## Welche Ziele möchte ich in den drei Stunden erreichen?

Zuerst einmal werde ich versuchen dafür zu sorgen, dass ich die geplante Pausenzeit, die ich mir heute verloren gegangen ist, mit nachhole. Ansonsten ist mein eigentliches Ziel für den Nachmittag im Dungeon Crawl die Kampfparameter der Figuren, um das Attribut Agilität zu erweitern. Diese will folglich nutzen, um die Entscheidung, wer als nächstes dran ist, besser ausbalancieren zu können.

## Wie möchte ich meine Ziele erreichen?

Zunächst erweitere ich die Klasse Figure um dieses Attribut und ziehe das in allen Konstruktoren nach die bedient werden müssen (zur Not erstmal Direktzuweisung).  
Allgemein muss an allen Stellen ergänzt werden, die Kampfattribute enthalten.  
  
Nun will ich den Randomizer nicht nur einfach einen Boolean erzeugen lassen, sondern den Randomizern die jeweiligen Agilitätswerte als Bound geben und je einen Wert erzeugen lassen, welche dann verglichen werden, wodurch entschieden wird wer dran ist. Soll bewirken: Hat der Spieler doppelt so viel Agilität wie der Gegner, soll auch die Wahrscheinlichkeit, dass er zum Zug kommt, doppelt so hoch sein.

## Reflexion: Habe ich meine Ziele erreicht? Und was habe ich gelernt?

Vorab: Ich vermute, ich werde mich wieder in Details verlieren, aber zur Not belasse ich alles beim Nötigsten wie bisher. Mal schauen, wie weit ich komme.

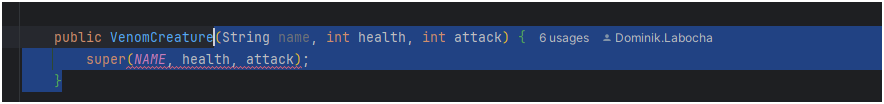
Also meine gewünschten Kriterien für heute habe ich erreicht. Ich habe das Attribut der Agilität eingepflegt und dieses chancenwirksam eingebaut um den booleschen Zufallswert für Heilungserfolg und Spieler ist dran zu ersetzen.

Man muss hierbei natürlich beachten, dass die Werte alle nur pseudozufällig sind und Dinge wie die Systemzeit Einfluss auf das Gambling haben, aber geschafft ist es jetzt.

Interessant fand ich, wie ich einer statischen Methode das aufrufende Objekt mit this übergeben konnte. Das habe ich so nie gemacht, aber irgendwie fiel es mir einfach ein.

Es hat mir Spaß gemacht wieder in diesem Projekt zu arbeiten.   
Ich hoffe, dass ich Freitag noch Zeit dafür haben werde.

Wirf auch gerne einem Blick in mein folgendes Arbeitstagebuch. 😊

Arbeitstagebuch:  
Alle Konstruktoren wurden entweder um die Variable *int agility* erweitert, wenn generisch, oder 50 für Spieler und 25 für Gegner, wenn der vereinfachte Konstruktor verwendet wird. Mir ist aufgefallen, dass alle generischen bei den Gegnern unnötigerweise den Namen in der Parameterliste hatten, obwohl alle Klassen eine Namenskonstante (mit Farbcodes) haben, die als solche an die Super übergeben wird.  


Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Software, Multimedia-Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

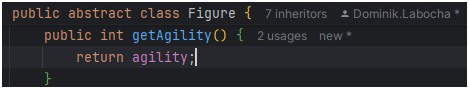
Um den Erfolg empirisch zu testen, habe ich bewusst den Spieler bevorteilt, bei den Schwierigkeiten entspräche diese Parameterzuteilung später aber der Stude leicht.

Mir fällt grad ein, dass nicht nur, die „wer ist dran“ Frage Agilität brauchen könnte, sondern auch die Chance eine Heilung anwenden zu können wir den Spieler.

Aufgrund dessen macht es Sinn das Gambling auszulagern und quasi das Ergebnis, also den Boolean zurückzugeben, wer das Gambling gewonnen hat, Gegner oder Spieler.

Der Einfachheit halber fangen wir aber erstmal in der Kampfmethode an und schauen, dann wie man es so sinnvoll auslagert, dass es in der Klasse Player mitgenutzt wird.

Verflixt, da habe ich doch glatt den Getter für die Agility vergessen.



Mir ist aufgefallen, dass ich eine Boolean playersTurn habe und eine void-Methode playersTurn. Das funktioniert zwar wegen unterschiedlicher Datentypen, aber kann verwirrend werden, spätestens, wenn die Kampfmethode für den Spieler einen Boolean hätte liefern sollen. Ich nenne die Methode nun DoPlayersTurn und die Variable gemäß der Konvention für Booleans (is/has) isPlayersTurn.

Die Implementierung scheint geklappt zu haben.

Ein Bild, das Text, Software, Schrift, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Nun gut, am Ende hat alles zwei Zeilen mehr gebraucht, lohnt sich das Auslagern?  
Ich würde sagen der Modularität und Übung wegen mache ich es einfach.

Der Aufruf im Gameplay wirkt dadurch zwar etwas redundant, aber so kann ich die statische Methode auch in heal des Players aufrufen. Da zum Angreifen eh die Figure bzw. der Gegner auch an den Spieler übergeben wird, ist es kein Problem.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Nun das Ganze für die Heilung.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Das wirkt jetzt etwas verwirrend, aber um die Heilungschance im Vergleich mit der Schnelligkeit des Gegners zu bestimmen, habe ich mich analog nach den anderen Methoden gerichtet. Nur als Enemy, damit ich von Figure nicht downcasten muss.  
(Warum geht nicht figure? Weil ich die Gambling-Methode mit enemy habe).

Übrigens fällt mir grade ein, ich könnte selbiges Prinzip auch mit Genauigkeit (accuracy) für die Treffsicherheit von Attacken oder für kritische Treffer und Spezialangriffe (critCh).

Ich habe mal die Abfrage ob schon full Life nach oben gesetzt, ist womöglich performanter, falls der Fall eintrifft. (Else des Rests nicht nötig, wegen Returns).

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Hier eine schöne neue Verwendung des this-Keywords. Auch wenn der Datenaustausch vermutlich etwas Ping-Pong-artig ist, aber so bleibt der Code modular.  
(Stellt sich jetzt natürlich die Frage, ob 2 Zeilen Code den Aufwand wert sind, aber *DRY*).

Hoppla, ich habe die Negation beim Heal vergessen. Abbruch wenn Gegner schneller.  
Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

So müsste es passen:

Ein Bild, das Text, Multimedia-Software, Software, Screenshot enthält.

Automatisch generierte Beschreibung

Jup läuft.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung