Universidad de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

Universidad de Valparaíso CHILE

Matemáticas discretas
Proyecto 2 (Recursividad)

Prof: Dr. Fabián Riquelme Csori
Ayudante: Diego A. Monsalves Cabello
2020-II

1. Objetivos

- Aplicar conceptos teóricos de programación recursiva.
- Comprender el funcionamiento del juego de la vida (*Conway's game of life*, en inglés).
- Implementar este juego utilizando funcionalidades recursivas.
- Desarrollar competencias de trabajo grupal.

2. Descripción

- El proyecto se realiza en **grupos de 4 a 5 integrantes**, organizados primero por preferencia, y luego aleatoriamente, para aquellos estudiantes que no formen grupos directamente.
- Se debe entregar una **solución ejecutable**, sin errores, acompañada de **código bien documentado** y ajustado a los requerimientos.
- La información e interfaz gráfica entregada por los estudiantes debe ser clara, consistente y ordenada.
- Se debe además entregar un **informe técnico** sobre la solución realizada.
- Eventualmente, el profesor o ayudante podría pedir que un grupo le explique la entrega en reunión por videoconferencia, de forma de validar el trabajo realizado.

2.1. Juego de la vida

El juego de la vida (en inglés, *Game of Life*), es un autómata celular diseñado por el matemático británico John Horton Conway en 1970. Un autómata celular es un modelo matemático que permite representar un comportamiento evolutivo mediante pasos discretos. Con ellos es posible modelar sistemas dinámicos discretos, en los que una colección masiva de objetos simples interactúan localmente uno con otros.

El ambiente discreto del juego de la vida es una cuadrícula o matriz que va evolucionando a través del tiempo. En un tiempo t, cada celda o célula de la matriz posee uno de dos estados posibles (vivo/muerto, o para no ser tan dramáticos, activo/inactivo). En un tiempo $t+1$ (una "nueva generación"), el estado de cada celda podrá cambiar (o mantenerse) de acuerdo al estado

que poseían sus vecindades (es decir, sus celdas directamente adyacentes de forma horizontal, vertical o diagonal) en el tiempo t. Esta variación dependerá de ciertas reglas de transición que se definirán a continuación:

- Despoblación o exposición:
  Una célula activa con menos de dos vecinos activos se desactiva.
- Sobrepoblación o hacinamiento:
  Una célula activa con más de tres vecinos activos se desactiva.
- Una célula activa con dos o tres vecinos activos no sufre cambios en la próxima generación.
- Una célula inactiva con exactamente tres vecinos activos se activará (cobrará vida).

La evolución está determinada por el estado del autómata, y no necesita la participación de jugadores, los cuales solo se centran en observar cómo evoluciona el sistema.

La Figura 1 representa el funcionamiento del algoritmo, con una representación gráfica para cinco generaciones del autómata. Para simular el comportamiento del algoritmo de una manera más general y en línea, puede ingresar a `https://playgameoflife.com`.

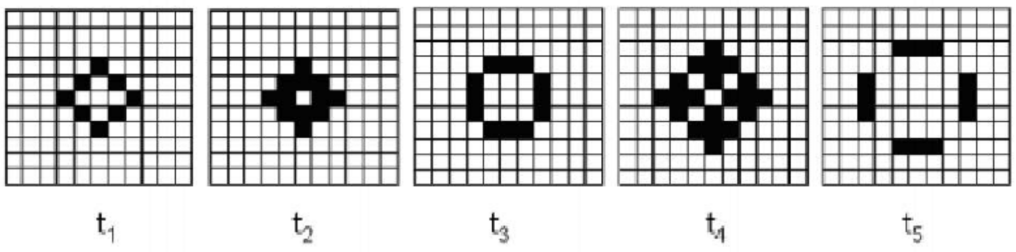

Figura 1: Ejemplo de funcionamiento del algoritmo durante 5 generaciones.

2.1.1. Trabajo a realizar

- Implementar el juego de la vida en el lenguaje de programación C++, cumpliendo con la totalidad de las reglas establecidas para el autómata. Cualquier copia de código será sancionada, incluido las implementaciones que se puedan encontrar en internet.
- La estructura de datos para el manejo de la cuadrícula y el paradigma de programación (estructurado, orientado a objetos) se deja a libre elección, siempre considerando la eficiencia de funcionamiento.
- Las reglas de transición del algoritmo debe ser diseñadas e implementadas obligatoriamente como funciones recursivas. No se aceptarán implementaciones iterativas.
- Cada generación del algoritmo debe presentarse mediante una interfaz gráfica, en donde, entre las generaciones t y $t + 1$ debe haber un retardo o *delay* adecuado que permita ver los cambios del autómata.

- Deben entregar en formato PDF un informe técnico de su solución, que describa el diseño e implementación desarrollada. El informe puede ser breve, pero sobre todo debe ser conciso y completo, detallando cada aspecto que sea importante para comprender su solución (ver rúbrica de evaluación).
- Todos los integrantes del grupo deben comprender y ser capaces de explicar en su totalidad el código y la teoría subyacente.
- El algoritmo debe permitir ingresar las siguientes variables:
  - Tamaño del tablero (-T): Tamaño de la matriz cuadrada $n \times n$ que representa el ambiente del sistema.
  - Probabilidad inicial (-P): Este parámetro indica la probabilidad inicial de que una célula esté activa al iniciar el tablero con la primera generación. Esta probabilidad es un valor aleatorio del intervalo $[0, 1]$, donde 1 significa que todas las células del tablero están activas y 0 lo opuesto.
  - Cantidad de generaciones (-G): Número de generaciones que se ejecuta el algoritmo.

2.2. Entrega

El proyecto debe ser subido en un único archivo `.zip` al aula virtual, en el recurso respectivo, con el siguiente formato:

Proyecto2-Grupo NUM_GRUPO.zip
Ejemplo: Proyecto2-Grupo5.zip

Dentro de este archivo deben incluir:

- Archivo(s) fuente(s), bien documentado(s) y, en caso de existir, todo recurso adicional utilizado por la aplicación (archivos o bibliotecas externas, etc.).
- Archivo PDF con el informe técnico grupal.

Además, cada estudiante debe responder a una encuesta que se le hará llegar, indicando el factor de participación (FP2) para cada compañero/a de grupo (ver Sección 3.3).

Fecha de Entrega: 9 de noviembre de 2020, 13:30 horas.

Frente a cualquier duda sobre el proyecto, pueden contactar también al ayudante a través de su correo electrónico institucional (`diego.monsalves@alumnos.uv.cl`).

3. Sistema de evaluación

Este proyecto vale un 30 % de la nota final del curso. La nota del Proyecto 2 (NP2) estará dada por una nota grupal (NG2) resultante del algoritmo implementado, el informe técnico (IF2) y un factor de participación (FP2) entregado por sus pares de forma anónima:

$$\text{NP2} = (\text{NG2} \cdot 0{,}7 + \text{IF2} \cdot 0{,}3) \cdot \text{FP2}$$

3.1. Nota grupal (NG2)

Criterio	Puntos
Calidad del código. El código está implementado en el lenguajes de programación requerido (no se aceptarán trabajos en otro lenguaje de programación). No utiliza bibliotecas externas que resuelvan el problema directamente. El código es claro y está bien documentado. Las entradas del algoritmo están condicionadas para valores erróneos.	10 pts
Completitud. Se implementan correctamente las cuatro reglas de transición de manera recursiva (30pts). El tamaño del tablero es modificable como parámetro de entrada. (10pts). La generación inicial se escoge aleatoriamente mediante una probabilidad inicial ingresada como parámetro (10 pts). El algoritmo ejecuta un número de generaciones ingresado como parámetro de entrada (10pts).	60 pts
Correctitud global. El algoritmo funciona globalmente de forma correcta, sin errores y en un tiempo acorde a lo solicitado. Los parámetros de entrada son validados y funcionan correctamente.	20 pts
Visualizaciones. Las interfaz gráfica muestra de buena manera cada generación del algoritmo.	10 pts

3.2. Informe técnico (IF2)

Criterio	Puntos
Calidad del documento. El documento entregado es conciso y directo, tiene un formato ordenado y sin faltas ortográficas. Se detallan claramente las secciones que encontraron pertinente documentar.	30 pts
Diseño de la solución. Presentan una sección de diseño de solución, donde se describe cómo pensaron resolver el problema de manera recursiva.	30 pts
Implementación. Presenta una sección de implementación de la solución, explicando técnicamente las secciones del código, de una manera ordenada. Se puede seguir un hilo del proceso del algoritmo. Ejemplifican con extractos de códigos de la solución.	30 pts
Recursos utilizados. Presentan una sección que detalla las herramientas que utilizaron, como bibliotecas, librerías, etc.	10 pts

3.3. Factor de participación (FP2)

Nivel de trabajo	Factor
Excelente. Mi compañero/a fue indispensable para realizar del trabajo. Sin él/ella no se podría haber realizado (solo puede evaluar a **máximo 1** integrante de esta manera).	1.1
Bueno. Mi compañero/a se desempeñó de manera satisfactoria y responsable en el equipo.	1.0
Insuficiente. Mi compañero/a no participó lo suficientemente en el trabajo. Su desempeño no fue el ideal esperado.	0.8
Deficiente. Mi compañero/a no fue un aporte al equipo. El equipo habría funcionado mejor sin él/ella.	0.6

La nota FP2 para un/a estudiante corresponde al promedio de los factores ingresados por sus compañeros/as.