

2º Sprint Grupo Fracaso

En la duración de este segundo sprint hemos tenido 2 problemas principales a repartir y uno menos laborioso y más sencillo que era la implementación de patrones. Los 2 problemas principales fueron rehacer el TamaDigOut desde 0 ya que el diseño primero era pobre y no seguía los criterios del MVC especificados, y el otro el diseño/implementación de la expansión.

TamaDigOut (Recomposición)

En cuanto a la recomposición del TamaDigOut no partíamos de 0 en ciertas funciones que ya teníamos hechas y que implementó Abdessamad en el primer sprint, pero aun así había que plantear otra forma de hacer las cosas muy distinta. Este trabajo fue realizado en por Abdessamad en un principio pero luego Iker lo fue avanzando poco a poco con ayudas de Ivan hasta que le dieron el toque final y se ajustaba perfectamente a las reglas del MVC.

Expansión

El minijuego en un primer momento fue diseñado por Jon y Nicolás, el primer diseño no cumplía con los criterios del MVC y tras eso Jon lo hizo aplicando las reglas especificadas. Tras ello la última parte de la creación de este minijuego Iker ayudó a terminar de solventar los errores que se le presentaron a Jon.

Patrones

En cuanto a la aplicación de los patrones esta parte correspondió a Iker y a Abdessamad, Iker implementó los patrones State y Composite en Tamagotchi.java y Abdessamad proporcionó el diseño inicial del Factory el cual no tuvimos problema para reutilizar en la recomposición del minijuego.

Controladores

Expansión

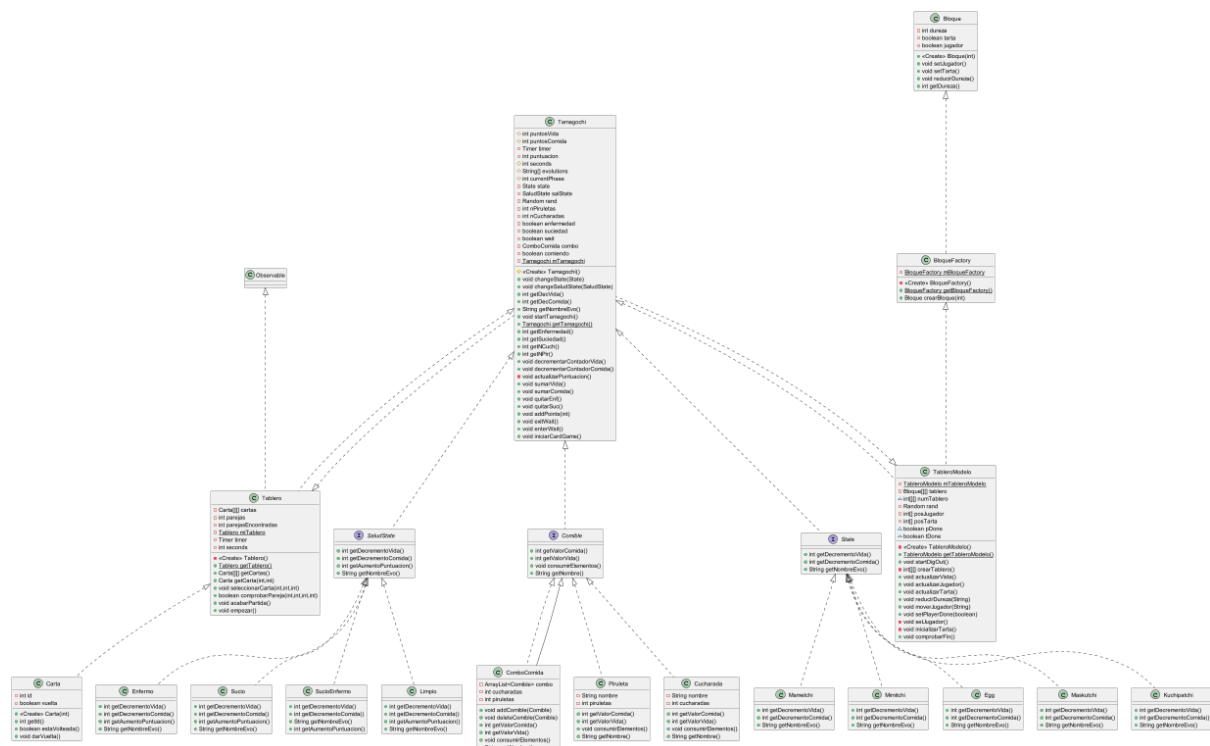
En cuanto al controlador de la expansión, este fue implementado por Jon que se inspiró en los usados en el juego principal. Consiste en un ActionListener el cual según cual sea el Source del boton “accionado” realiza unas llamadas a la MAE tablero (modelo de la expansión) que alteran los valores del modelo que son enviados de vuelta a través de un observer.

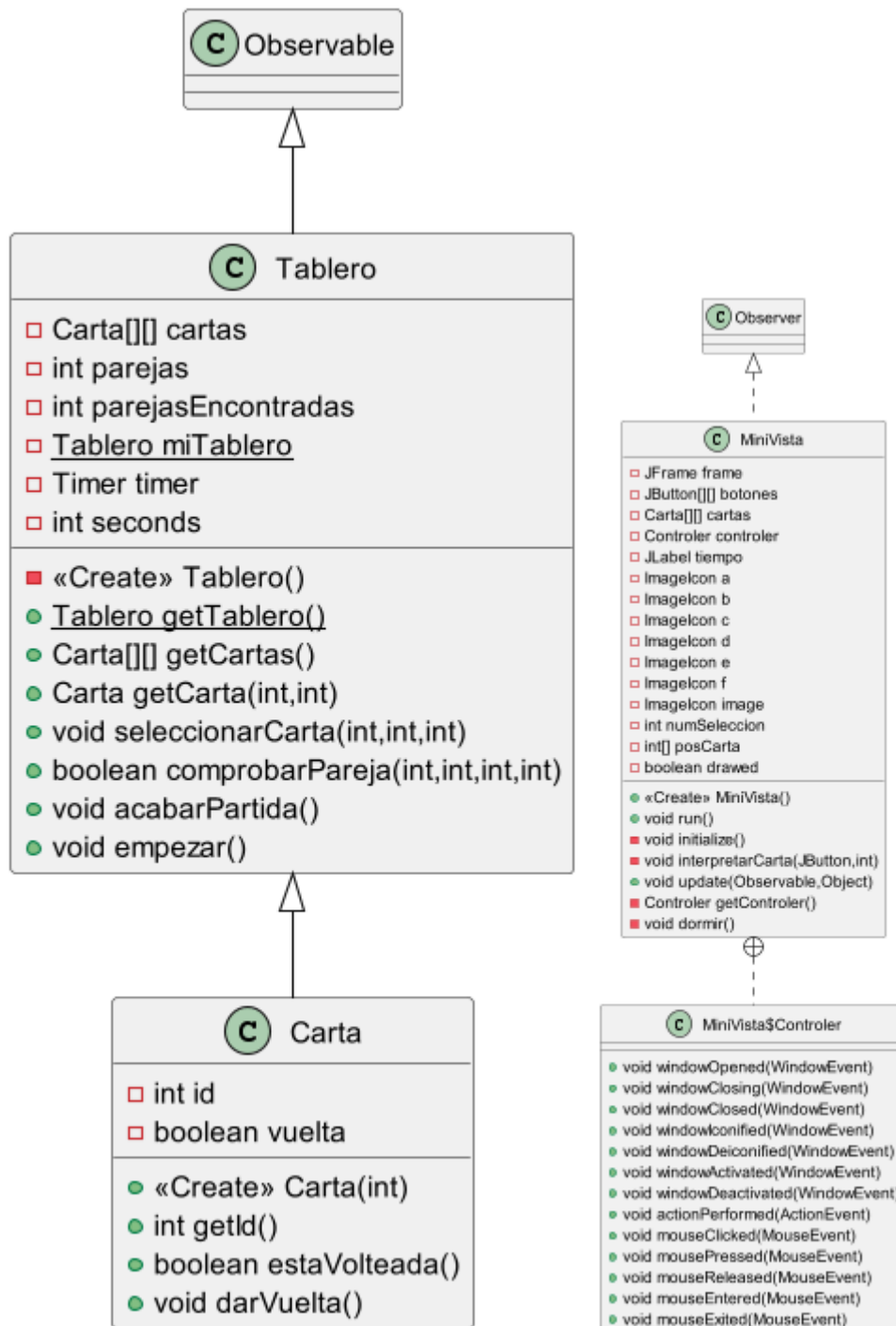
TamaDigOut

En el TamaDigOut los controladores a usar fueron 3, WindowListener para comunicar el comienzo de la partida, MouseListener para saber cuando el ratón entra en un bloque y por último un KeyListener que fue utilizado para comunicar las teclas de movimiento con el modelo y así poder mover el personaje.

En cuanto a técnicas destacable usadas en esta parte se puede destacar que a la hora de crear los botones le dimos un nombre que se corresponde con su posición en el tablero para así a la hora de hacer un `.getSource()` podamos obtener con facilidad de que bloque se trata. Con esa información es muy sencillo para el modelo modificar el bloque en cuestión en forma lógica y reenviar con un observer el Tablero actualizado.

Diagramas del 2º Sprint





[para ver mejor los diagramas ver la carpeta, están los png adjuntos.](#)

Reparto de horas y tareas

| Tareas | MENU | JUEGO TAMA | JUEGO DIGOUT | EXPANSION | DIAGRAMAS Y DOCUMENTACION |
|-------------------------|------|------------|------------------|-----------|------------------------------|
| Iker López | | 100% | 60% | 20% | 30% |
| Iván Andrés | | | 20% | | 70% |
| Nicolas Pasucal | | | | 10% | |
| Abdessamad | | | 20% | | |
| Jon Gete | | | | 70% | |
| Diego Pomares | x | x | x | x | x |
| | | | | | |
| | | | | | |
| HORAS TOTALES POR TAREA | 0 | 2-3 HORAS | 12 HORAS REHACER | 10 HORAS | 2 HORAS |