Documentație Proiect: Joc Labirint Educativ

# Obiectivele proiectului

Proiectul are ca scop crearea unui joc educativ de tip labirint care combină elemente de navigare 3D cu provocări sub formă de întrebări și răspunsuri. Obiectivele principale sunt:

1. Crearea unui mediu de joc 3D sub formă de labirint, navigabil de către utilizator.
2. Implementarea unui sistem de întrebări și răspunsuri care să testeze cunoștințele jucătorului.
3. Dezvoltarea unei interfețe grafice intuitive, incluzând un meniu principal, meniu de pauză (pause.h) și setări.
4. Integrarea de elemente vizuale și auditive (sound.h) pentru o experiență de joc captivantă.
5. Asigurarea unei structuri modulare a codului pentru mentenanță și extindere facilă.
6. Utilizarea bibliotecii Raylib pentru randare grafică și gestionarea input-ului.
7. Posibilitatea de compilare și rulare pe multiple platforme, așa cum este configurat în Makefile.

# Analiza / documentare („state of the art”)

Pentru realizarea acestui proiect, s-au analizat și utilizat următoarele tehnologii și concepte:

1. **Limbajul de programare C**: Ales pentru performanță și controlul granular asupra resurselor sistemului.

2. **Biblioteca** [**raylib**](https://www.raylib.com/): O bibliotecă simplă și ușor de utilizat pentru dezvoltarea jocurilor, scrisă în C. Oferă funcționalități pentru grafică 2D/3D, input, audio și multe altele. Calea către raylib este configurată în Makefile (de ex. **RAYLIB\_PATH ?= C:/raylib/raylib**).

3. **Biblioteca** [**raygui**](https://github.com/raysan5/raygui): O extensie pentru raylib, accesibilă tot pe GitHub, a fost utilizată pentru crearea elementelor de interfață grafică (GUI). Este inclusă prin **#include "raygui.h**" în fișiere precum **renderer.c**.

4. Generarea procedurală a labirinturilor: Algoritmi specifici pentru crearea dinamică a structurii labirintului, implementați în **maze.c**.

5. Sistemul de întrebări și răspunsuri: Logica pentru încărcarea întrebărilor (din **Questions.txt**) și a răspunsurilor (din **Answers.txt**), validarea acestora și interacțiunea cu jucătorul, implementată în **questions.c**.

6. Manageriarea stării jocului: Implementarea diverselor stări ale jocului (meniu principal, joc activ, pauză, setări, fereastra cu o întrebare aleatorie) folosind un sistem de manageriere al stării (**gameState.h**).

# Justificarea soluției

Alegerea tehnologiilor și a arhitecturii s-a bazat pe următoarele considerente:

1. **raylib** și **C**: Combinația oferă un echilibru bun între rudimentaritate, performanță și portabilitate. raylib este recunoscut pentru ușurința în utilizare, cât și pentru faptul că este potrivită pentru proiectarea jocurilor de dimensiuni mici și medii, indiferent de scopul de divertisment sau educațional.

2. **raygui**: Pentru interfața grafică, **raygui.h** a fost aleasă datorită integrării native cu raylib și funcționalitatea intuitivă pentru dezvoltarea de elemente de UI standard.

3. **Modularitate**: Proiectul este structurat în multiple fișiere sursă (cu extensia **.c**)și header (**camera.h**, **game.h**, **maze.h**, **pause.h**, **questions.h**, **renderer.h**, **settings.h**, **sound.h**, **title.h etc.**), fiecare pereche de fișier sursă + fișier header implementând funcționalitatea unei anumite componente din joc.M Modularizarea facilitează dezvoltarea, depanarea și testarea jocului.

4. **Resurse externe**: Întrebările și răspunsurile sunt stocate în fișiere text externe (**resources/Q\_A/Questions.txt**, **resources/Q\_A/Answers.txt**), permițând modificarea ușoară a conținutului educațional fără recompilarea codului. Similar, resursele grafice și audio sunt încărcate din directorul **resources**. **Makefile**-ul include directiva **--preload-file resources** pentru compilările jocului ce sunt menite să ruleze în tehnologii web, asigurând disponibilitatea acestor resurse.

5. **Sistem de compilare**: **Makefile** este configurat pentru a face posibilă compilarea pe diferite platforme (Desktop, Web, Raspberry Pi), demonstrând flexibilitatea soluției.

# Descrierea implementării

## Structura generală a proiectului

Proiectul este organizat într-o serie de module, fiecare cu responsabilități specifice:

* **main.c**: Punctul de intrare al aplicației, inițializează jocul și rulează bucla principală.
* **game.c** / **game.h**: Conține logica centrală a jocului, inclusiv interacțiunile jucătorului cu labirintul și sistemul de întrebări.
* **renderer.c** / **renderer.h**: Responsabil pentru toate operațiunile de desenare pe ecran. Inițializează fereastra (**InitializeWindow**), desenează scena 3D, elementele 2D ale interfeței (ex. mini-harta prin **DrawMinimap**, bara de stamină prin **DrawStaminaBar**) și gestionează camera.
* **camera.c** / **camera.h**: Gestionează camera 3D, perspectiva și mișcările acesteia.
* **maze.c** / **maze.h**: Implementează algoritmii de generare și ilustrare a labirintului inspirați din această [implementare](https://www.youtube.com/watch?v=zbXKcDVV4G0):

## Algoritmul de generare al unui labirint aleator

Algoritmul are două reguli/condiții cheie:

1. Fiecare nod trebuie să indice exact un nod vecin, cu excepția unui singur nod „de origine” care nu indică nicăieri. Aceasta înseamnă că va exista întotdeauna o conexiune mai puțin decât numărul de noduri.
2. Fiecare nod trebuie să fie conectat la nodul de origine - adică, dacă urmați calea de la orice nod călătorind de-a lungul săgeților, ar trebui să ajungeți în cele din urmă la nodul de origine.

Aceste reguli asigură:

1. Nu pot exista zone izolate în labirint
2. Nu se pot forma bucle (deoarece nodurile indică unul către celălalt într-un ciclu)
3. Labirintul rămâne „perfect” (complet conectat fără bucle) după fiecare iterație

Algoritmul funcționează prin:

1. Mutarea nodului de origine într-o nouă locație
2. Eliminarea indicatorului său
3. Indicarea originii anterioare către noua origine
4. caracteristică unică a acestui algoritm este că labirintul rămâne „perfect” după fiecare iterație, permițându-i să fie jucabil în timp ce încă se generează. Acest lucru îl face util în special pentru jocurile în care doriți un labirint care se poate schimba și schimba în jurul jucătorului.

Algoritmul a fost conceput special pentru Minecraft, dar poate fi utilizat și în alte contexte în care doriți un labirint generat aleator sau un labirint care se modifică dinamic, menținându-și validitatea pe tot parcursul procesului de generare.

* **questions.c** / **questions.h**: Gestionează încărcarea întrebărilor și răspunsurilor din fișiere (LoadQuestions), afișarea acestora și validarea inputului utilizatorului.
* **pause.c** / **pause.h**: Implementează funcționalitatea meniului de pauză. Structura **Pause\_menu** definește elementele meniului, iar funcțiile precum **InitializePauseControls**, **UpdatePauseControls** și **DrawPauseControls** gestionează logica și afișarea acestuia.
* **settings.c** / **settings.h**: Implementează meniul de setări.
* **sound.c** / **sound.h**: Responsabil pentru încărcarea și redarea efectelor sonore (ex**. correct\_answer.wav**) și a muzicii de fundal (**Music.wav**).
* **gameState.c** / **gameState.h**: Administrează stările curente ale jocului (ex. în meniu, în joc, în pauză).
* **gameResources.c** / **gameResources.h**: Gestionează încărcarea și eliberarea resurselor jocului (texturi, modele, sunete).
* **title.c** / **title.h**: Probabil gestionează ecranul de titlu sau introducerea jocului.
* **types.h**: Definește tipuri de date comune utilizate în proiect.

## Resurse

Toate resursele externe sunt plasate în directorul **resources**:

* **icon.png**: Iconița aplicației, încărcată în InitializeWindow.
* **Q\_A**: Conține fișierele Questions.txt și Answers.txt pentru sistemul de întrebări.
* **Sound**: Conține fișierele audio pentru muzică și efecte sonore.
* **textures**: Conține texturile utilizate în joc.

Au fost folosite câteva resurse third party, ce au fost alterate folosind programul [Audacity](https://www.audacityteam.org/download/):

* [**Sunetul de pop-up**](https://www.youtube.com/watch?v=4j_Jsw-inEo)
* [**Sunete pentru răspuns corect și răspuns greșit**](https://www.youtube.com/watch?v=AHLCE7iH9qI)

## Sistemul de compilare

**Makefile** este configurat pentru a compila proiectul pentru diverse platforme (**PLATFORM\_DESKTOP**, **PLATFORM\_WEB**, **PLATFORM\_RPI**). Acesta definește variabilele necesare pentru compilator, linker, căile către biblioteci (inclusiv raylib) și opțiunile specifice fiecărei platforme. De exemplu, pentru versiunea web a jocului, se folosește Emscripten și se preîncarcă directorul resources (**CFLAGS += --preload-file resources**).

# Concluzii

Proiectul demonstrează capacitatea de a dezvolta un joc educativ 3D funcțional utilizând limbajul C și biblioteca raylib. Arhitectura modulară, separarea logicii de prezentare și utilizarea resurselor externe contribuie la un cod organizat și ușor de întreținut.

Principalele realizări includ:

* Un mediu 3D navigabil.
* Un sistem interactiv de întrebări și răspunsuri.
* interfață grafică funcțională, incluzând meniu de pauză și alte elemente UI.
* Managementul stărilor jocului și al resurselor.
* Suport pentru sunet și muzică.

Proiectul are o bază robustă și poate fi extins cu noi funcționalități, cum ar fi niveluri de dificultate variate, tematici educaționale diverse sau elemente de gamification mai complexe. Licența (**LICENSE**) specifică termenii de utilizare și distribuție a software-ului.