AGAR Project Seconda revisione

Alessia Boní, Giovanni Bonura, Andrea Spinelli, Raffaele Terracino

Tecnologie utilizzate





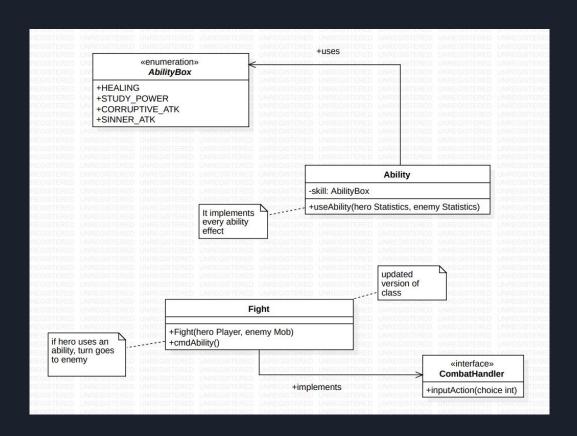
Punti della prima revisione

- Product backlog, primo sprint backlog
- Architettura del gioco (MVC)
- Inizializzazione e avvio del gioco
- Sistema di combattimento a turni: attacco
- Esplorazione del mondo di gioco
- Avvio del combattimento
- Prima progettazione database
- Storia del gioco

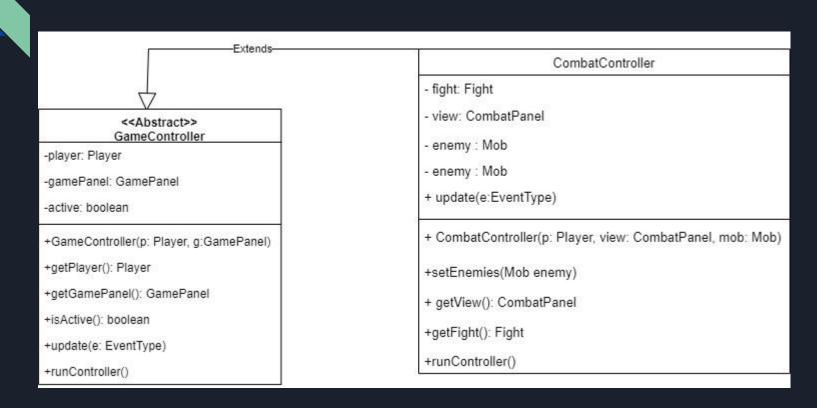
Secondo sprint backlog

Goal:	Completare il primo livello			IEH:	100						
ID	Item/Acceptance Critera	Tasks	SP	Owner	Planned hours	Giov	Ven	Sab	Mer	Giov	
6	"In qualità di giocatore, voglio poter utilizzare le abilità in modo tale da avere vantaggi in combattimento"		М			Remaining hours					
	Il giocatore deve poter: - Premere un bottone - Scelgliere l'abilità - Attaccare il nemico	Business logic: skill tree e combattimento		Giovanni							
		Controller combattimento: abilità		Alessia							
		View abilitá		Andrea							
7	"In qualità di giocatore, voglio poter prendere decisioni in modo tale poter influenzare il proseguimento del gioco"		L				Remaining hours				
	"Il giocatore deve poter: - Visualizzare almeno due pulsanti - Sceglierne solo uno - Influenzare (aumentare/diminuire) le	Business logic: NPC e dialoghi		Andrea							
		Disegno NPC su schermo		Raffaele							
	caratteristiche del gioco"	Controller: dialoghi		Andrea							
		Business logic: sistema decisionale		Raffaele							
		Controller: sistema decisionale		Raffaele							
		Realizzazione sprite NPC		Andrea							
		View: dialoghi		Andrea							
8	"In qualità di sistema, voglio poter fornire un primo livello di gioco in modo tale da introdurre la storia "		L				Rem	naining ho	ours		
	Il primo livello deve corrispondere alla parte iniziale della storia di gioco, contenere due mappe di gioco con varie situazioni di stallo	Connessione al DB		Giovanni							
		Business logic: livelli e mappe		Alessia							
		Realizzazione sprite seconda mappa		Andrea							
		Controller livelli e mappe		Raffaele							
25	Bug: più finestre si aprono durante lo switch tra combattimento e esplorazione		M								
	Deve essere presente un solo JFrame comune per esplorazione e combattimento, per tutto il gioco			Andrea							

Sistema di abilità - Bonura



Controller delle abilità - Bonì



Sistema di mappe e livelli - Bonì

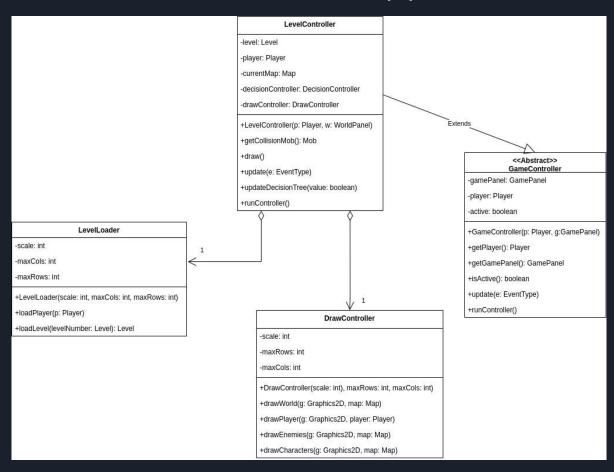
Map -mapEnemies : Mob[][] - mapNpc : Npc [][] worldTiles : BufferedImage[][] -endMap : boolean[][] + getEndMap(): BufferedImage[][] +removeMob(index:int) +removeNpc(npc Npc) +Map(map Map) +getEndMap(): boolean[][] +getWorldTiles(): BufferedImage[][] +getMapEnemies(): Mob[][] +getNpc(): Npc[][] getEnemy(x: int, y : Int) : Mob +aetTile(x; int. v; Int) : BufferedImage +getEndMap(x:int, y:int):boolean Level counter: int - maps: ArrayList<Map> + switchMap() + Level(ArrayList<Map> maps) + getCurrentMap(): Map

```
<<Interface>>
                       IMapBuilder
       + buildMapNpc(x: int, y: int):
       + buildWorldTiles(x : int, y : int):
       + buildEndMap(x : int, y: int)
       +buildMapEnemies(x:int, y:int)
       +build(): Map
       +reset()
                         Use
                    MapBuilder
-mapEnemies : Mob[][]
- mapNpc : Npc [][]
worldTiles : BufferedImage[][]
-endMap : boolean[][]

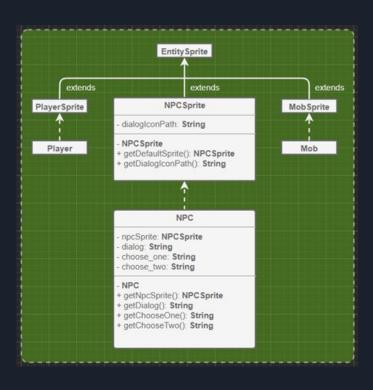
    worldX : int

- worldY : int
+build(): Map
+reset();
+ buildMapNpc(x: int, y: int):
+ buildWorldTiles(x : int, y : int):
+ buildEndMap(x:int, y:int)
+buildMapEnemies(x:int, y:int)
+addMob(Mob enemy, int x,int y)
+addNpc(Npc npc, int x,int y)
+addWorldTile(BufferedImage tile, int x,int y)
+addEndTile(x:int, y: int)
+ set...: void
```

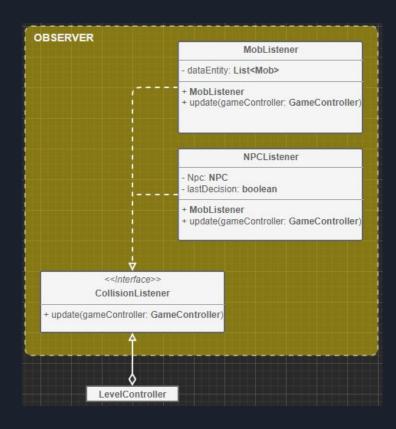
Controller livelli e mappe - Terracino



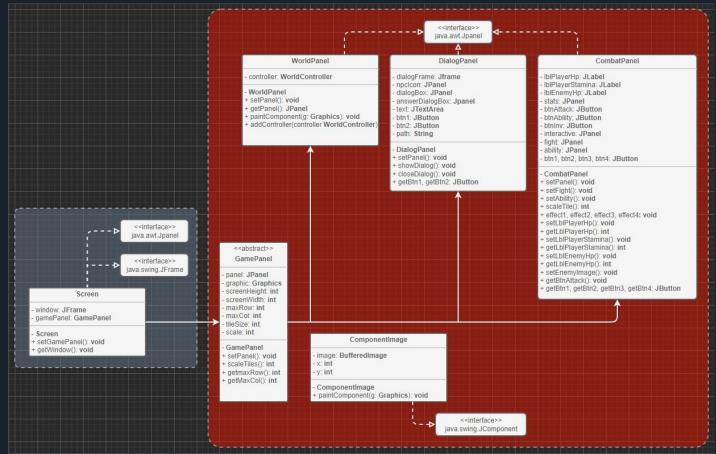
NPC e controller dialoghi, Business Logic - Spinelli



NPC e controller dialoghi, Controller - Spinelli

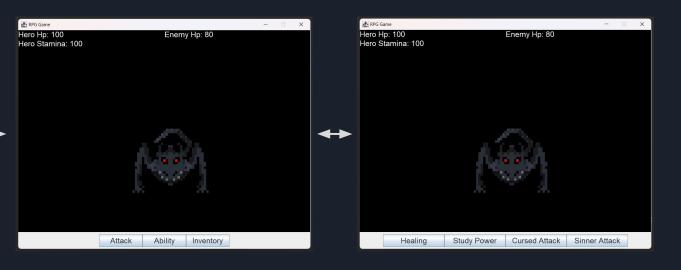


NPC e controller dialoghi, View - Spinelli

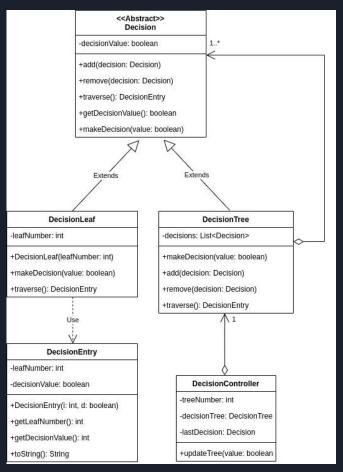


View: Abilità - Spinelli

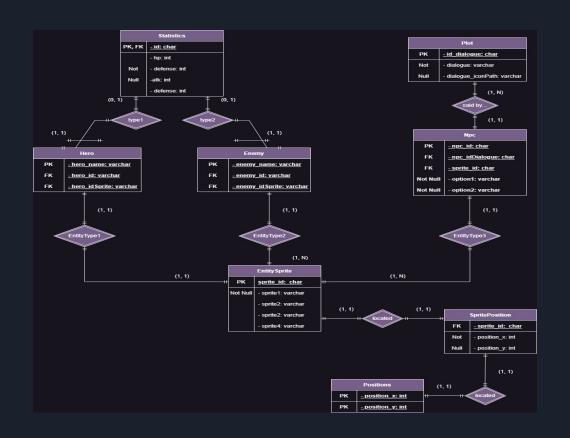
CombatPanel - IbiPlayerHp: JLabel - IblPlayerStamina: JLabel - IblEnemyHp: JLabel - stats: JPanel - btnAttack: JButton - btnAbility: JButton - btnlnv: JButton - interactive: JPanel - fight: JPanel - ability: JPanel - btn1, btn2, btn3, btn4; JButton - CombatPanel + setPanel(): void + setFight(): void + setAbility(): void + scaleTile(): int + effect1, effect2, effect3, effect4: void + setLblPlayerHp(): void + getLblPlayerHp(): int + setLblPlaverStamina(): void + getLblPlayerStamina(): int + setLblEnemyHp(): void + getLblEnemyHp(): int + setEnemyImage(): void + getBtnAttack(): void + getBtn1, getBtn2, getBtn3, getBtn4: JButton



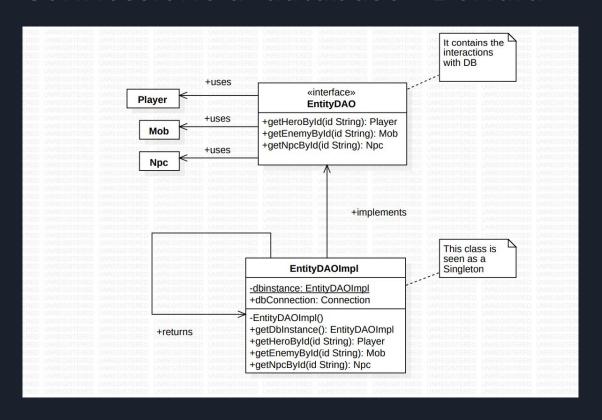
Sistema decisionale - Terracino



Database delle entità di gioco - Bonura



Connessione al database - Bonura



TDD e incremento del secondo sprint