Tecnología de Programación (TdP)

Segundo Cuatrimestre 2023

Proyecto 1:

Modelado e implementación de una versión sobre simplificada de un juego.
Caso de estudio: Candy Crush Saga.

Candy Crush Saga es un videojuego multidioma desarrollado por la compañía británica *King*, originalmente disponible como una aplicación de Facebook y adaptado para los sistemas operativos Android, iOS y Windows Phone, que se lanzó el 12 de abril de 2012 y cuya aplicación para plataformas móviles vio la luz el 14 de noviembre del mismo año.

En el *juego*, los *jugadores* completan *niveles* intercambiando *caramelos de colores* en un *tablero* de *juego* para hacer una combinación de tres o más del mismo color, eliminando esos *caramelos* del *tablero* y reemplazándolos por otros nuevos, lo que podría crear más coincidencias. Las combinaciones de cuatro o más *caramelos* crean *caramelos* únicos que actúan como *potenciadores* con capacidades más grandes para limpiar el *tablero*. Los *tableros* tienen varios objetivos que deben completarse dentro de un número fijo de *movimientos* o una cantidad de *tiempo* limitada.

Candy Crush Saga es un *juego* de match tres, en el que los *jugadores* pueden mover dos *caramelos* adyacentes de entre varios en el *tablero* de *juego*, para formar una *fila* o *columna* de al menos 3 *caramelos* del mismo color. En esta *partida*, los *caramelos* emparejados se retiran del *tablero*, y los *caramelos* encima de ellos caen en los espacios vacíos, con nuevos *caramelos* que aparecen en la parte superior del *tablero*. Esto puede crear un nuevo conjunto de *caramelos* emparejados, que se borran automáticamente de la misma manera. Además, la creación de combinaciones de cuatro o más *caramelos* creará un *caramelo especial* denominado *potenciador* que, cuando se combinan, puede limpiar una *fila*, *columna* u otra *sección* del *tablero*.

El *juego* finaliza una vez que todos los *niveles* fueran completados, o bien cuando el *jugador* pierde un total de 3 *vidas* a lo largo del *juego*. Una *vida* se pierde toda vez que, agotado el *tiempo* o la cantidad de *movimientos* disponibles, el *jugador* no cumple con los *objetivos* del *nivel* que se encuentra jugando. Al momento de perder una *vida* (y siempre que se tengan más), el *jugador* inicia nuevamente el *nivel* actual desde cero.

A partir de la descripción general realizada previamente, el proyecto consiste en el diseño e implementación de una versión sobre simplificada e inspirada en *Candy Crush,* en la que **se debe respetar todas y cada una de las siguientes condiciones**:

- el tema del *juego* es de libre elección (en lugar de *caramelos*, se puede utilizar otro dominio de aplicación, pero deberán respetar las reglas generales del juego descripto independientemente del dominio).
- el *tablero* debe tener **al menos un tamaño** correspondiente a una *grilla* de 6x6 (los *niveles* pueden definir el tamaño de cada *tablero*).
- deben existir **al menos 5 niveles** por los que el *jugador* avanza, teniendo en cuenta que en cada *nivel* se va incrementando la dificultad del juego. La dificultad quedará dada por **la cantidad de movimientos disponibles,** el **tiempo disponible**, y los **objetivos específicos planteados** para el *nivel*.
- cada *nivel* se definirá **a partir de uno o varios archivos de texto plano**, en el que se deberán establecer las características del *tablero* al momento de comenzar a jugar: cantidad y ubicación de *caramelos*, *tiempo* y *movimientos* disponibles, así como el *objetivo* a cumplir.
- existirán **al menos 5** tipos de *caramelos* diferentes, con una gráfica distintiva (ver *Tabla 1*).
- existirán **al menos 2** tipos de *caramelos potenciadores*, con una gráfica distintiva (ver *Tabla 2*).

- existirán **al menos 4** tipos de reglas que podrán usarse como parte de los objetivos (ver *Tabla 3*). Estas características se combinarán, luego, con la indicación de *movimientos disponibles* y de *tiempo disponible*.
- El *juego* se operará **únicamente** mediante las letras *AWSD* y las fechas *arriba*, *abajo*, *izquierda* y *derecha* del teclado (**no se implementará** el *juego* según un esquema de *drag and drop* entre los *caramelos*). Las fechas permitirán avanzar el *cursor* que se posiciona sobre las diferentes *celdas*, resaltando estas de forma de entender dónde el *jugador* se encuentra posicionado, y las letras *AWSD* accionarán, siempre que sea posible, el intercambio entre dos *caramelos* adyacentes.
- el *juego* debe mostrar **en todo momento** la cantidad de *movimientos disponibles* y el *tiempo disponible* por el *jugador*, así como se debe indicar la distancia que existe respecto al cumplimiento del objetivo del nivel (por ejemplo, si se deben juntar una X cantidad de caramelos de un color, cuántos caramelos aún restan recolectar).
- el *juego* **no llevará** cuenta del *puntaje*, **ni requiere** de *sonidos* y *animaciones* avanzadas.
- toda *regla* que no se encuentre descripta en este enunciado, debe seguir la *política original del juego*.

Tabla 1: Tipos de caramelos y colores asociados

Color de caramelo	***Forma asociada***
Rojo	Frijol o chorizo
Naranja	Pastilla
Amarillo	Lágrima
Verde	Almohada
Azul	Planeta o piruleta
Púrpura	Flor

Tabla 2: Tipos de caramelos potenciadores

Caramelo	***Color***	
Rayado	Todos	**Apariencia**: Rayas horizontales o verticales en un caramelo regular. **Efecto**: Borra una fila o columna entera, dependiendo de la dirección de las rayas. **Formación**: Exactamente 4 caramelos regulares seguidos. **Ubicación de aparición**: en la posición hacia donde se movió el caramelo que generó la formación del mismo. Si el caramelo rayado es el resultado de un movimiento horizontal, las rayas serán horizontales, si es el resultado de un movimiento vertical, serán verticales y el resultado es aleatorio durante las cascadas más grandes.
Envuelto	Todos	**Apariencia**: Un caramelo normal en un envoltorio rectangular del mismo color. **Efecto**: Elimina los 8 caramelos circundantes. **Formación**: 5 o 6 caramelos regulares en forma de T, L o +. **Ubicación de aparición**: en la posición hacia donde se movió el caramelo que generó la formación del mismo.

Tabla 3: Tipos de reglas a considerar para formar los objetivos, junto con descripción de caramelos especiales

Regla	***Descripción de la regla***
1 – básica	Juntar una cantidad determinada de caramelos de un color.
2 – moderada	Eliminar una cantidad determinada de caramelos glaseados.
3 – avanzada	Eliminar una cantidad determinada de caramelos envueltos o gelatina.

Los caramelos gelatina son como los descriptos en la *Tabla 1*, recubiertos de una gelatina transparente; los glaseados son un tipo de caramelo especial, que solo aparecen si es que forman parte de los objetivos de un nivel dado.

Tecnología de Programación (TdP)

Segundo Cuatrimestre 2023

A partir de la descripción de arriba, resuelva:

Ejercicio 1: Consulte el video *Integrando lógica y gráfica: grillas y labels con Java Swing* desarrollado por la cátedra, disponible en el siguiente enlace: https://youtu.be/EeAKbwGPxaY

Ejercicio 2: Acceda al formulario de registro de usuarios GitHub, y notifique los nombres de usuario de todas las cuentas GitHub personales con las que, cada integrante de la comisión de cursado, podrá modificar el repositorio que la cátedra le creará para operar el proyecto.

Ejercicio 3: Mediante alguna herramienta de modelado, construya un diagrama de clases en UML en donde se puedan observar las clases que modelan el sistema a implementar, así como sus relaciones, atributos y servicios. Construya también un diagrama de secuencia que muestre cómo opera el sistema a partir de la interacción *"El usuario la presiona tecla D"* en el contexto de intentar intercambiar dos caramelos adyacentes. Exporte los diagramas en formato imagen o pdf y suba el/los archivos dentro de una carpeta denominada *Diagramas-UML* al repositorio *p1-comision-xx* (donde *xx* corresponderá con su número de comisión) que creará la cátedra y lo asignará como colaborador. Se esperan obligatoriamente **al menos siete archivos**, uno para el *diagrama de secuencia*, tres donde se encuentran los *diagramas reducidos de clases,* y tres para los *diagramas completos de clases*. Por cada acto de defensa asociado a este ejercicio (ver más adelante), deberán incorporar un *diagrama reducido* y *extendido* de clases y, finalmente, un *diagrama reducido* y *extendido* de clases final, con la entrega final del proyecto: de aquí, los siete archivos en total.

Ejercicio 4: A partir del modelo definido en el *Ejercicio 3*, e incorporando y considerando las observaciones indicadas por el ayudante designado para la corrección de su comisión, implemente completamente una versión del juego en Java haciendo uso de Java Swing. La versión deberá cumplir con cada uno de los siguientes ítems:

a) Implementar completamente la lógica del juego.
b) Desarrollar una interfaz adecuada para el desarrollo del juego.
c) Los objetos para completar cada celda de la GUI deberán mostrarse basándose en algún tipo de imagen.

Ejercicio 5: Suba al repositorio *p1-comision-xx* (donde *xx* corresponderá con su número de comisión) que creará la cátedra y lo asignará como colaborador, el proyecto Java mediante el cual implementó todo lo solicitado en el *Ejercicio 4*. Los archivos subidos al repositorio deben permitir el correcto funcionamiento y visualización del proyecto completo. Asegúrese de no olvidar ningún archivo. El proyecto debe compilar, caso contrario se considerará *Desaprobado*. Es **condición de aprobación** que todos los integrantes de la comisión hayan trabajado sobre la concreción del proyecto. Para esto, la cátedra se reserva el derecho de analizar los *insights* de cada repositorio, a fin de establecer el nivel de participación de cada usuario.

Condiciones de entrega:

- El proyecto deberá resolverse en grupos, según las comisiones de cursado indicadas en el aula virtual de la materia en Moodle. La copia o plagio del proyecto es una falta grave. Quien incurra en estos actos de deshonestidad académica, desaprobará automáticamente el proyecto.

- El proyecto será evaluado en **cinco etapas**. Tres de estas etapas **serán presenciales** y los alumnos deberán presentarse a clase para realizar una **defensa** del mismo.

- **Entrega y evaluación** del proyecto:
  - Etapa 1: considera la concreción de los ejercicios 1 y 2.
  - Etapa 2: considera la concreción parcial del ejercicio 3, respecto del diagrama reducido de clases.
  - Etapa 3: considera la concreción parcial del ejercicio 3, respecto de los diagramas completo de clases y de secuencia.
  - Etapa 4: considera la concreción parcial del ejercicio 4, aguardando un avance del 50% del proyecto.
  - Etapa 5: considera la concreción del ejercicio 4.

- Las **fechas límite de entrega y evaluación** de cada etapa, seguirán el siguiente formato:
  - Etapa 1: la fecha límite operará el **viernes 08/09,** a las **12:00 hs**.

  - Etapa 2: la fecha de puesta en común sobre el **diagrama de clases reducido** se realizará la **semana del 12 y 14 de septiembre**, entre las **10:00 y 12:00 hs**. Cada comisión tendrá un turno asignado en el cual todos los integrantes deberán asistir a clases con el ayudante asignado para la corrección del proyecto, y donde se debatirá sobre el diagrama confeccionado, observando los detalles a corregir o mejorar. La asistencia a este espacio es obligatoria, y la evaluación realizada impactará en la nota individual de cada uno de los alumnos que integre la comisión. Los turnos se publicarán en el aula virtual de la materia en Moodle.

  - Etapa 3: la fecha de puesta en común sobre el **diagrama de clases completo** se realizará la **semana del 26 y 28 de septiembre**, entre las **10:00 y 12:00 hs**. Cada comisión tendrá un turno asignado en el cual todos los integrantes deberán asistir a clases con el ayudante asignado para la corrección del proyecto, y donde se debatirá sobre el diagrama confeccionado, observando los detalles a corregir o mejorar. La asistencia a este espacio es obligatoria, y la evaluación realizada impactará en la nota individual de cada uno de los alumnos que integre la comisión. Los turnos se publicarán en el aula virtual de la materia en Moodle.

  - Etapa 4: la fecha de puesta en común sobre la **implementación de como mínimo un 50% del proyecto**, se realizará la **semana del 03 y 05 de octubre**, entre las **10:00 y 12:00 hs**. Cada comisión tendrá un turno asignado en el cual todos los integrantes deberán asistir a clases con el ayudante asignado para la corrección del proyecto, y donde se debatirá sobre el diagrama confeccionado, observando los detalles a corregir o mejorar. La asistencia a este espacio es obligatoria, y la evaluación realizada impactará en la nota individual de cada uno de los alumnos que integre la comisión. Los turnos se publicarán en el aula virtual de la materia en Moodle.

  - Etapa 5: la fecha límite para subir la última versión considerable para corrección al repositorio operará el **viernes 20/10,** a las **23:59 hs**. Todo *commit* realizado con posterioridad a esta fecha y hora no será considerado para la evaluación del proyecto.

- Los proyectos serán calificados según la siguiente escala: A, B y C corresponden a proyectos *Aprobados*; D corresponde a proyectos *Desaprobados*.

- Todo proyecto que no sea completado, entregado y defendido mediante las condiciones de entrega enunciadas, se considerará *Desaprobado*.