

Informe de documentacion Model Alpaca Emblem

Profesor: Alexandre Bergel
Fecha: 7 de septiembre de 2019
[Link de Trabajo](#)

Autor: Sebastián Sepúlveda A
User GitHub: [sesepulveda17](#)
RUT: 19.640.031-1

Resumen

En el presente informe, se presentará un resumen con diagramas *UML* (Unified Modeling Language) de la estructura del código para el modelo del juego “*Alpaca Emblem*”. La estructura del código fue proporcionada a través del siguiente [Template](#) ¹, el cual tenía errores y malas implementaciones de métodos que debían ser arregladas y mejoradas. Para esta parte además se solicitó implementar nuevas funcionalidades al código, como poder lograr **Combates**, **Intercambio de Items** e implementar una nueva unidad **Sorcerer**, cuyas características están especificadas en el enunciado de la tarea ².

En los siguientes diagramas no se incluyen los métodos de cada clase debido a la gran cantidad de métodos en cada una de ellas, lo que dificulta su visualización. Sin embargo se pueden encontrar los archivos UML en los siguientes links: [UML_Items](#), [UML_map](#), [UML_units](#)

Diagrama UML Items

Ver [Figura 1](#). Cambios respecto al *Template*.

- Se añaden nuevos métodos `receive<Item>Attack`, `canAttack`, `magicAttack`, `canAttack` y `setOwner`, además de modificar el constructor de la clase abstracta para recordar valores como la *máxima cantidad de items* y la *maxima cantidad de vida*.
- Nueva clase abstracta `AbstractItemMagic` que hereda métodos de `IEquipableItem` y `AbstratItem`.
- Nuevos items mágicos: `Soul`, `Darkness`, `Light` que solo pueden ser usados por la unidad `Sorcerer`

Diagrama UML Unit

Ver [Figura 2](#). Cambios respecto al *Template*.

- Se añaden nuevos métodos: `equiItem`, `equipItemOther`, `equipItem<Nom_Item>`, `isItemFull`, `addItem`, `removeItem`, `attack`, `receiveAttack<Tipo_de_Ataque>`, `receiveRecovery`, `isInRange`, `giveItem`, `setEquippedItem`.
- Nueva funcionalidad para intercambio entre unidades: `give_Item`
- Nueva unidad que puede utilizar items mágicos: `Sorcerer`
- Mejoras: Cada unidad puede equiparse cuando tiene un item en su inventario y ese item es el que puede equiparse. Cada unidad tiene un tipo distinto de ataque (ataque normal, debil, fuerte, curacion o no ataca).

Diagrama UML Map

No hay cambios. Ver [Figura 3](#)

¹Cuerpo Docente CC3002, Segundo Semestre 2019, Universidad de Chile

²Revisar repositorio

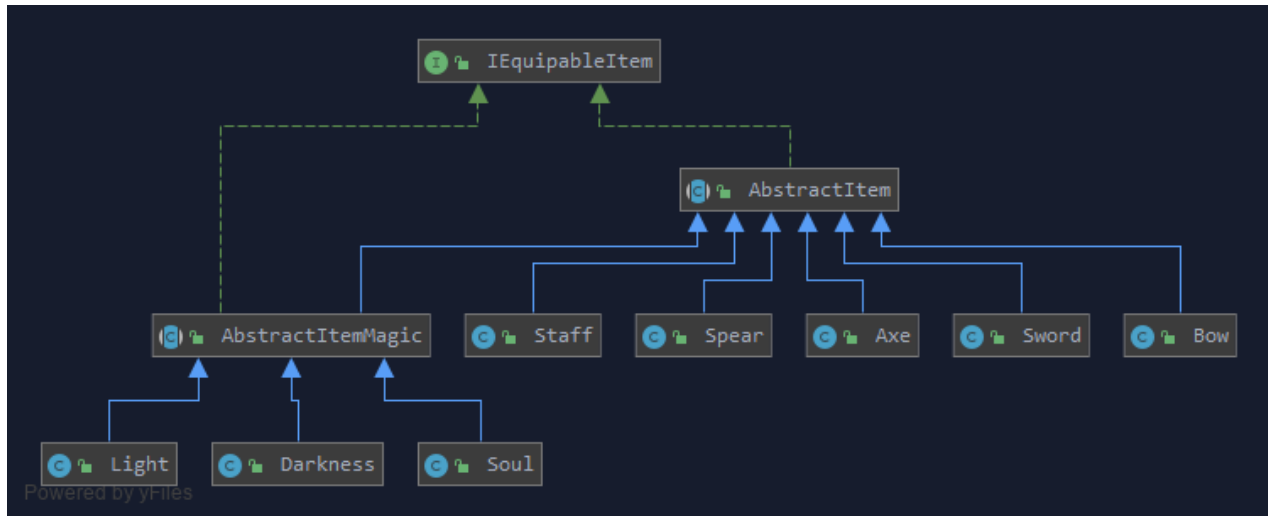


Figura 1: Diagrama UML items

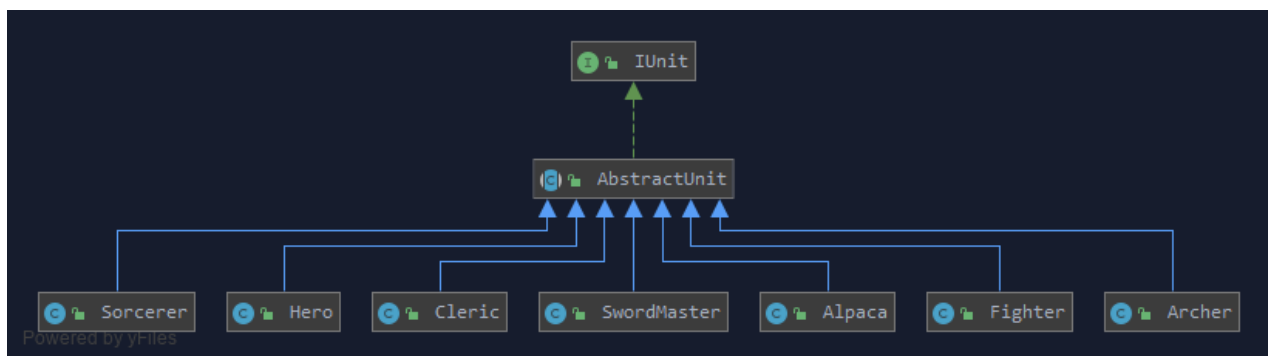


Figura 2: Diagrama UML unit

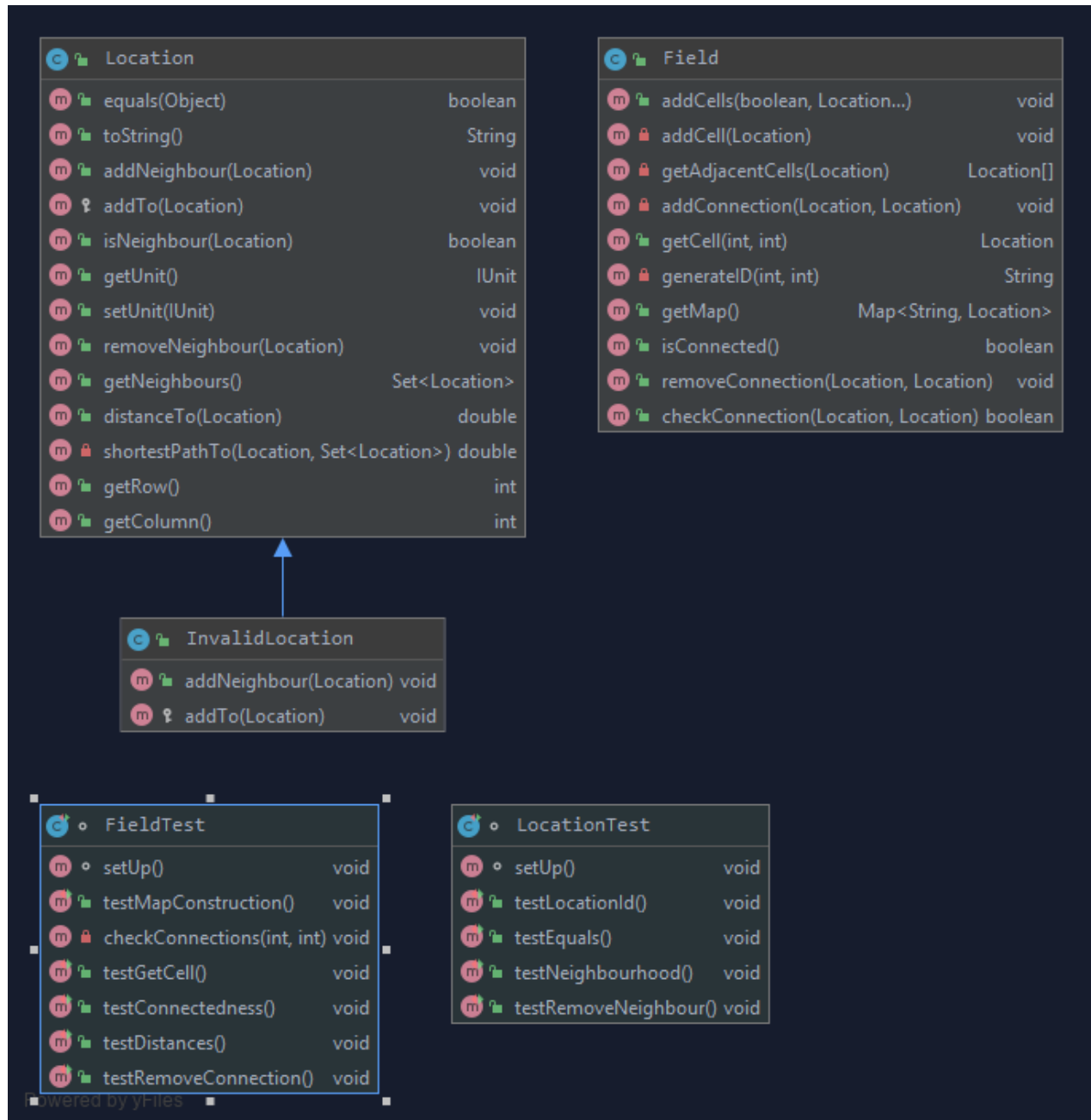


Figura 3: Diagrama UML map