📚 Mystery Library Puzzle Game

نظرة عامة

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

🎮 كيف اللعبة بتشتغل؟

1️⃣ البداية

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

2️⃣ المسار

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

3️⃣ الألغاز

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

4️⃣ 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)

هو الـ **Node** في الـ Linked List. كل كتاب عنده:

```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 تانية 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

```
[Entrance 1] ──┐
[Entrance 2] ──┼──> [Middle Books] ──> [Final Book]
[Entrance 3] ──┤       ↓ ↓
[Entrance 4] ──┘     (10-15 كتاب)
```

الكتب السهلة (EASY):

```
    ┌──> [Book A]
[Book]
    └──> [Book B]
```

الكتب الصعبة (HARD):

```
[Book] ──> [Book A]
```

💡 نصائح للفهم

1. ابدأ من main.cpp

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

2. افهم BookNode

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

3. شوف Library::linkAllBooks()

- هنا بنربط الكتب ببعضها
- هنا السحر بيحصل □

4. اتبع Game::start()

- 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
```

أو باستخدام Makefile:

```bash
make
./game
```

🎓 مفاهيم مهمة تعلمتها

✅ Linked Lists

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

✅ Dynamic Memory

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

✅ Pointers

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

✅ Enums

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

✅ 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

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

مشكلة: Invalid input

الحل: في كل مكان checks `cin.fail()`

🎨 تحسينات ممكنة

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

📞 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**

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

Good Luck! 🎉