## Nice to know:

Alle Bewegungsanimationen in welchen sich die Charaktere bewegen/ihre Position verändern werden bewerkstelligt, indem die Margin verändert wird.

Alle Animationen (Laufen, attackieren, stehen usw.) werden durchgehend ausgeführt. Da man nicht alle Animationen gleichzeitig sehen sollte, wird bei der gewünschten Animation die Visibility auf Visible geändert und bei den nicht benötigten auf Hidden.

Die Bewegungsanimationen erscheinen deshalb so flüssig, weil jede 10tel Sekunde die Margin leicht verändert wird. Dies geschieht mit: await Task.Delay(«Wert in Ms»);.

Die Applikation verfügt nur über ein Fenster, namens Start. Von diesem werden die verschiedenen Pages aufgezeigt. Nach jedem Fensterwechsel wird die aktuelle Seite resettet.

# Klassen:

## -Enemy

In dieser Klasse wird immer nur eine Instanz von Enemy generiert. Die Werte von Enemy kann man mit z.B. Enemy.getInstance().Health --; verändern. Diese Klasse dient als «Zwischenspeicher» von der Tabelle Enemy der Datenbank, sodass man nicht jedesmal wenn man dem Gegner Schaden hinzufügen möchte, die Datenbank updaten muss.

## - Levelpage

Diese Page ist das Level in welchem die Fragen gestellt und beantwortet werden. Die «Animation» wird dargestellt indem die Visibility der gewünschten Animation auf Visible gestellt wird. Nach einem Klick auf die Pfeiltasten bewegt sich der Charakter. Dies geschieht, indem die Margin verändert wird. In jedem Level kommen 3 Gegner. Diese werden mit der Methode ShowEnemy «gespawnt». Jetzt wird ein Delay von 1ner Sekunde gemacht und anschliessend eine Frage mit der Methode ShowQuestion angezeigt. Wenn der Timer ausläuft oder die Frage falsch beantwortet wird, attackiert der Gegner den Charakter, wenn richtig wird der Gegner angegriffen. Wenn der Gegner noch lebt, kann man sich nicht bewegen.

## - Music

Diese Klasse dient dazu, Musik zu laden, zu spielen und zu pausieren.

## - Player

In dieser Klasse wird immer nur eine Instanz von Player generiert. Die Werte vom Player kann man mit z.B. Player.getInstance().Health --; verändern. Diese Klasse dient als «Zwischenspeicher» von der Tabelle Player der Datenbank, sodass man nicht jedes Mal wenn man Schaden bekommt, die Datenbank updaten muss.

## - Sound

Diese Klasse dient dazu, Sounds zu laden, zu spielen und zu pausieren.

## - Start

Das ist die Startklasse, welche als erstes geöffnet wird. In dieser wird ein Dictionary vom Typ Page erstellt in welchem jede Page beim initialisieren geaddet wird. Nach dem adden der Pages ins Dictionary wird standardmässig auf die Startpage navigiert. Beim Navigieren auf andere Pages wird immer die statische Methode navigateTo(destinationpage) ausgeführt, welche auf die mitgegebene Seite navigiert. Alle statischen Methoden in dieser Klasse sind zwar nur wenige Zeilen, welche aber nicht auf der Seite LevelPage oder WorldPage ausgeführt werden könnten.   
Die statische Methode getNavigationService gibt den aktuellen NavigationService zurück welchen man braucht um in der gewünschten Klasse ein Event beim Navigieren auf diese Seite auszulösen.  
Die statische Methode resetPage(pageToReset) setzt die mitgegebene Page zurück, indem es den Datacontext auf null setzt und eine neue Page vom Typ der mitgegebenen Page erstellt. (Wenn pagetoReset == «Startseite», wird eine neue Startpage erstellt usw.)

## - Startpage

Hier kann der Spieler ein neues Spiel beginnen, oder bereits einen vorhandenen Spielstand spielen. Ausserdem hat er auch die Möglichkeit die Credits zu sehen und das Spiel zu beenden.

## - Worldpage

Hier kann der Charakter das Level auswählen welches er spielen möchte (vorausgesetzt er hat bereits 1 Level freigeschaltet). Wenn der Charakter ein Level zum ersten Mal abschliesst, schaltet er das nächste Level frei. Mit den Pfeiltasten kann man sich zwischen den Levels navigieren. Dabei kann man nur die Levels auswählen welche man bereits freigeschalten hat.