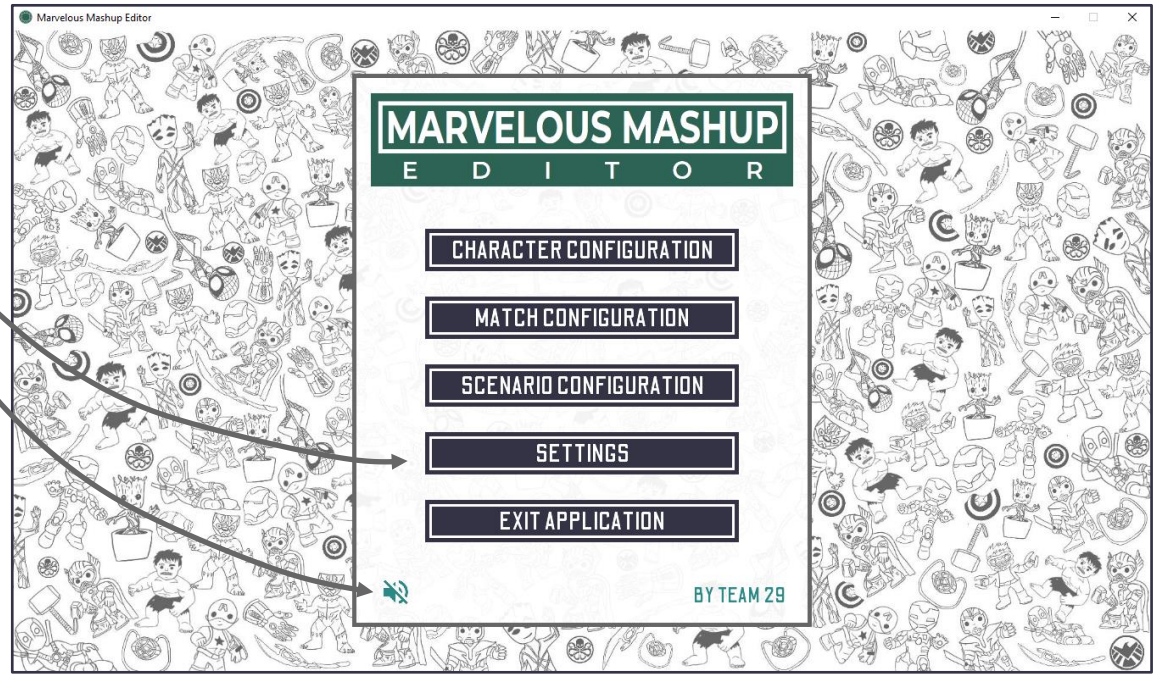
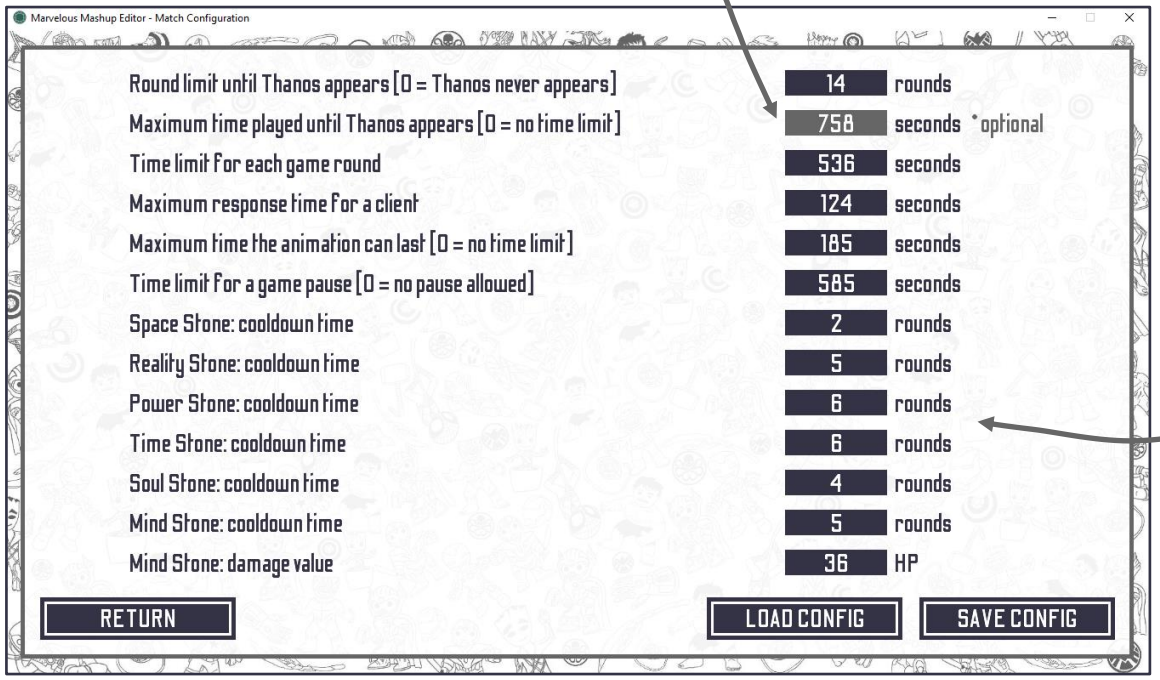
Der Ton kann direkt auf dem Startbildschirm stumm geschalten werden. Zusätzlich können die Lautstärke der Musik und der Sound Effekte im Settings-Screen individuell angepasst werden.



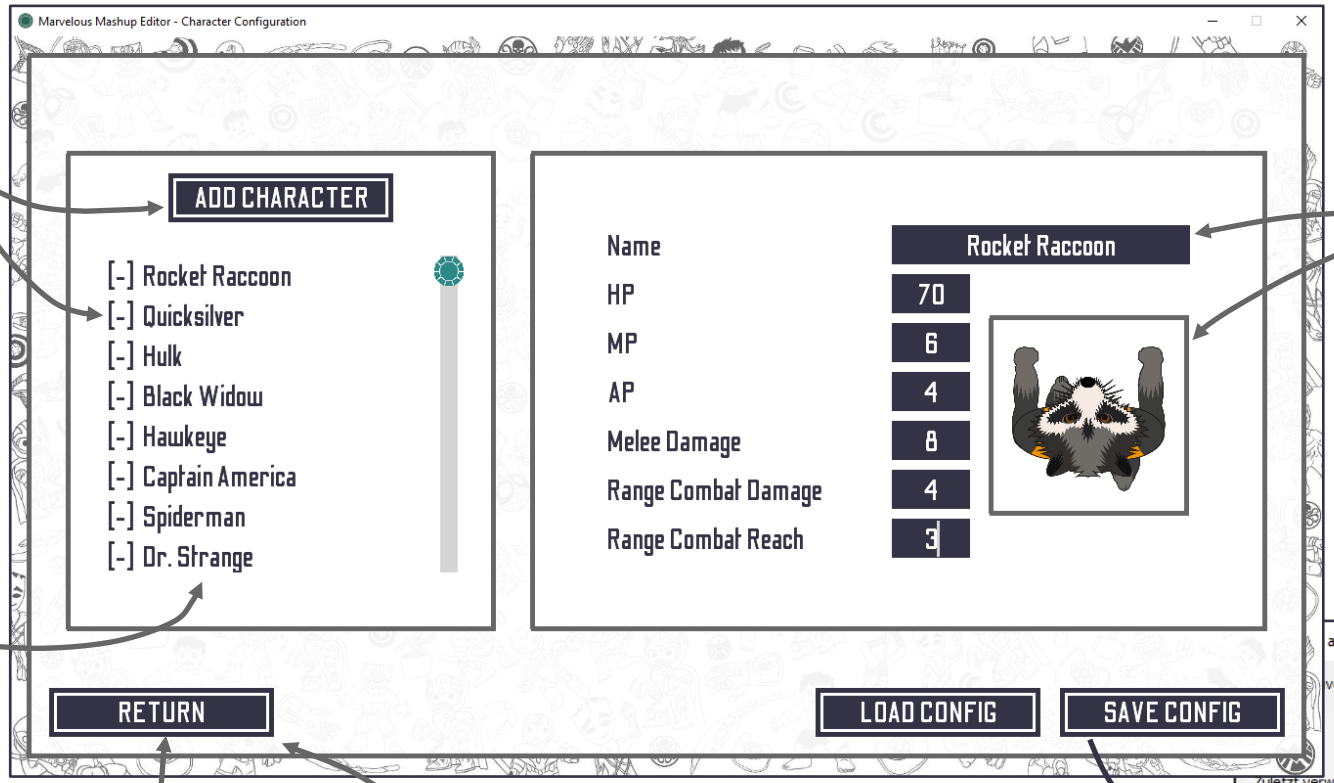
Optionale Attributs-Felder können einfach leer gelassen werden. „0“ wird auch hier ganz regulär als Eingabe gewertet.



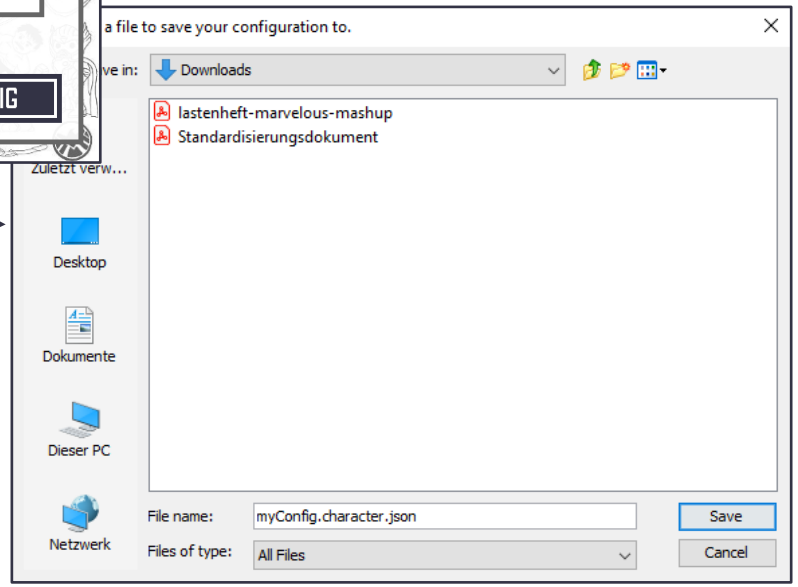
In all unseren Textfeldern sind Filter eingebaut. Auf diese Weise können unsinnige Eingaben erst gar nicht getätigt werden. Hier können zum Beispiel nur die Ziffern 0-9 eingegeben werden und nur maximal 5 davon (andere Tastendrücke werden einfach nicht registriert).

Charaktere können nach belieben zur Liste hinzugefügt und wieder entfernt werden.

Die Namen in der Liste ändern sich automatisch, wenn der Name rechts angepasst wird.



Sobald der angegebene Name einem bekannten Character entspricht, wird direkt das passende Bild geladen und die richtige Charakter ID gesetzt. Sobald der Name wieder geändert wird, werden diese Informationen automatisch zurückgesetzt.



Falls Änderungen an einer Konfiguration vorgenommen wurden, erscheint vor dem Screen-Wechsel noch ein Pop-Up zum Bestätigen der Aktion.

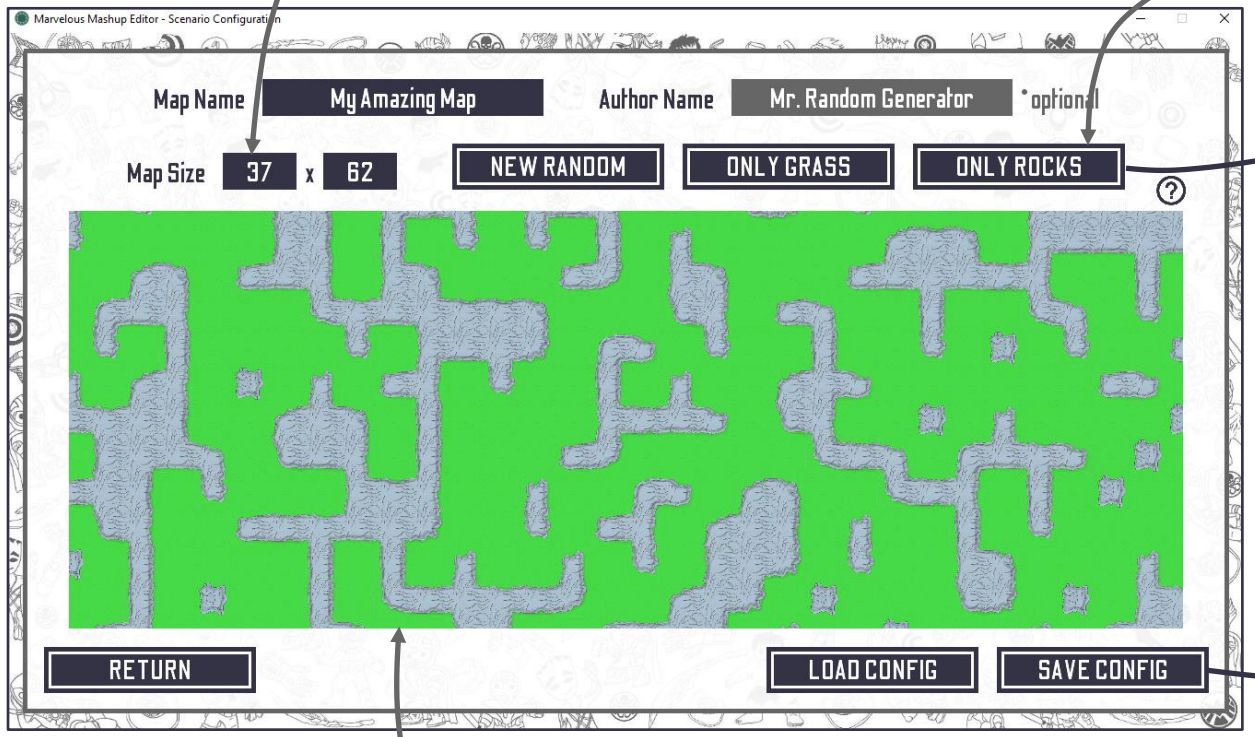
Beim Betreten einer Konfigurations-Ansicht wird direkt eine zufällige, aber immer nach Netzwerkstandard valide und bestmöglich ausgeglichene, Konfiguration geladen.

SZENARIO-KONFIGURATION

TEAM 29

Durch einen Druck auf Enter wird die Map-Größe direkt angepasst.

Dem Nutzer stehen verschiede Möglichkeiten zur Verfügung, wie ein Szenario generiert werden soll. Dies erleichtert die Map-Erstellung erheblich, da so auch für die Erstellung von sehr dünn oder sehr dicht besetzten Maps kaum Klicks notwendig sind.



Left click to place and delete stones.
Right click and hold to move the map.
Scroll the mouse wheel to zoom.

Unsere Felstexturen bilden Formationen, sobald Felsen nebeneinander platziert werden.

Noch bevor der User ein File zum Speichern der Konfiguration auswählen muss, wird geprüft, ob die offene Konfiguration valide ist.

Validierung:
Sämtliche Konfigurationen, die geladen oder gespeichert werden sollen, werden nicht nur gegen das offizielle Schema validiert, sondern auch durch unsere Logik. Dies ermöglicht sehr detaillierte Fehlermeldungen für den Nutzer und außerdem das Abfangen unsinniger Werte.

Skin:
Wir nutzen für den Editor unseren selbsterstellten LibGdx Skin. Es steht euch frei diesen nach Belieben (evtl. auch für euren Client) zu nutzen und anzupassen. Der Skin verfügt über mehrere Abstraktionsebenen, weshalb seine Elemente zumindest aus unserer Sicht nun sehr intuitiv nutzbar und konfigurierbar sind [siehe Grafik rechts].

```
public static TextButton configureButton(String text) {
    return configureButton(text, ComponentsSizeEnum.COMPONENT_WIDTH);
}

public static TextButton configureButton(String text, ComponentsSizeEnum width) {
    return configureButton(text, width, ComponentsSizeEnum.COMPONENT_HEIGHT, getColorScheme(),
            UISettings.getStylingDefaults().getPrimaryFontColor(),
            UISettings.getStylingDefaults().getFontStyle(),
            UISettings.getStylingDefaults().getTextSize());
}

public static TextButton configureButton(String text, Color[] scheme) {
    return configureButton(text, ComponentsSizeEnum.COMPONENT_WIDTH,
            ComponentsSizeEnum.COMPONENT_HEIGHT, scheme,
            UISettings.getStylingDefaults().getPrimaryFontColor(),
            UISettings.getStylingDefaults().getFontStyle(),
            UISettings.getStylingDefaults().getTextSize());
}

public static TextButton configureButton(String text, Color font) {
    return configureButton(text, ComponentsSizeEnum.COMPONENT_WIDTH,
            ComponentsSizeEnum.COMPONENT_HEIGHT, getColorScheme(), font,
            UISettings.getStylingDefaults().getFontStyle(),
            UISettings.getStylingDefaults().getTextSize());
}

public static TextButton configureButton(String text, TextSizeEnum size) {
    return configureButton(text, UISettings.getStylingDefaults().getFontStyle(), size);
}
```

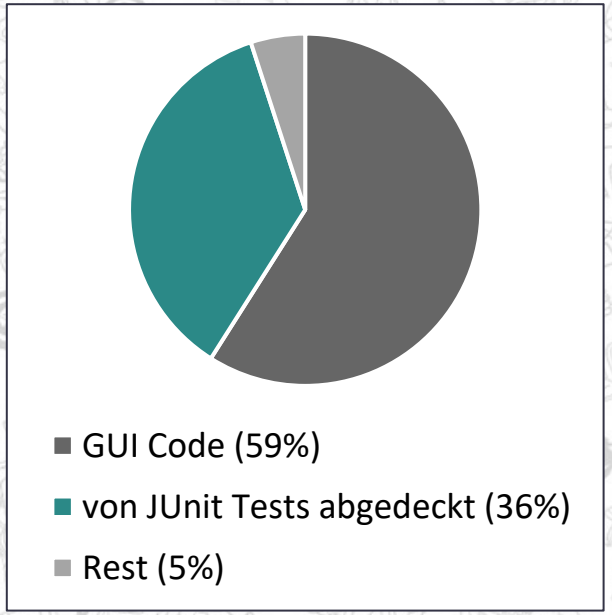
Skalierung:
Unsere Anwendung skaliert immer auf 70% eurer Monitorgröße. Sie bleibt dabei stets im 16:9 Format und orientiert sich im Zweifelsfall für die 70% an der Bildschirmhöhe. Diese Skalierung funktioniert unseren Tests zufolge auf allen Monitoren mit einer Skalierung von maximal 100% - egal ob auf Omas altem 4:3 Rechner, oder eurem neuen 4K Ultra-HD Bildschirm.

Wir haben versucht euch eine möglichst einfach personalisierbare Anwendung zur Verfügung zu stellen: Wichtige Konstanten, wie die Einstellungen des Skins, dessen Farben, das Styling der UI-Komponenten, erlaubte Werte für die Attribute der Konfigurationen [siehe Grafik rechts] oder auch die Einstellungen der Zufalls-Generatoren, müssen nicht irgendwo versteckt im Code abgeändert werden.
Wir haben diese Werte in extra Klassen ausgelagert und diese gut kommentiert, sodass ihr euren Editor leicht selbst konfigurieren könnt.

```
// MATCH
public static final int MIN_VALUE_MAX_ROUNDS = 0;
public static final int MAX_VALUE_MAX_ROUNDS = 500;
public static final int MIN_VALUE_MAX_GAME_TIME = 0;
public static final int MAX_VALUE_MAX_GAME_TIME = Integer.MAX_VALUE;
public static final int MIN_VALUE_MAX_ROUND_TIME = 1;
public static final int MAX_VALUE_MAX_ROUND_TIME = 5000;
public static final int MIN_VALUE_MAX_RESPONSE_TIME = 1;
public static final int MAX_VALUE_MAX_RESPONSE_TIME = 5000;
public static final int MIN_VALUE_MAX_ANIMATION_TIME = 0;
public static final int MAX_VALUE_MAX_ANIMATION_TIME = 5000;
public static final int MIN_VALUE_MAX_PAUSE_TIME = 0;
public static final int MAX_VALUE_MAX_PAUSE_TIME = Integer.MAX_VALUE;
```

Verwendete Technologien:

- Wir nutzen Java 15 mit Gradle 7, JUnit 5 und LibGdx (LWJGL3).
- Außerdem haben wir mit Sonar Lint und Sonar Qube gearbeitet, um euch eine gute Codequalität gewährleisten zu können.
- Alle Klassen und nicht selbsterklärenden Funktionen sind mit JavaDoc kommentiert.



In unserem GitLab-Repository findet ihr noch viele weitere Informationen:

- Die Anwendung als JAR, sodass ihr sie direkt selbst ausprobieren könnt.
- Ein Video, das einen kurzen Rundgang durch den Editor zeigt.
- Ein detaillierter JUnit Test Coverage Report.
- Benutzer- und Anwenderdokumentation inklusive Anleitungen.

Ihr könnt uns gerne jederzeit auf Discord kontaktieren (Maffel#8907) .