*Detalles de implementacion agregados al anterior informe:*

*-Para el ataque especial de MajinBoo (inutilizar a un personaje durante sus siguientes 2 turnos), se intento una especie de Proxy, siendo que toda accion que es sometida al ataque especial de majinBoo (ataqueBasico, ataqueEspecial, Mover y Transformar) son delegadas a la clase transformacion de dicho personaje, se crea una nueva transformacion "chocolate" que es cambiada por la actual transformacion y la cual implementa esos 4 metodos lanzando las correspondientes excepciones sin hacer nada mas. Ademas cuenta los turnos restantes y guarda en si la transformacion original del personaje para poder devolversela al finalizar el efecto.*

*(Cuando desde la partida se pasa de turno, el equipo que termino su turno revisa sus personajes para reducir el tiempo restante o eliminar el efecto del chocolate si ya pasaron los turnos correspondientes)*

*-Para la segunda transformacion de Gohan (SuperSayajin1 a SueprSayajin2) es necesario que sus compañeros (Goku y Picolo) esten en menos del 30% de vida. Asi que al querer transformar a Gohan, Se revisa si existen estos aliados, y de ser asi les pregunta a cada uno si cumplen la condicion de vida necesaria. Si cumplen todo esto se permite transformar a Gohan. (como cada personaje tiene guardado su equipo, y el equipo tiene referencia a todos los personajes integrantes, se pueden relacionar)*

*-Lo que seria la parte de la vista del juego, se tiene la clase "aplicacion" que engloba todo lo necesario, como ser:*

*CaminoController: Es quien se encarga de ir guardando los casilleros seleccionados para generar el camino cuando sea necesario (al pedir mover un personaje)*

*PersonajeController: Misma idea de caminoController, pero para guardar los personajes seleccionados.(Cuando pulse el boton de atacar, los participantes de ese ataque son los previamente guardados aca)*

*ContenedorPrincipal: Aca se guarda la barra de menu, vista del tablero, la consola, setea los botones, menues, etc..*

*Cada boton tiene asignado un handler especifico que se encarga de ejecutar la accion esperada siendo ellos quien interactuan con el modelo.*

*-Para la generacion random de consumibles, existe un GeneradorDeConsumiblesRandom que se ejecuta al pasar de turno en la partida, y tiene un .25 de probabilidad de generar uno de los 3 posibles consumibles en alguna posicion random del tablero.*

*-Los efectos en el tiempo que generan los consumibles (doble de velocidad o aumento de daño) son guardados en la transformacion del personaje que los consume, y ese "efecto" es usado a la hora de que la transformacion calcule el daño o velocidad (para saber la velocidad, la transformacion le pregunta a su efecto.. asi que si no se consumio nada, la transformacion tiene el efecto: "SinEfectoEspecial" que simplemente devuelve el numero original esperado. En el caso de haber consumido por ejemplo, una NubeVoladora, el efecto pasa a ser este que al preguntarle por la velocidad, devuelve el doble de la original)*