

Programare Orientată pe Obiecte

PokeMock - joc management creaturi fantastice

Tema proiectului

În ideea jocurilor Pokemon, proiectul intenționează să permită utilizatorului să crească și să evolueze creaturi fantastice de diferite tipuri, prin hrănirea lor, oferirea de îngrijire sau folosirea lor pentru luptă.

Obiecte

1. Being (Interfață cu diferite câmpuri - viață maximă, vitalitate, putere, apărare)
2. Creature (Clasă abstractă care definește o creatură care poate fi îngrijită de jucător, implementează un Being)
3. Monster (Clasă care definește un inamic cu care se poate lupta o Creatură, implementează un Being)
4. Dragon (Tip de creatură, extinde un Creature. Are, în general, și putere mare, și apărare mare)
5. Golem (Tip de creatură, extinde un Creature. Are, în general, apărare foarte mare, dar putere mai redusă)
6. Phoenix (Tip de creatură, extinde un Creature. Are, în general, putere mare, dar apărare mai redusă)
7. InventoryItem (Clasă abstractă care definește un obiect care poate fi ținut în inventar)
8. FoodItem (Obiect care poate fi dat ca hrană unei creaturi - îi crește vitalitatea, și poate să îi crească puterea sau apărarea unei creaturi, extinde un InventoryItem)
9. EvolutionAmulet (Obiect care poate fi oferit unei creaturi pentru a o evolua - îi crește toate elementele de status și rangul, extinde un InventoryItem)
10. Amuletă de înviere (Obiect care poate fi oferit unei creaturi pentru a o învia în cazul în care este moartă, extinde un InventoryItem) - nu e implementat
11. Amuletă de contopire (Obiect care poate fi folosit pe două creaturi pentru a le contopi și transforma într-o creatură mai puternică, extinde un InventoryItem) - nu e implementat
12. Inventory (Obiect cu număr limitat de slot-uri unde pot fi ținute obiecte de tipul InventoryItem)
13. Narrator (Clasă cu funcții statice folosite pentru a afișa text / narațiune la diferite acțiuni) - nu e implementat
14. BattleService (Clasă helper care efectuează toate operațiunile legate de lupta dintre un Creature și un Monster)
15. Game (Singleton folosit pentru acțiunile care pot fi luate în cadrul jocului, prin GameState)
16. GameState (Interfață care expune două metode, display și handleInput. Template-ul pentru fiecare meniu din joc)
17. MainMenuState (Implementează GameState, oferă acces la meniul de Creaturi și Inventar)
18. CreatureSelectMenuState (Implementează GameState, permite selectarea unei creaturi)

19. SingleCreatureActionState (Implementează GameState, permite acționarea asupra unei creaturi)
20. UseItemOnCreatureState (Implementează GameState, permite utilizarea unui InventoryItem pe o creatură anterior selectată)
21. FightState (Implementează GameState, prezintă lupta dintre o Creatură și un Monstru)
22. FightFinishedState (Implementează GameState, prezintă rezultatul luptei dintre o Creatură și un Monstru)
23. InventoryMenuState (Implementează GameState, permite selectarea unui InventoryItem)
24. InventoryItemActionState (Implementează GameState, permite acționarea asupra unui InventoryItem)
25. InventoryUseTargetState (Implementează GameState, permite selectarea creaturii pentru care să se aplice InventoryItem-ul anterior selectat)
26. BeingFactory (Clasă helper care permite crearea claselor de tip Being)
27. FoodFactory (Clasă helper care permite crearea claselor de tip FoodItem)
28. NameFactory (Clasă helper care permite generarea numelor pentru Being sau pentru InventoryItem)
29. ItemTemplate (Record cu 2 câmpuri pentru InventoryItem)
30. InputHelper (Clasă helper care expune funcții pentru input)
31. UIHelper (Clasă helper care expune funcții pentru afișarea meniurilor)
32. CreatureType (Enum care descrie tipurile de Creature)
33. StatType (Enum care descrie tipurile de statistici ale unui Being)

Acțiuni

1. Vezi listă creaturi - afișează numele și tipul creaturilor deținute de jucător
2. Vezi detalii creatură - afișează informații despre o anumită creatură deținută de jucător
3. Vezi inventar - afișează toate obiectele din inventarul jucătorului
4. Folosește obiect inventar
 - a. Folosește mâncare - hrănește o creatură
 - b. Folosește amuletă de evoluție - crește rangul unei creaturi
 - c. Folosește amuletă de înviere - învie o creatură moartă - nu e implementat
 - d. Folosește amuletă de contopire - contopește două creaturi într-una mai puternică - nu e implementat
5. Mângâie creatură - crește puțin vitalitatea unei creaturi - nu e implementat
6. Trimite creatură la luptă - Plasează Creatura într-o luptă cu un Monster. Pierde viață și poate muri, dar dacă câștigă, primește un bonus de stats și poate descoperi InventoryItems utile pentru jucător.
7. Alungă o creatură - Elimină o creatură din lista de creaturi deținute de jucător - nu e implementat
8. Îngroapă o creatură - Elimină (îngroapă) o creatură moartă din lista de creaturi deținute de jucător - nu e implementat
9. Aruncă obiect - Elimină un obiect din lista de obiecte deținute de jucător

De implementat

Pe lângă ce e marcat ca „nu e implementat”, există numeroase feature-uri de implementat

Creaturi

- Redenumirea creaturilor
- Creștere automată în rank prin XP sau altfel
- Mai multe tipuri de creaturi (cu abilități speciale)
- Evoluție specifică pentru diferitele tipuri de creaturi
- Statistică Noroc, care poate duce la mai multe obiecte descoperite după o luptă

Obiecte

- Obiect pentru extinderea spațiului în inventar

Mecanisme

- Luptă de antrenament între două Creature
- Trimiterea unei creaturi la muncă pasivă pentru a deveni mai puternică de una singură
- Luptă în grup

Foarte multe altele pot fi gândite.