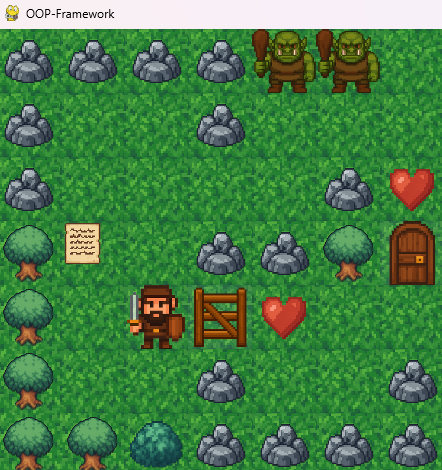
**Exkurs: Benutzer- vs. Entwickleransicht**

Als **Benutzer** eines Programms (z. B. Windows, PowerPoint, FireFox, Chrome, Minecraft,…) siehst du nur die **Benutzeroberfläche**, also Figuren, Aktionen und Reaktionen. Du kannst über **Eingaben** (z. B. Tastatur, Maus) etwas bewirken, hast aber **keinen direkten Zugriff** auf den Programmcode oder auf einzelne Daten im Hintergrund.



Wenn du ein Programm **entwickelst**, siehst du den **gesamten Code** und kannst **alles manipulieren**. Zugriffsregeln für Attribute kannst du also einfach umgehen und getter und setter hinzufügen, wie es dir beliebt.

Ein Bild, das Text, Pixel, Display, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Jedes Objekt wird jedoch mit einem **klaren Ziel** konzipiert und implementiert. Ein **unkontrolliertes Eingreifen** in diese Struktur kann leicht zu **Fehlern** und damit zu Abstürzen, ungewolltem Verhalten oder sogar Sicherheitslücken führen. Dies trifft umso mehr zu, wenn du deinen Code **anderen Entwicklern** zur Verfügung stellst und diese ihn weiterverwenden wollen.

Aus diesem Grund werden genau die getter und setter (und weitere Methoden) zur Verfügung gestellt, die benötigt werden, um Objekte wie intendiert zu verwenden. Ergänzen und ändern kann man diese im Nachhinein immer noch. Der Benutzer wird davon nichts merken.

**Beispiel:**

Der Held besitzt die Methode setze\_position(x,y), mit der er teleportiert werden kann. Diese Methode ist z.B. nützlich, wenn in das Spiel Portale eingeführt werden sollen. Insbesondere stellt sie sicher, dass der Held nicht an ungültige Stellen teleportiert werden kann. Ein Benutzer kann nie auf diese Methode zugreifen, wenn dies nicht über die **Benutzeroberfläche** (z.B. durch einen benutzbaren Teleportatationszauber) ermöglicht wird.