

1.4 Abschlussübung Spiel "Jage den Wumpus"

Erstelle eine einfache Implementierung des klassischen Textspiels "Jage den Wumpus". Dieses Problem soll hier nachrichtenbasiert und inkrementell gelöst werden. Das Programmgerüst und die nötigsten Nachrichten sind bereits implementiert, um die Anwendung zu starten und durch Eingabe zu beenden. Jedes Inkrement wird eine Nachrichtenbehandlung beinhalten. Die Regeln können weiter unten nachgelesen werden.

1.4.1 Vorgehen

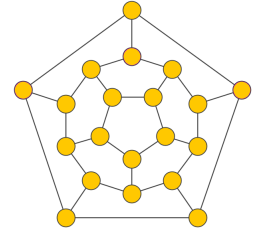
1. Untersuche das bestehende Programm hinsichtlich Struktur(Stil) und Ablauf. Füge dann die Nachricht `InitCaveMsg` und den Handler `InitCaveMsgHandler` hinzu. Implementiere das Initialisieren der Cave (erst einmal ohne Zufall) und ersetze das passende `TODO` in `HuntTheWumpus` durch das Nachrichtenhinzufügen.
2. Füge die Nachricht `RoomEnteredMsg` und den Handler `RoomEnteredMsgHandler` hinzu. Dieser soll den Raum der betreten worden ist ausgeben z.B.:

```
Du bist in Raum 3
  Es raschelt
  Ein kalter Windzug
Tunnel nach 6, 11, 4
```
3. Füge eine neue Nachricht `MoveMsg` und dessen Handler(`MoveMsgHandler`) hinzu. Erweitere den `UserInputMsgHandler`, damit die Nachricht `MoveMsg` in dem richtigen Moment gesendet wird. Implementiere das Bewegen in dem Labyrinth. Was passiert bei ungültiger Raumnummer (`InvalidRoomMsg`)
4. Spare Code und schreibe dir Hilfsklassen (in `MessageHandlerBase.h`) die von `MessageHandler` erbt und die Konstruktoren und Variablen für deine `MessageHandler` bereitstellen (benutze *Inheriting Constructor*).
5. Implementiere den Zufallsgenerator `RandomGenerator`.
6. Implementiere die Nachrichten und Handler für die Fälle die beim Betreten in den Raum mit Fledermaus, Gruben, Wumpus ausgelöst werden. Bestimme in `MoveMsgHandler`, welche Nachricht ausgelöst werden soll.
7. Erweitere den `InitCaveMsgHandler`, indem die Raumnummern und Orte der Spezialräume zufällig bestimmt werden.
8. Füge eine neue Nachricht `ShootMsg` und dessen Handler(`ShootMsgHandler`) hinzu. Erweitere den `UserInputMsgHandler`, damit die Nachricht `ShootMsg` in dem richtigen Moment gesendet wird. Implementiere das Schießen in die Nachbarräume und das Aktualisieren der Anzeige der Restpfeile `UpdateArrowLevelMsg...`
9. Implementiere die Nachrichten und Handler für die Fälle die beim Schießen (Wumpus getroffen, Wumpus verfehlt) auftreten können. Bestimme in `ShootMsgHandler`, welche Nachricht ausgelöst werden soll.
10. Teste deine Anwendung vollständig (alle Scenarios die in den Regeln beschrieben sind)! Tipp verwende die `Print(Cave)` Funktion, um zu sehen ob das Aktualisieren des Labyrinths wie gedacht funktioniert.
11. Optional: Füge ein eigenes Feature dem Spiel hinzu. Viel Spaß!

1.4.2 Regeln

Das Ziel des Spielers ist es, das schreckliche Biest Wumpus zu finden und zu töten, das in der Höhle lauert.

Das Spiel spielt in einer Höhle, die aus einem Labyrinth mit 20 Räumen besteht. Jeder Raum ist mit 3 anderen Räumen verbunden. Angeordnet sind die Räume wie ein Dodekaeder.



Der Spieler hat 5 spitze Pfeile. Wenn die Pfeile ausgehen, bevor der Wumpus getötet wurde, ist das Spiel verloren.

In der Höhle gibt es:

- Ein Wumpus
- Zwei Riesenfledermäuse
- Zwei bodenlose Gruben

Betritt der Spieler einen Raum mit dem Wumpus, wird er von ihm mit einer 25% Wahrscheinlichkeit gefressen und verliert das Spiel. Mit 75% verlässt der Wumpus den Raum.

Betritt der Spieler einen Raum mit einer bodenlosen Grube, fällt er hinein und das Spiel ist verloren.

Betritt der Spieler einen Raum mit einer riesigen Fledermaus, nimmt die Fledermaus ihn und transportiert ihn in einen zufälligen Raum.

Immer wenn der Spieler einen Raum betritt, "spürt" er, was in angrenzenden Räumen passiert. Die Nachrichten sind:

- In der Nähe des Wumpus: "Es riecht schrecklich."
- In der Nähe der Fledermaus: "Es raschelt."
- In der Nähe der Grube: "Ein kalter Windzug."

Wenn der Spieler in einen angrenzenden Raum schießt, wo sich der Wumpus befindet, gewinnt er das Spiel. Wenn er in einen anderen Raum schießt, hat der Wumpus eine 75% Chance, aufzuwachen und in einen von ihm angrenzenden Raum zu ziehen. Wenn dies der Raum mit dem Spieler ist, frisst er ihn auf und das Spiel ist verloren.