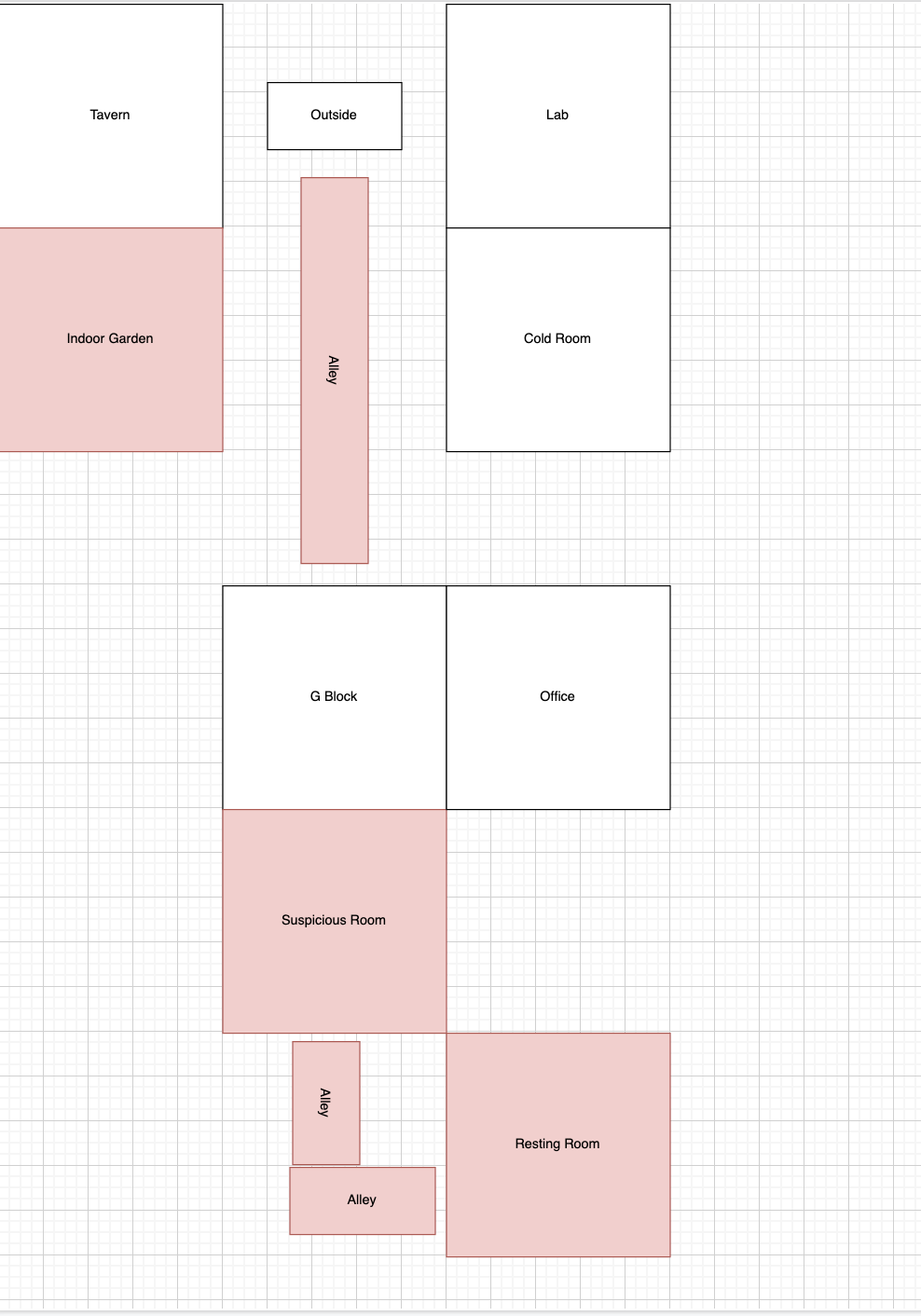
# Zork Game

## Idee:

Unsere Idee war es, das Spiel, um paar Räume zu erweitern. Wir haben einen Gang erstellt, mithilfe des Ganges erreicht man einfach die anderen Räume. Zudem haben wir auch noch drei Räume erstellt, Indoorgarden, Suspicious Room, Resting Room. In ein paar Räumen haben wir Items hinzugefügt, die man mit den commands «get» und «put», aufheben und wieder ablegen kann. Mit dem command «show» kann man das Inventory des Spielers darstellen und die aufgehobenen Items sehen.

## Visualisierung:

 Hier sind alle Räume des Spieles visualisiert.

Die pinken Räume sind die neu erstellen Räume.

## JUnit

Wir hatten beide Schwierigkeiten mit JUnit Tests, Jaspreet hat vor der Schule noch nie mit Tests gearbeitet und selbst ich (Tabea) konnte nicht gut mit JUnit Tests umgehen daher haben wir am Ende Herrn Lanza um Hilfe gebeten. Wir haben die Test Klassen InventoryTest, ItemTest und RoomTest hinzugefügt. In diesen Klassen haben wir getestet, ob man Items in Räume und in das Inventory hinzufügen oder wieder entfernen kann.

## Git Vorgehen

Unser Projekt ist auf der GitHub Repository, unter den Namen «Zork» gespeichert. Wir haben verschiedene branches (main, map, item, inventory) erstellt und auch gemerged. Am Anfang der Lektionen haben wir immer einen Push und Commit durchgeführt. Damit wir sicher gehen können, dass alles auf dem neuen Stand ist und alle Teilnehmer den aktuellen Stand haben. Diesen Vorgang haben wir regelmässig wiederholt. Wir haben das Grundprinzip «GitFlow» geführt.

## Clean Code

Warum Clean Code?

Das Schreiben von Clean Code ist ebenso eine Frage der persönlichen Gewohnheit wie des Könnens. Mit der Zeit wird das Anwenden von Clean Code eine Angewohnheit.

* Entwicklung von Software

Clean Code kann bei der Entwicklung von Software helfen, da jede Software eine produktive Lebensdauer hat und während dieser Zeit Änderungen erfordert.

* Einfachere Fehlerbehebung

Software, die mit Clean Coding-Prinzipien entwickelt wurde, lässt sich leichter auf Probleme beheben. Es kann oft eine schnelle Bearbeitung in Bezug auf «Fixes»/ «Fehlerbehebung» erfordern.

Prinzipien:

**DRY-Don’t Repeat Yourself**

Jede Verdopplung von Code oder Information in einer Software ist schlecht.

**KISS-Keep it simple, stupid**

Halte den Source-Code möglichst einfach, gibt es einen einfachen Weg etwas zu implementieren, sollte dieser Weg begangen werden.

**YAGNI-You ain’t gonna need it**

Es sollte nichts implementiert werden, was vielleicht in der Zukunft nicht verwendet werden kann.

Regeln:

* Auskräftige Name

Der Namen einer Variablen, Funktion oder Klasse sollte sofort erklären, warum sie existiert oder für was sie verwendet wird.

* Verständliche Abkürzungen

Keine unklaren Abkürzungen verwenden.

* Kommentare

Bei komplizierten oder unklaren Stellen, Kommentare hinzufügen. Damit andere es verstehen, oder wenn man es selbst nach einer Weile anschaut auch versteht.

* JUnit Tests

Junit Tests einfügen, um die Software zu testen.

* Keine überflüssigen Informationen

Es werden häufig bestimmte Präfixe in den Klassenamen einfügt, aber versuchen sie diese zu vermeiden. Namen sollten kurz und deutlich sein, nicht unnötigen Kontext einfügen.

* Kleine Funktionen

Die Funktionen sollten klein gehalten werden und nicht länger als 20 Zeilen sein.

# Reflektion

Wir haben die Idee von Herr Cavouti und Herr Lanza sehr gut gefunden. So haben wir mit einem Projekt sehr viele neue verschiedene Sachen gelernt. Die Aufgabenstellung war von Anfang an klar und wir haben auch genug Zeit erhalten.

In der Gruppe hat auch alles gut funktioniert, wir haben uns die Aufgaben zugeteilt und somit Zeit gespart. Aber fürs nächste Mal nehmen wir mit, realistische Ziele setzen und schnellere Entscheidungen treffen. Wir hatten am Anfang sehr viele Ideen und konnten uns nicht entscheiden, deshalb haben wir auch nicht all unsere Ideen umsetzten können. Auch haben wir zu lange selbst studiert bei einigen Problemen, wir haben nicht den Lehrer gefragt, was ein Fehler von uns war. Aber in der Zukunft werden mit einem noch genaueren Plan vorgehen.