecinas vivas, la célula nace y en el próximo turno estará viva. Se le suma 100% de energía multiplicado por el factor de nacimiento de la Parcela. Cuando una célula nace copia el color promedio de las células vecinas, y sino la de la Parcela donde está.

Para jugar, el usuario solo debe indicar la ruta del archivo de texto en el que se encuentran los datos iniciales del juego. Posteriormente, el programa mostrará el menú y el usuario tendrá la opción de elegir entre ejecutar una cantidad X de turnos (a indicar por el usuario), reiniciar el juego o terminarlo, en ese orden.

- Si elige ejecutar la cantidad X de turnos (1): el juego continuará su curso natural hasta que indique otra acción a tomar o hasta que hayan transcurrido X turnos.
- Si elige reiniciar (2): el juego se reiniciará y regresará a la parte en la que le pide al usuario la ruta del archivo de texto.
- Si elige terminar (3): el juego habrá terminado y en caso de querer jugar, deberá volver a correr el programa.