

Mystery Library Puzzle Game

نظرة عامة

اللاعب ينتقل من كتاب لكتاب، يحل ألغاز، لحد ما يوصل للكتاب النهائي. **Linked Lists** لعبة ألغاز مبنية على

كيف اللعبة بتشتغل؟

1 البداية

- اللاعب بيختار واحدة من 4 بوابات دخول
- كل بوابة عندها كتاب بلغز واحد


2 المسار

- بعد كل كتاب، اللاعب ينتقل للكتاب التالي
- عندها مسارين - اللاعب يختار: (EASY) الكتب السهلة
- عندها مسار واحد بس: (HARD) الكتب الصعبة

3 الألغاز

- كل كتاب عنده ألغاز لازم تتحل
- عندك 3 محاولات لكل لغز
- Game Over = لو فشلت

4 النهاية

- ! لما توصل للكتاب النهائي وتحل اللغز = تفوز

بنية المشروع

Mystery-Library/

```
|
|— BookNode.h      # تعريف الكتاب (Node)
|— BookNode.cpp    # تنفيذ دوال الكتاب
|
|— Utils.h         # دوال مساعدة
|— Utils.cpp       # معالجة النصوص والألغاز
|
```

Library.h	# تعريف المكتبة
Library.cpp	# بناء وربط الكتب
Game.h	# تعريف اللعبة
Game.cpp	# منطق اللعبة
main.cpp	# نقطة البداية
data/	
puzzles.txt	# ملف الألغاز

شرح كل ملف

BookNode (الكتاب)

كل كتاب عنده Linked List في الـ Node هو الـ

```
cpp
struct BookNode {
    int id; // رقم الكتاب
    BookType type; // نوعه (مدخل/وسطي/نهائي)
    BookDifficulty difficulty; // صعوبته (سهل/صعب)

    Clue* clues; // الألغاز
    int clueCount; // عددهم

    BookNode* next1; // المسار الأول
    BookNode* next2; // المسار الثاني (للسهل فقط)

    bool visited; // زرناه؟
    bool puzzleSolved; // حلينا ألغازه؟
};
```

الدوال المهمة:

- `setupClues(count)` - تجهيز مكان للألغاز
- `addClue(index, problem, solution)` - إضافة لغز
- `hasTwoPaths()` - هل عندنا مسارين؟

Utils (الدوال المساعدة)

دوال علشان نسهل الشغل:

```
cpp

// معالجة النصوص
string cleanInput(string text); // تنضيف وتصغير الحروف

// تحميل الألغاز
vector<Puzzle> loadPuzzlesFromFile(string filename);

// أرقام عشوائية
int randomNumber(int min, int max);
```

Library (المكتبة)

المكتبة بتعمل كل الشغل الثقيل:

```
cpp

class Library {
private:
    BookNode* entranceBooks[4]; // كتب مداخل 4
    vector<BookNode*> middleBooks; // كتاب وسطي 10-15
    BookNode* finalBook; // الكتاب النهائي

public:
    Library(vector<Puzzle> puzzles); // بناء المكتبة
    BookNode** getEntranceBooks(); // جلب المداخل
};
```

كيف بتشتغل:

1. `createEntranceBooks()` - تنشئ 4 كتب مداخل
2. `createMiddleBooks()` - تنشئ 10-15 كتاب وسطي
3. `createFinalBook()` - تنشئ الكتاب النهائي
4. `linkAllBooks()` - تربط كل الكتب ببعضها

الربط:

- المداخل → كتب وسطية عشوائية
- الكتب الوسطية → كتب وسطية ثانية أو الكتاب النهائي

- آخر 3 كتب → الكتاب النهائي

Game (اللعبة)

دي اللي بتدير اللعبة

```
cpp

class Game {
private:
    Library* library;
    BookNode* currentBook;

public:
    void start(); // تشغيل اللعبة
};
```

Flow اللعبة:

```
showIntro()      // عرض المقدمة
↓
chooseEntrance() // اختيار بوابة
↓
while (currentBook != null) {
    solvePuzzles() // حل الألغاز
    ↓
    if (FINAL)     // وصلنا للنهاية؟
        WIN!
    else
        chooseNextPath() // نكمل
}
```

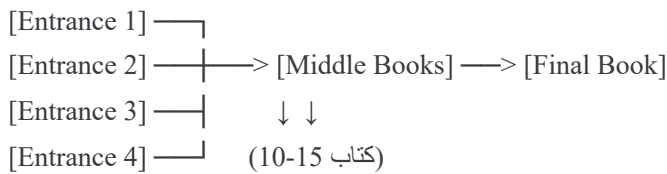
ملف الألغاز (puzzles.txt)

الشكل:

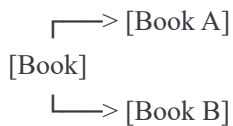
```
[Category Name]
Riddle: What has keys but no locks? | Answer: keyboard
Riddle: I have cities but no houses. What am I? | Answer: map

[Math]
Riddle: What is 2 + 2? | Answer: 4
```

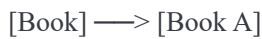
Linked List Structure



الكتب السهلة (EASY):



الكتب الصعبة (HARD):



نصائح للفهم

1. main.cpp ابدأ من

- الأساسي flow شوف الـ
- تحميل أَلغاز → بناء مكتبة → تشغيل لعبة

2. افهم BookNode

- الأساسي building block هو الـ
- Node = كل كتاب في Linked List

3. Library::linkAllBooks() شوف

- هنا بنربط الكتب ببعضها
- □ هنا السحر بيحصل

4. Game::start() اتبع

- هنا اللعبة الفعلية
- بسيط: حل → انتقل → حل → انتقل Loop

Compilation

```
bash

g++ -o game main.cpp BookNode.cpp Library.cpp Game.cpp Utils.cpp
./game
```

Makefile: أو باستخدام

```
bash

make

./game
```

مفاهيم مهمة تعلمتها

Linked Lists

- كل كتاب عنده مؤشرات للكتب التالية
- links هم الـ `next1` و `next2`

Dynamic Memory

- استخدام `new` و `delete`
- Constructor و Destructor إدارة الذاكرة في

Pointers

- مؤشر لكتاب = `BookNode*`
- مؤشر لمصفوفة مؤشرات = `BookNode**`

Enums

- `BookType` و `BookDifficulty`
- strings بدل استخدام

Structs

- `BookNode` و `Clue` و `Puzzle`

- تنظيم البيانات
-

Common Issues

مشكلة: "Can't open puzzles.txt"

الحل: تأكد إن الملف موجود في `data/puzzles.txt`

مشكلة: Memory Leaks

بيمسح كل الكتب تلقائياً `((~Library()))` الحل: المدمر

مشكلة: Invalid input

في كل مكان `cin.fail()` checks الحل: في

تحسينات ممكنة

- ☐ Hints إضافة نظام
 - ☐ Timer للألغاز
 - ☐ حفظ التقدم
 - ☐ Difficulty levels مختلفة
 - ☐ Sound effects
 - ☐ ASCII art رسومات
-

الخلاصة

المشروع ده عبارة عن:

1. **Linked List** من الكتب
2. كل كتاب عنده ألغاز
3. اللاعب ينتقل من كتاب لكتاب
4. يحل الألغاز
5. لحد ما يفوز أو يخسر

عندها تعليقات توضح إيه اللي بيحصل function الكود مكتوب بشكل بسيط وواضح للمبتدئين. كل

Good Luck! 

