Realizarea unui platforme de învățare bazată pe gamification (escape room)

Proiect realizat de:

* Bolohan Stefan Tudor
* Ghiorghita Adrian Dragos
* Pricopi Robert Alexandru
* Visan Robert Cristian

Cuprins

**Introducere……………………………………………………3**

MOTIVATIA LUCRARII………………………………..3

AVANTAJE ALE ESCAPE ROOM……………………..3

**Descrierea Implementarii……………………………………5**

CONCEPTE POO………………………………………..5

JUSTIFICAREA SOLUTIEI…………………………….6

EXPLICAREA IN DELUNG A IMPLEMENTARII……7

MANUAL DE UTILIZARE……………………………12

**Diagrame UML…………………………………………….13**

**Concluzii……………………………………………………15**

**Bibliografie…………………………………………………16**

**Introducere**

**MOTIVATIA LUCRARII**

Un **escape room** este un joc de aventură în care jucătorii sunt blocați într-o cameră și trebuie să utilizeze elemente ale camerei pentru a rezolva o serie de puzzle-uri și să evadeze sau să îndeplinească o misiune într-un termen stabilit. Jocurile sunt versiuni fizice ale jocurilor video "Escape Rooms". Jocurile sunt stabilite într-o varietate de locații fictive, cum ar fi celule de închisoare, temnițe, temple, birouri, stații spațiale, etc. și sunt populare ca exerciții de team building.

Cea mai veche cameră se pare că a fost creată în 2006 deși informațiile nu sunt foarte exacte pe acest subiect, Escape Rooms a devenit popular în Statele Unite ale Americii, Japonia, Taiwan, Canada, Israel și China continentală în anii 2010. Primii operatori de escape rooms în locații permanente au fost deschise pentru prima dată în Asia și mai târziu a urmat în America de Nord, Europa, Australia, Noua Zeelandă, Rusia și America de Sud.

**AVANTAJE ALE ESCAPE ROOM**

Escape room este un mod bun de a comunica diferit. In cadrul escape room, ierarhia nu mai exista, ramane doar coeziunea. Toti participantii sunt in necunoscut, nimeni nu stie la ce sa se astepte. Raman impreuna si spera sa rezolve puzzle-urile in doar 60 de minute, astfel incat sa poata iesi din camera. Una dintre cheile succesului unui joc de evadare este mobilizarea diferitelor abilitati ale membrilor grupului. Un singur participant nu poate reusi sa plece, sunt obligati sa coopereze. Coeziunea si coordonarea sunt cuvintele cheie.

Blocat intr-o camera, grupul trebuie sa reuseasca sa evadeze intr-un timp limitat rezolvand o serie de puzzle-uri. Principiile jocului de evadare, in escape room este un scenariu din viata reala inspirat de lumea jocurilor video. In ultimul timp, s-a vazut o creste in interesul pentru escape room.

Dar recent, profesorii si formatorii au preluat acest joc de evadare pentru a-l transforma intr-un nou instrument educational. Participarea la un joc de evadare, este distractiva, un moment real de relaxare pentru a descoperi investigatii.

Esti indepartat de lume timp de saizeci de minute si impartasesti impreuna cu ceilalti groaza, fantezie, un univers magic sau realist. Avantajul unui escape room este ca acest joc este unic in felul sau, neimitabil.

Escape room in marime naturala indeplineste perfect cerintele unui manager care doreste sa consolideze coeziunea echipei sale, sa-si dezvolte inteligenta colectiva, sa stimuleze motivatia individuala si colectiva sau chiar sa integreze noi angajati printr-o activitate de stimulare a companiei. Jucatorii intra intr-o camera cu scenarii cu decor captivant, cautand cel mai mic indiciu in incercarea de a rezolva puzzle-urile in 60 de minute.

Toti uniti pentru a reusi sa scape, sentimentul de apartenenta dezvolta motivatia tuturor celor din cadrul grupului si creste dorinta de a servi colectivului pentru a atinge obiectivul comun. Este modalitatea perfecta de a prezenta abilitatile individuale, subliniind in acelasi timp valoarea efortului echipei, precum si rolul fiecarui membru si modul in care fiecare contribuie la succesul general.

**Descrierea Implementarii**

**CONCEPTE POO**

Conceptele principale ale POO folosite in implementarea noastra sunt:

* Clase
* Incapsulare ( inclusiv Friend Class )
* Metode

**Clase**

O**clasa** este o **colectie de obiecte** care au proprietati, operatii si comportamente comune. O clasa este o combinatie de stari (date) si de comportamente (metode). In POO o **clasa este un tip de date,** iar **obiectele sunt instante**aleacestuitipde date.

**Incapsulare**

Incapsulare este procesul prin care tinem datele si functiile separate de exterior. Daca luam tot exemplul precendent in care tu faci functionalitatea de adaugare in baza de date, e posibil ca acolo sa ai niste variabile in care sa tii username-ul si parola de conectare la baza de date. Ei bine, nu ai vrea ca o alta functionalitatea sa poata modifica variabila username sau din gresala ea sa fie suprascrisa de un username folosit in zona de creare useri. Prin incapsulare creem o capsula care delimiteaza interiorul de exterior.

**Functiile Friend**

Functiile friend sunt functii utilizate in programarea orientata pe obiecte in CPP care reprezinta functii de sine statatoare ce au access la atributele si metodele unei clase spre deosebire de metodele care sunt functii declarate in cadrul claselor.

**Metode**

Metodele sunt functii utilizate in programarea orientata pe obiecte care reprezinta functii declarate in cadrul claselor si sunt apelate prin intermediul unui obiect spre deosebire de functiile friend care sunt functii de sine statatoare ce au access la atributele si alte metode ale clasei.

**JUSTIFICAREA SOLUTIEI**

Noi ca solutie am propus stilizarea tipului classic al jocului Escape Room. Am vrut sa il facem in forma unui Text Based deoarece folosirea unui engine ar fi necesistat mult mai multe cunostinte, atat asupra conceptelor de POO dar si despre folosirea corespunzatoare si corecta a engine-ului.

De asemenea noi am preluat acest concept al jocului si l-am mai stilizat in forma unui “dungeon” fiecare camera a acestuia avand o diferita intrebare ce necesita un raspuns pentru completarea acesteia.

Pentru a evita raspunsurile gresite fara consetinte am adaugat un system pentru viata jucatorului. Pentru corectitudinea user-ului a fost folosit un timer, mai exact un timp exact pentru completarea dungeon-ului, timpul incepand in momentul in care user-ul primeste prima intrebare.

**EXPLICATIA INDELUNG A IMPLEMENTARII**

Am inceput prin introducerea bibliotecilor necesare pentru realizarea codului.

Librariile folosite de noi in cod sunt:

1. Iostream
2. String
3. Vector
4. Algorithm
5. Chrono
6. Fstream
7. Sstream
8. Iostream: aceasta biblioteca a fost folosita in scopul citirii si afisarii. Cu ajutorul acesteia am fost capabili sa detectam locul unde se produceau anumite erori dar si sa ne verifica ca lucrurile functioneaza in modul cum ne-am asteptat. De asemenea am mai folosit “iostream” pentru a creea un fel de interfata in consola si pentru a afisa intrebarile, hinturile, iesirile din camera, viata user-ului dar si pentru a putea citi raspunsul la fiecare intrebare.
9. String: acesta a fost folosit pentru a putea lucra, prelucra si in principiu folosi enunturi. Am preferat aceasta librarie din cauza ca, spre deosebire de char, acesta este mult mai buna la gestionarea memoriei, noi avand intrebari, raspunsuri, hinturi de dimensiuni variabile, cu elemente speciale care puteau genera frangementari in enunturile intrebarii.
10. Vector: acesta a fost folosita pentru generarea unui vector cu dimensiune variabila. Noi am folosit unuil pentru obiectele generate clasei (obiecte despre care vom vorbi mai amanuntit mai tarziu), unul pentru a putea afla positizia user-ului, si unul pentru iesirile din camere. Acesta mai bun decat un vector alocat static deoarece biblioteca ofera metode utile care au ajutat in eficientizarea codului.
11. Algorithm: acesta ne ofera metode pentru probleme des intalnite, cu scopul de a evita sa le creem noi, micsorand riscul aparitiei unei erori neasteptate care sa nu ne blocheze din implementarea codului. Unele erori generate de probleme des intalnite in programare pot fi foarte costisitoare din punct de vere a timpului, facand creearea proiectului mult mai lenta.
12. Chrono: a fost folosit pentru a putea genera timerul pentru joc, aceeasta fiind mult mai buna ma mangementul timpului decat biblioteca asemanatoare cu cea <ctime>. Acesta produce mult mai putine erori din punct de verede al calcularii timpului decat <ctime>.
13. Fstream: aceasta a fost folosita pentru a reusi sa citim din fiesierele care contin intrebarile, dar si pentru scierea sau incarcarea unui fisier de tip save, pentru a nu obliga utilizator sa stea fortat pentru tot parcusul jocului.
14. Sstream: Biblioteca sstream din C++ face parte din C++ Standard Template Library (STL) și oferă o modalitate de a citi și scrie într-un șir ca și cum ar fi un flux, similar cu citirea dintr-un fișier sau consolă. Clasa stringstream este clasa principală din biblioteca sstream și vă permite să citiți și să scrieți date într-un buffer de șir. Acest lucru poate fi util pentru o varietate de sarcini, cum ar fi analiza șirurilor, convertirea numerelor în șiruri și invers și multe altele. Unele funcții comune în clasa stringstream includ str(), str(string), operator>> și operator<<.

Urmatoarea parte a codului este compusa din cateva variante globale. Acest lucru nu este recomandat pentru ca limiteaza accesul in restul codului a folosirii variabilelor cu acelasi nume. Pentru a evita sa le repetam numele acestea au fost scrise caps, tip de scriere a variabilelor diferit de modul obisnuit (cu litere mici si cuvintele din numele variabilei despartite prin “\_”).

Urmeaza definirea clasei principale care contine variabile si metode. Variabilele sunt de tip private pentru a evita accesul la datele intrebarii de catre oricine. Variabilele folosite in acesta clasa sunt:

* question este întrebarea pe care utilizatorul trebuie să o răspundă pentru a părăsi camera
* answer este răspunsul corect la întrebarea din camera
* hint este un indiciu care poate fi oferit utilizatorului pentru a ajuta la răspunsul întrebării
* exits este un vector de string-uri care stochează toate ieșirile disponibile din camera
* is\_completed este un bool care indică dacă utilizatorul a răspuns corect la întrebarea din camera
* is\_end este un bool care indică dacă camera este camera finală a jocului

Declarate de timp public, se pot regasi metodele clasei respective. Printre metodele prezente in clasa se pot distinge:

* setters: folositi pentru a stabili fiecare atribut al fiecarui obiect create. Printre acestia se numara si constructorul default.
* getters: folositi pentru a putea creea o functie de afisare ce avea ca scop verificarea citirii din fisier
* destructor: folosit pentru eliberarea memoriei unde a fost stocate toate informatii deoarece sunt destul de costisitoare din punct de vedere a memoriei
* friend class: am folosit friend class pentru a putea face un fel de separare intre metodele de functionalitate si metodele de constructive ale jocului.

Urmatoarea clasa prezentata in cod este clasa de “Map” care prezinta functionalitatea jocului. Acestea sunt:

* validMove: acesta metoda are rolul de a restrictiona posibilatea user-ului de a se misca in afara hartii, dar si a creea anume shortcut-uri prin miscarea intre camera appropriate dar nu si interconectate
* printRoom: este metoda pentru creeata pentru a putea prelua raspusul la intrabari, creeaza savefile-uri dar si validitatea raspunsului. Aceasta metoda este foarte complexa deoarece ne creeaza aprope toata functionalitatea jocului, face toate elementele esentiale.
* answerQuestions: este metoda ce verifica daca raspunsul este corect sau gresit, dar acesta genereaza si afisarea hint-ului daca a fost prima incercare in a raspunde la intrebarea din acea camera, dar si cea care scade viata user-ului in cazul in care raspunsul este gresit si hint-ului a fost deja afisat.
* showHint: a fost creata strict pentru a afisa hint-ul intrebarii corespunzatoare
* getMove: este metoda ce asigura mutarea dintr-o camera in alta. Aceasta afiseaza lista de posibile iesiri pe care o compara cu iesirea citita de la tastatura de catre utilizator. Metoda afiseaza un mesaj de eroare in cazul in care user-ului introduce o iesire care nu este prezenta in list.

Dupa metodele clasei “Map” intram in componenta “main()” a codului. Aceasta contine un fel de interfata prin intermediul textului afisat. Ulterior user-ului trebuie sa introduca una din optiunile prezentate in meniu. Acestea sunt:

1. Start Game : care are rolul de a incepe un joc si de cere un input de la user in legatura cu fisierul in care doreste sa ii fie salvat jocul
2. Load Game: care la fel cere utilizatorului sa introduca fisierul din care doreste sa preia salvarea, este sarind peste intrebarile la care deja a raspuns.
3. Credits: in care sunt regasiti toti participantii la creearea acestuil proiect in ordine alfabetica
4. Exit: functia de iesire din joc, care a afiseaza un mesaj de apreciere pentru timpul acordat parcugerii jocului de catre utilizator.

In urma afisarii meuniului au fost instantiate variabile termporare ce au un rol de transmitere a datelor catre variabilere corespunzatoare, urmand ca mai apoi sa fie introduce in vectorului de camere.

**MANUALUL DE UTILIZARE**

Instructiuni pentru compilarea si rularea aplicatiei

Cerintele minime pentru compilate:

* MinGW (G++ compiler)
* Windows OS

Pentru rulare:

1. Din command prompt va plasati in directorul in care se afla fisierul main.cpp
2. Se ruleaza fisierul .exe create

In urma rularii:

* Se va deschide un command promp cu meniul generat de inceputul codului si alegerea fuctiei

**DIAGRAMELE UML**

**Diagrama de clase**

Diagram

Description automatically generated

**Giagrama de activitati**

Diagram

Description automatically generated

**CONCLUZII**

Acest proiect a avut ca scop realizarea un joc cu scop educativ dar si fixarea notiunilor necesare realizarii acestuia. Astfel că, de-a lungul acestui referat, au fost explicate noțiuni de teorie, concepte de programare orientată pe obiecte și descrierea funcționalității aplicației. Alături de cele menționate anterior, vor fi atașate fișierele necesare pentru a folosi aplicația propriu-zisă, dar și un manual de utilizare al acesteia.

**Bibliografie**

* <https://stackoverflow.com/questions/74102311/escape-room-is-there-an-alternative-for-if-statements-so-you-can-print-somethin>
* <https://www.geeksforgeeks.org/puzzle-haunted-house-escape/>
* <https://chat.openai.com/chat>