JOC RPG

Programare orientată pe obiecte

Student: Florescu Denis

Grupa 3124a

Profesor coordinator: s.l. dr. inf. Bilius Laura-Bianca

**Cuprins:**

1. Tema proiect
2. Elemente Teoretice

2.1. Descrierea Problemei

2.2. Abordarea Teoretică a Problemei

2.3. Elemente Specifice POO

3. Implementare

3.1. Tehnologii Folosite

4. Analiza Soluției Implementate

4.1. Formatul Datelor de I/O

4.2. Studii de Caz

5. Manual de Utilizare

6. Concluzii

1. Tema proiect

Tema proiectului este un joc RPG ( role-playing game ). Un joc RPG este un joc în care fiecare participant își asumă rolul unui personaj care poate interacționa în cadrul unei lumi imaginare. Multe RPG-uri sunt situate în medii fantastice sau științifico-fantastice. Aplicația Joc RPG este un joc de aventură care se desfășoară într-o lume științifico-fantastică fascinantă. Scopul principal al jocului este de a oferi jucătorilor o experiență captivantă în care ei devin eroii unei povești epice. Fiecare jucător va fi provocat să-și conducă eroul ales prin diferitele etape ale aventurii, înfruntând obstacole și încercând să învingă creatura finală pentru a completa jocul cu succes.

Jucătorul va fi condus printr-o serie de etape și locații diverse, fiecare cu propriile lor provocări și secrete de descoperit. Această aventură va fi plină de suspans și va oferi jucătorilor posibilitatea de a explora și de a interacționa cu lumea din jurul lor.

Pe măsură ce progresează în aventură, jucătorii vor întâlni diferite obstacole sub formă de evenimente care le vor testa abilitățile și strategiile. Aceste obstacole pot include întâlniri cu creaturi periculoase și situații de conflict.

Cea mai mare provocare a jocului este înfruntarea creaturii finale, o ființă puternică și malefică care trebuie învinsă pentru a completa aventura cu succes. Această confruntare finală va testa tot ce a învățat și a câștigat jucătorul pe parcursul jocului și va oferi un moment culminant captivant.

Am ales această temă deoarece mi s-a părut o provocare să fac un joc de la zero, și în același timp o ambiție de a vedea ce poate ieși la final.

1. **Elemente Teoretice**

Problema specifică abordată în acest proiect este dezvoltarea unui joc RPG care să includă interacțiuni cu mediul, utilizarea obiectelor și gestionarea resurselor.

**2.1. Descrierea Problemei**

Proiectul de față implică dezvoltarea unui joc RPG (Role-Playing Game) în care fiecare participant își asumă rolul unui personaj care poate interacționa cu lumea imaginară a jocului. Scopul principal este de a crea o experiență imersivă pentru jucători, oferindu-le posibilitatea de a explora, a lupta, a colecta obiecte și a interacționa cu diverse elemente și personaje non-jucător (NPC-uri) într-un mediu virtual.

**\* Elemente cheie:**

**\* Gestionarea Inventarului:**

Jucătorii vor putea colecta și gestiona diverse obiecte (arme, poțiuni) pe parcursul jocului. Aceasta necesită implementarea unui sistem eficient de inventar care să permită stocarea și utilizarea obiectelor.

Este necesară alocarea dinamică a memoriei pentru a gestiona adăugarea și eliminarea obiectelor din inventar.

**\* Sistemul de Luptă și Abilități:**

Personajele jucătorilor vor avea abilități și statistici care vor influența modul în care se desfășoară luptele. Sistemul de luptă trebuie să fie echilibrat și să ofere diferite strategii și tactici de combatere a inamicilor.

Implementarea unui sistem de creștere a abilităților și nivelului personajului pe baza experienței acumulate.

**\* Gestionarea Resurselor Jocului:**

Resurse trebuie gestionate eficient pentru a asigura performanța optimă a jocului.

**\* Implementarea Evenimentelor de Joc:**

Sistemul de evenimente va gestiona acțiunile jucătorilor și va declanșa reacțiile corespunzătoare din partea jocului. Acest lucru include gestionarea input-ului de la tastatura, precum și reacțiile la interacțiunile cu mediul și NPC-urile.

**2.2. Abordarea Teoretică a Problemei**

**Manipularea Fișierelor:**

Citirea și Scrierea Datelor: Vom folosi operațiile de citire și scriere pentru a salva și încărca datele jocului, cum ar fi progresul jucătorului, povestea și alegerile jucătorului.

**Modul Text:**

Afisarea Textului pe Ecran: Vom învăța cum să afișăm text pe ecran pentru a comunica informații jucătorului, cum ar fi mesaje de avertizare și statistici ale personajului.

**Programare Orientată pe Obiecte (POO):**

Structurarea Codului: Vom folosi conceptele POO pentru a organiza codul în clase și obiecte, facilitând dezvoltarea și mentenanța jocului.

Utilizarea Claselor și Obiectelor: Vom crea clase pentru personaje, obiecte, inamici, și alte entități din joc și vom utiliza obiectele acestora pentru a gestiona starea și comportamentul lor.

**Alte Concepte:**

Gestionarea Evenimentelor: Vom implementa un sistem de gestionare a evenimentelor pentru a detecta și a reacționa la acțiunile jucătorului și la alte evenimente din joc, cum ar fi coliziunile sau interacțiunile cu obiectele.

**2.3. Elemente Specifice POO**

În cadrul acestui subcapitol, vom explora conceptele și elementele specifice ale Programării Orientate pe Obiecte care vor fi aplicate în rezolvarea problemei noastre legate de dezvoltarea unui joc RPG. Aceste elemente includ:

1. Clase și Obiecte:

Definirea Claselor: Vom identifica entitățile principale din joc, cum ar fi personaje, obiecte, inamici, etc., și vom defini clase separate pentru fiecare dintre acestea.

Crearea Obiectelor: Vom crea obiecte ale acestor clase în cadrul jocului, folosind constructori și inițializându-le cu date relevante.

2. Mostenire:

Utilizarea Mostenirii: Vom utiliza mostenirea pentru a crea ierarhii de clase care împart comportamente și caracteristici commune.

3. Compoziție:

Relații între Clase: Vom stabili relații între clase pentru a descrie conexiunile și dependențele dintre diferitele entități din joc.

4. Gestionarea Memoriei:

Alocarea și Dezalocarea Memoriei: Vom aloca și dezaloca dinamic memoria pentru obiectele noastre, utilizând operatorii new și delete pentru a asigura utilizarea eficientă a resurselor.

Prevenirea Scurgerilor de Memorii: Vom fi atenți la gestionarea corectă a memoriei pentru a preveni scurgerile de memorie și fragmentarea memoriei.

Aceste elemente specifice POO vor servi drept fundament pentru proiectarea și implementarea structurii de cod a jocului nostru RPG, permițându-ne să organizăm și să gestionăm eficient entitățile și funcționalitățile sale.

**3. Implementare**

Pentru a aborda rezolvarea temei propuse, am început prin a identifica cerințele și funcționalitățile necesare pentru dezvoltarea unui joc RPG. Am analizat aspecte precum interacțiunea jucătorului cu mediul virtual, gestionarea personajelor și obiectelor din joc, mecanicile de luptă, sistemul de inventar și altele.

Am dorit să creez o structură modulară și flexibilă. Am optat pentru o abordare orientată pe obiecte, în care am definit clase separate pentru diferitele entități din joc, precum personaje, obiecte, inamici și alte elemente.

Am planificat utilizarea conceptelor precum moștenirea și compoziția pentru a organiza și gestiona entitățile jocului într-un mod eficient. Am dorit să asigur o gestionare corectă a memoriei.

**3.1. Tehnologii Folosite**

Pentru construcția proiectului, am utilizat următoarele tehnologii, limbaje de programare și medii de dezvoltare integrat (IDE):

Limbajul de programare: C++.

Mediul de dezvoltare: Code Blocks.

GitHub pentru a stoca și gestiona repository-ul proiectului.

**4. Analiza Soluției Implementate**

**4.1. Formatul Datelor de I/O**

**Formatul Datelor de Intrare:**

Datele de intrare precum numele jucătorului sunt introduse de utilizator prin intermediul consolei. În aceeași manieră sunt introduce și răspunsurile utilizatorului în cadrul alegerilor din joc.

**Formatul Datelor de Ieșire:**

Datele de ieșire sunt afișate utilizatorului prin intermediul consolei.

Formatul datelor de ieșire este structurat într-un mod clar și ușor de înțeles pentru utilizator, folosind formatare adecvată, precum spații, tab-uri, linii noi sau delimitări cu caractere ASCII.

În general, formatul datelor de intrare și de ieșire trebuie să fie intuitiv și coerent cu scopul și funcționalitatea programului, asigurându-se că utilizatorul poate interacționa eficient și corect cu aplicația.

**4.2. Studii de Caz**

**Gestionarea Inventarului:**

Date de intrare: Utilizatorul are deja un set de date minime introduse în inventor, care pe parcurs pot fi modificate.

Cum trebuie să le interpretăm: Aplicația trebuie să preia aceste date și să le stocheze într-un mod corespunzător pentru a putea fi gestionate în continuare.

Date de ieșire: Obiectul adăugat în inventar poate fi vizualizat prin metoda de “inv.verifica\_stare()”, care va afișa chiar mai multe informații decât doar despre obiectul adăugat.

**Numele**

Numele care trebuie completat în caz contrar va primi un valoare default:

Date de intrare: Utilizatorul introduce numele. În cazul în care nu furnizează un nume, sistemul va folosi un nume implicit.

Cum trebuie să le interpretăm: Aplicația trebuie să verifice dacă a fost furnizat un nume. În caz negativ, va folosi un nume implicit.

Date de ieșire: Aplicația afișează numele personajului, fie acesta introdus de utilizator, fie cel implicit, confirmând astfel faptul că a fost salvat cu succes.

**5. Manual de Utilizare**

**Descrierea interfeței produsului program:**

În cadrul proiectului, interfața grafică este la nivel de consolă. Partea interactivă limitându-se la selectarea unor variante din meniu și completarea anumitor campuri cu variante de y (da) și n (nu).

**Care sunt datele de intrare și cum sunt introduse:**

Primele date de intrare sunt reprezentate de alegerea unui nume pentru a începe jocul. Dacă utilizatorul nu a introdus un nume, i se va atribui un nume implicit. Următoarele date de intrare sunt aceleași ca în subcapitolul anterior.

**Ce obținem la ieșire și în ce format:**

La ieșire, utilizatorului i se vor furniza informațiile în modul consolă, sub formă de text. Nu există o implementare grafică adițională. De asemenea informațiile de la șfârșitul jocului vor fi regăsite într-un fișier text.

**Logica jocului:**

Jocul este implementat astfel încât să fie intuitiv și ușor de folosit. Mai mult, la început, jocul va avea un set de reguli ce îl va ajuta pe utilizator să înțeleagă mecanica din spatele acestuia.

**6.Concluzie**

Avantaje:

Experiență de utilizare captivantă: Jocul oferă o experiență captivantă și interactivă pentru utilizatori, permitându-le să exploreze și să interacționeze într-o lume imaginară.

Diversitatea funcționalităților: Jocul include o gamă largă de funcționalități, cum ar fi gestionarea inventarului, explorarea lumii virtuale, luptele cu monștrii și altele, oferind o experiență bogată și variată utilizatorilor.

Flexibilitate în gameplay: Gameplay-ul este flexibil și personalizabil, permițând jucătorilor să-și aleagă propriile strategii.

Dezavantaje:

Posibile probleme de echilibrare: Unele aspecte ale gameplay-ului, cum ar fi echilibrarea puterilor și dificultatea nivelurilor, ar putea necesita ajustări suplimentare pentru a asigura o experiență optimă pentru toți jucătorii.

Optimizare performanță: În anumite circumstanțe, jocul ar putea să se confrunte cu probleme de performanță sau de încărcare lentă.

Direcții viitoare:

Extinderea conținutului: Proiectul poate fi continuat prin adăugarea de noi caracteristici, personaje și alte elemente de conținut pentru a îmbogăți și diversifica experiența jucătorilor.

Optimizare și polizare: Este important să se continue optimizarea și perfecționarea jocului pentru a asigura o experiență plăcută pentru toți utilizatorii.

În concluzie, proiectul nostru de joc a reușit să ofere o experiență distractivă și captivantă utilizatorilor, iar cu atenție continuă la detalii și îmbunătățiri suplimentare, are potențialul de a deveni și mai remarcabil în viitor.