Trabajo Práctico Final:

Programación Orientada a Objetos

Integrantes:

Baliarda Gonzalo 61490,

Pérez Ezequiel Agustín 61475

Introducción

Este informe detalla las modificaciones, implementaciones y dificultades encontradas al momento de realizar los distintos modos de juego y su correcta funcionalidad para una versión académica y reducida del juego “Candy Crush”

Funcionalidades agregadas

1. Opción en la barra de menú que despliega los distintos modos de juegos disponibles para su uso.
2. Efecto visual al seleccionar el caramelo con el que se va a ejecutar el movimiento deseado, exceptuando en el modo “GoldenBoard”.
3. Alerta al ganar o perder en cualquiera de los modos de juego habilitados.
4. Para el modo de juego “Clásico”, la cantidad de movimientos restantes y el puntaje necesario para “ganar”.
5. El modo de juego “GoldenBoard”, en el que hay que “pintar” el tablero de color dorado mediante movimientos, donde muestra la cantidad de movimientos restantes para utilizar y las celdas que aún faltan pintar.
6. El modo de juego “TimeBomb”, en el que hay que eliminar los caramelos con contador antes de que lleguen a 0, donde cada movimiento que no elimina a estos caramelos le resta 1 a su contador, la implementación incluye la cantidad de caramelos que faltan eliminar, junto al puntaje.
7. El modo de juego “TimeLimit”, en el que hay que llegar a un puntaje determinado antes de que el contador de tiempo llegue a 0. En este nivel se cuenta con caramelo con un texto por encima que indica la cantidad de segundos que le agrega al contador.

Dificultades durante el desarrollo

Durante el desarrollo del Trabajo Práctico nos encontramos con cierta complicación al comprender el código provisto por la Cátedra, sumado a donde y como realizar las implementaciones en el código dado.

Cambios en la implementación provista

* Implementación de la barra de menú en el inicio, previa a cargar los juegos para no tener que cargar esta barra cada vez de cambiar el juego.
* Adición de un efecto visual para mayor claridad a la hora de saber qué movimiento se está por realizar.
* Modularización en donde dos o más secciones del código tenían porciones de código similares o iguales, principalmente en el desarrollo de los niveles.
* Creación de nuevas clases y métodos abstractos para lograr una modularización más eficaz.