شرح تفصيلي لملفات Game.cpp و Game.h

نظرة عامة

هذه الملفات تمثل محرك اللعبة الرئيسي في لعبة **Mystery Library**، حيث يتنقل اللاعب بين الكتب ويحل الألغاز للوصول إلى الكتاب النهائي والهروب من المكتبة.

ملف Game.h (ملف الواجهة)

تعريف الفئة

```cpp
class Game {
private:
    Library* library;       // مؤشر للمكتبة التي تحتوي على كل الكتب
    BookNode* currentBook;  // مؤشر للكتاب الحالي الذي يتفاعل معه اللاعب
```

الدوال الخاصة (Private Functions)

1. `void showIntro()`

- **الوظيفة**: عرض شاشة الترحيب والمقدمة للعبة
- **متى تُستخدم**: في بداية اللعبة مرة واحدة فقط

2. `void chooseEntrance()`

- **الوظيفة**: السماح للاعب باختيار بوابة الدخول من 4 بوابات
- **التفاصيل**: كل بوابة لها مستوى صعوبة مختلف (EASY أو HARD)

3. `bool solvePuzzles()`

- **الوظيفة**: عرض ألغاز الكتاب الحالي وإعطاء اللاعب فرصة لحلها
- **القيمة المُرجعة**:
  - `true` إذا حل اللاعب جميع الألغاز بنجاح
  - `false` إذا فشل في حل أي لغز

4. `void chooseNextPath()`

- **الوظيفة**: الانتقال للكتاب التالي
- **التفاصيل**: إما مسار واحد (EASY) أو مساران للاختيار (HARD)

5. `bool checkAnswer(string playerAnswer, string correctAnswer)`

- **الوظيفة**: التحقق من صحة إجابة اللاعب
- **الآلية**: تنظيف النصوص ومقارنتها (غير حساسة لحالة الأحرف أو المسافات)

الدوال العامة (Public Functions)

1. `Game(Library* lib)`

- **البناء (Constructor)**: ربط اللعبة بمكتبة معينة وتهيئة المتغيرات

2. `void start()`

- **الوظيفة الرئيسية**: بدء اللعبة وإدارة دورة اللعب الكاملة

ملف Game.cpp (التنفيذ)

1. البناء (Constructor)

```cpp
Game::Game(Library* lib) {
    library = lib;          // حفظ مرجع المكتبة
    currentBook = nullptr;  // لم يتم اختيار كتاب بعد
}
```

---

2. دالة showIntro()

```cpp
void Game::showIntro() {
    // طباعة رسالة ترحيب جميلة مع إطار
    cout << "WELCOME TO THE MYSTERY LIBRARY" << endl;
    // شرح الهدف: حل الألغاز والهروب
}
```

الهدف: خلق جو تشويقي وإخبار اللاعب بهدف اللعبة.

---

3. دالة chooseEntrance()

أ. الحصول على البوابات

```cpp
BookNode** entrances = library->getEntranceBooks();
// يعيد مصفوفة من 4 مؤشرات لكتب البداية
```

ب. عرض الخيارات

```cpp
for (int i = 0; i < 4; i++) {
    cout << "Gate " << (i + 1);
    if (entrances[i]->difficulty == EASY) {
        cout << "EASY (You'll have 2 choices later)";
    } else {
        cout << "HARD (Only 1 path forward)";
    }
}
```

- **بوابة EASY**: ستوفر خيارين في كل مرحلة (مسار أيسر وأيمن)
- **بوابة HARD**: مسار واحد فقط (لا خيارات)

ج. التحقق من الإدخال

```cpp
while (!validChoice) {
    cin >> choice;

    if (cin.fail()) {
        // التعامل مع إدخال غير صحيح (مثل حروف بدلاً من أرقام)
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    else if (choice < 1 || choice > 4) {
        // رقم خارج النطاق
    }
    else {
        validChoice = true;  // الاختيار صحيح
    }
}
```

ملاحظة هامة: استخدام `cin.fail()` للكشف عن الأخطاء و `cin.clear()` لإعادة تعيين حالة الإدخال.

د. تعيين الكتاب الحالي

```cpp
currentBook = entrances[choice - 1];  // choice 3-0 والمصفوفة من 4-1، من
```

---

4. دالة solvePuzzles()

أ. عرض معلومات الكتاب

```cpp
cout << "Book Type: " << currentBook->getTypeString() << endl;
cout << "Difficulty: " << currentBook->getDifficultyString() << endl;
```

ب. حالة خاصة: لا يوجد ألغاز

```cpp
if (puzzleCount == 0) {
    currentBook->puzzleSolved = true;
    currentBook->visited = true;
    return true;
}
```

ج. حلقة الألغاز

```cpp
```

```
for (int i = 0; i < puzzleCount; i++) {
    Clue puzzle = currentBook->getClue(i);

    // محاولات لكل لغز 3
    for (int attempt = 1; attempt <= 3; attempt++) {
        cout << "Riddle: " << puzzle.problem << endl;

        string playerAnswer;
        getline(cin, playerAnswer);  // قراءة سطر كامل (قد يحتوي على مسافات)

        if (checkAnswer(playerAnswer, puzzle.solution)) {
            // إجابة صحيحة - ننتقل للغز التالي
            solved = true;
            break;
        }
        else if (attempt < 3) {
            // إجابة خاطئة - محاولة أخرى
            cout << "WRONG! Try again..." << endl;
        }
        else {
            // نفذت المحاولات - نهاية اللعبة
            cout << "GAME OVER" << endl;
            return false;
        }
    }
}
```

د. النجاح في حل جميع الألغاز

```
cpp

currentBook->puzzleSolved = true;  // تمييز الكتاب كمحلول
currentBook->visited = true;       // تمييز الكتاب كمزار
return true;
```

---

5. دالة checkAnswer()

```
cpp

bool Game::checkAnswer(string playerAnswer, string correctAnswer) {
    return cleanInput(playerAnswer) == cleanInput(correctAnswer);
}
```

- تستخدم دالة `cleanInput()` من ملف Utils للتنظيف

- تزيل المسافات الزائدة وتوحد حالة الأحرف
- مثال: "The Book" و "the book" و " THE BOOK " كلها متساوية

---

6. دالة chooseNextPath()

أ. التحقق من الوصول للنهاية

```cpp
if (currentBook->type == FINAL) {
    return;  // لا داعي للانتقال - اللعبة انتهت
}
```

ب. مسار واحد

```cpp
if (!hasTwoPaths()) {
    cout << "Following the only path forward..." << endl;
    currentBook = currentBook->next1;
}
```

ج. مساران

```cpp
else {
    cout << "[1] Take the LEFT path" << endl;
    cout << "[2] Take the RIGHT path" << endl;

    // قراءة اختيار اللاعب (مع التحقق)

    if (choice == 1) {
        currentBook = currentBook->next1;  // المسار الأيسر
    }
    else {
        currentBook = currentBook->next2;  // المسار الأيمن
    }
}
```

---

7. دالة start() - القلب النابض للعبة

المراحل الرئيسية:

```cpp
void Game::start() {
    // 1. عرض المقدمة
    showIntro();

    // 2. اختيار بوابة البداية
    chooseEntrance();

    // 3. الحلقة الرئيسية للعبة
    while (currentBook != nullptr) {

        // 3.1. حل ألغاز الكتاب الحالي
        bool puzzlesSolved = solvePuzzles();

        // 3.2. التحقق من النجاح
        if (!puzzlesSolved) {
            // فشل - نهاية اللعبة
            cout << "GAME OVER" << endl;
            return;
        }

        // 3.3. التحقق من الوصول للنهاية
        if (currentBook->type == FINAL) {
            // فوز!
            cout << "CONGRATULATIONS!" << endl;
            break;
        }

        // 3.4. الانتقال للكتاب التالي
        chooseNextPath();
    }
}
```

تدفق اللعبة (Game Flow)

المسار الناجح:

1. showIntro() → عرض الترحيب
2. chooseEntrance() → اختيار بوابة من 4

3. Loop:
   a. solvePuzzles() → حل ألغاز الكتاب الحالي (3 محاولات لكل لغز)
   b. التحقق من النوع:
      - إذا FINAL → الفوز والخروج
      - وإلا → chooseNextPath()
   c. الانتقال للكتاب التالي
4. End → رسالة الشكر

المسار الفاشل:

1-3. نفس المسار الناجح
4. في solvePuzzles(): فشل في حل لغز بعد 3 محاولات
5. return false → العودة لـ start()
6. رسالة "Game Over"
7. Exit

نقاط تقنية مهمة

1. إدارة الإدخال

```cpp
cin >> choice;          // للأرقام البسيطة
getline(cin, answer);   // للنصوص الطويلة (مع مسافات)
cin.ignore(...);        // لتنظيف المخزن المؤقت
```

2. معالجة الأخطاء

```cpp
if (cin.fail()) {
    cin.clear();                          // مسح حالة الخطأ
    cin.ignore(numeric_limits<streamsize>::max(), '\n'); // تجاهل السطر
}
```

3. المؤشرات

- `library`: مؤشر للمكتبة (لا يتغير)
- `currentBook`: مؤشر يتحرك عبر شجرة الكتب
- `entrances`: مصفوفة مؤشرات مؤقتة

4. الحالات المنطقية (Flags)

- `puzzleSolved`: هل تم حل ألغاز الكتاب؟
- `visited`: هل تمت زيارة الكتاب من قبل؟
- `validChoice`: هل الاختيار صحيح؟

مثال على جلسة لعب

```
==================================================
     WELCOME TO THE MYSTERY LIBRARY
==================================================

You see 4 mysterious entrance gates...
  [1] Gate 1 - EASY (You'll have 2 choices later)
  [2] Gate 2 - HARD (Only 1 path forward)
  [3] Gate 3 - EASY (You'll have 2 choices later)
  [4] Gate 4 - HARD (Only 1 path forward)

>> Choose your gate (1-4): 1

>> You step through Gate 1...

==================================================
 Book Type: ENTRANCE
 Difficulty: EASY
==================================================

This book contains 2 puzzle(s).

+----------------------------------------------+
|  PUZZLE 1 of 2                               |
+----------------------------------------------+

Riddle: What has keys but no locks?
Attempts: 1/3

>> Your Answer: keyboard

>> CORRECT! Well done!

[... اللغز الثاني ...]

==================================================
   >> All puzzles solved! Great job!
==================================================
```

```
================================================
    THE PATH SPLITS IN TWO!
================================================
 [1] Take the LEFT path
 [2] Take the RIGHT path

>> Which path? (1 or 2): 1

[... المزيد من الكتب والألغاز ...]

================================================
        *** CONGRATULATIONS! ***
   You solved the final mystery book!
     The library sets you free!
================================================
```

الخلاصة

ملف **Game** هو القائد الذي ينسق بين:

- **Library**: مصدر البيانات (الكتب والألغاز)
- **BookNode**: العُقد في رحلة اللاعب
- **Utils**: أدوات مساعدة للتنظيف والمعالجة
- **اللاعب**: من خلال واجهة console تفاعلية

التصميم قوي ومرن ويسمح بتوسعات مستقبلية مثل:

- إضافة أنواع جديدة من الألغاز
- نظام نقاط وتصنيف
- حفظ التقدم واستعادته
- مستويات صعوبة متعددة