Guido Barbieri 59567

Informe TPE POO:

Agregado de funcionalidades:

- Se agregaron dos nuevos niveles(Level2,Level3). Los mismos extendien a la clase Level(también agregada) la cual a sí mismo extiende Grid
- Se agrego la clase DestroyableCell. La misma se hizo para no repetir código entre Jail y Jelly(dos clases que también se agregaron)
- La clase Jail y la clase Jelly son las dos funcionalidades elegidas para implementar y se usan en Level2 y Level3 respectivamente.

Modificacion de codigo:

Por otro lado también se modificó el código provisto:

- Se agregó en el campo images de ImageManager una entrada para Jail y Jelly y se modificó getImage para que funcione recibiendo el key en vez de recibiendo un element. Esto es para poder obtener la imagen de las dos clases desde el ImageManager modificando la menor cantidad de código posible
- Se modificó BoarPanel. El mismo ahora usa un array de Group(ImageView,ImageView) esto es para poder manejar imágenes en dos capas con mayor facilidad, lo cual facilita la superposición de la imagen de una celda(Jelly o Jail) sobre la del caramelo. Asi mismo tambien se modifico el GameListener que agrega CandyFrame a game para funcionar de esta nueva forma
- Se le agregó a Cell el método público getKey, el cual retorna un string con el key de la celda(en Cell dado que no se necesita devuelve null), con el objetivo de que sea sobrescrito por Jail y Jelly. Esto es para facilitar buscar su imagen en ImageManager
- Se agrego entradas al menú para poder seleccionar los distintos niveles
- Se modificó Grid.Initialize para que use un método nuevo de la clase Grid para decidir que Cell corresponde en cada punto. El mismo es protected, lo cual nos permite sobrescribir en Level para que haga celdas distintas. Esto es útil para crear Jail y Jelly en Level2 y Level3
- También Se agregó un método protected que se ejecuta al terminar Grid.initialize. El objetivo del mismo es que se pueda hacer modificaciones al grid después del proceso de initialize pero antes de que termine ese método. En Grid este método no hace nada.
- Además se agregó un chequeo al principio de Grid.tryMove que se asegura previo a probar el movimiento que ambas celdas sean movibles. si esto no es así retorna falso. Esto es dado que si una celda no es movible(Jail o Nothing) entonces el movimiento no puede hacerse
- Se eliminó código de la clase Level1 ya que con la creación de Level mucho código de Level1 pasó a estar en Level
- Se hizo pública la variable CELL_SIZE de candyframe. Esto es para poder hacer un menu principal parecido en tamaño a lo que será el juego luego
- Se cambió el código en la clase GameApp para hacer que inicie con un menú principal en el cual se pueda elegir el nivel deseado
- Se agregó a Grid.tryMove wasUpdated si el movimiento es válido. Esto es porque sino se puede dar el caso en el que los elementos no deben caer(dado que el elemento que está arriba es una celda que no se puede mover) entonces no updatea la representación de la grilla pero la misma cambio

Guido Barbieri 59567

Problemas encontrados durante el desarrollo:

El mayor inconveniente para mí fue entender exactamente qué hacía cada cosa. Esto tomó la mayoria del tiempo. Una vez que entendí como funcionaba, armar sobre eso fue fácil. Esta es la mayor ventaja de este paradigma, es fácil construir sobre lo que ya está hecho. No tuve que hacer muchos y todos los cambios que hice fueron fáciles de hacer sin que se rompieran otras cosas.