

Prüfungsleistung

Modul: Entwicklung Interaktiver Anwendung II

Betreuer: Prof. Jirka Dell'Oro-Friedl

EIA2 Endabgabe

Wintersemester 2020 / 2021

Tag der Einreichung: 16.02.2021

Sebastian Hoffmann, 259316

Funktionale Analyse

Den User erwartet eine Feuerwerkshow der Extraklasse, ganz nach seinen Wünschen angepasst. Er kann hierbei eigene Explosionen konfigurieren, bei dem er die Anzahl der Partikel, deren Größe sowie zwei Einstellungen zur Farbgestaltung zur Auswahl hat. Für eine optimale User Experience muss gewährleistet werden, dass die Anwendung bereits in den ersten Sekunden nach Aufrufen der Seite vom User als interaktiv erkannt wird. Dies kann durch kluge Überschriften, klare Struktur der Input Elemente sowie visuelle Effekte erreicht werden. Jedoch sollte man eine Reizüberflutung vermeiden, um die Anwendung nicht unnötig anstrengend zu gestalten. Heißt, dass alles auf das wesentliche reduziert werden muss. Je einfacher alles funktioniert, desto mehr Spaß macht die Anwendung am Ende auch.

Die Anwendung ist für die Nutzung am PC vorgesehen und optimiert. Das Erlebnis ist schließlich auf großen Monitoren am besten. Eine Smartphone Umsetzung könnte jedoch mit einer einfachen CSS Datei hinzugefügt werden, die sich um ein Responsive Design kümmert.

Was erwartet den Nutzer audiovisuell?

Eine Titelüberschrift, die direkt verrät, was die Anwendung kann.

FEUERWERK CONFIGURATOR

Input Elemente, wie Slider, um Eigenschaften schnell und einfach anzupassen.

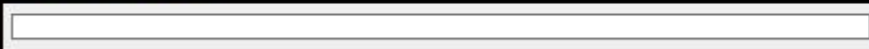
PARTIKELANZAHL



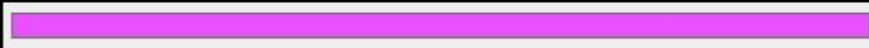
FORM DER PARTIKEL

☒ Kreis ☐ Dreieck ☐ Quadrat ☐ \$GME GO BRRR

Partikelfarbe



Leuchtfarbe



Zusätzlich gibt es noch ein Text-Input Feld, um seine Kreation auf einer Datenbank zu speichern und später wieder zu verwenden. Gespeicherte Raketen werden in einer Liste unter dem Speichern Dialog angezeigt.

Feuerwerk Speichern

Name

Speichern

Feuerwerk Laden

Laden

DreieckRotWeiss ▾

Der canvas ist hinterlegt mit dem Logo des subreddits /rwallstreetbets, welches in jüngster Vergangenheit viel Aufsehen in der Öffentlichkeit erregt hat, mit Hilfe der GameStop Aktie.



Klickt der User dann mit seiner Maus auf den canvas, explodieren die Raketen in einer visuell ansprechenden Weise: Es erscheint ein klarer Kreis an Partikeln, welcher sich langsam abgeschwächt. Zusätzlich sind einige Partikel etwas “unkontrollierter” Unterwegs, um einen etwas realistischeren Eindruck der Explosion zu vermitteln.



Wählt man die Partikelform “\$GME GO BRRR” und lässt die Raketen explodieren, sieht man folgendes.



Als Anlehnung auf das Meme, wurden die Partikel durch eine Bilddatei ersetzt, welches die GME Rakete zeigt. Zusätzlich wird eine Audio abgespielt. Zu hören ist eine Aufnahme des offiziellen r/wallstreetbets Discordservers. Die Leute unterhalten und diskutieren über die neuesten Geschehnisse, und wir bekommen ein exklusiven Einblick in die Denkweise der r/wallstreetbets Community.

Anleitung zur Benutzung der Anwendung

The screenshot shows a web application titled 'FEUERWERK CONFIGURATOR' with a red background. It contains several configuration sections: 'PARTIKELANZAHL' with a slider, 'PARTIKELGRÖSSE' with a slider, 'LEUCHTDAUER' with a slider, 'FORM DER PARTIKEL' with radio buttons for 'Kreis', 'Dreieck', 'Quadrat', and '\$GME GO BRRR', 'Partikelfarbe' with a color picker, 'Leuchtfarbe' with a color picker, 'Feuerwerk Speichern' with a 'Name' input field and a 'Speichern' button, and 'Feuerwerk Laden' with a 'Laden' button and a dropdown menu showing 'DreieckRotWeiss'.

Die ersten drei Slider können bewegt werden um die Partikelanzahl, -größe, sowie Leuchtdauer einzustellen.

Im zweiten Schritt wählt man die gewünschte Partikelform (oder wahlweise das Meme) aus.

Anschließend kann man Kernfarbe und als Zusatz die Leuchtfarbe der einzelnen Partikel bestimmen.

Zuletzt kann man einen Namen für seine Rakete eingeben, und mit dem Speichern-Button an die Datenbank senden.

Darunter sieht man in der Dropdown-Liste bereits gespeicherte Einträge, die man mit dem Laden Button.

Installation der Anwendung

Für die Verwendung der Anwendung werden eine Datenbank und ein Server benötigt. Um diese einzurichten, muss man folgende Schritte befolgen:

1. Heroku

<https://heroku.com> besuchen und Nutzerkonto anlegen oder falls vorhanden, bei seinem bestehenden Account einloggen.

2. Eine neue App erstellen und mit seinem Github Repository verlinken.
3. "package.json" öffnen und in Zeile 7 den Pfad der zur Server.js verweist gegebenenfalls anpassen.
4. "Main.ts" öffnen und in Zeile 4 den Herokulink ersetzen mit dem Link zu seiner eigenen App ersetze. Wichtig: "https://" vor dem Link nicht vergessen!
Code speichern, kompilieren und auf das zu Heroku verlinkte Repository pushen.

5. MongoDB

<https://mongodb.com> besuchen Nutzerkonto anlegen oder falls vorhanden, bei seinem bestehenden Account einloggen.

6. Ein neues Cluster erstellen und in diesem eine Datenbank mit dem Namen "Firework" und eine Collection mit dem Namen "Rockets" erstellen.
7. Die IP Adresse 0.0.0.0/0 im Menü "Network Access" hinzufügen.
8. Einen User erstellen.
9. "Server.ts" öffnen und in Zeile 12 den Verweis zur eigenen Datenbank eingeben. Den Link hierfür finden man unter "Connect" beim Cluster -> Connect your application.
10. An den Stellen <password> und <dbname> die jeweiligen Daten eingeben.
11. Den Code speichern, kompilieren und pushen.

Als letzten Schritt die branch auf der Heroku Seite deployen und die github.io Seite der index.html öffnen und es krachen lassen!

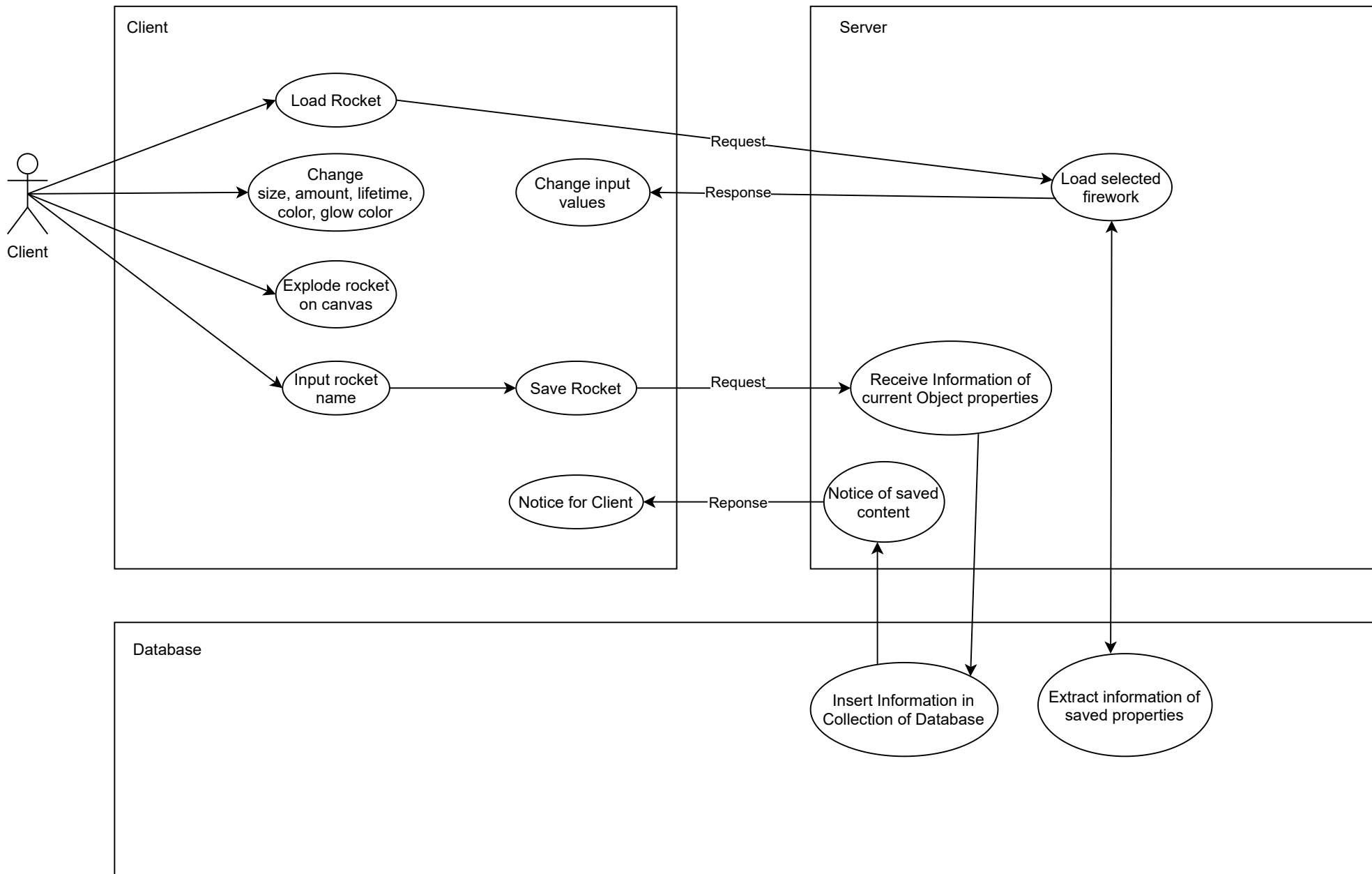
Links

Index.html auf Github pages:

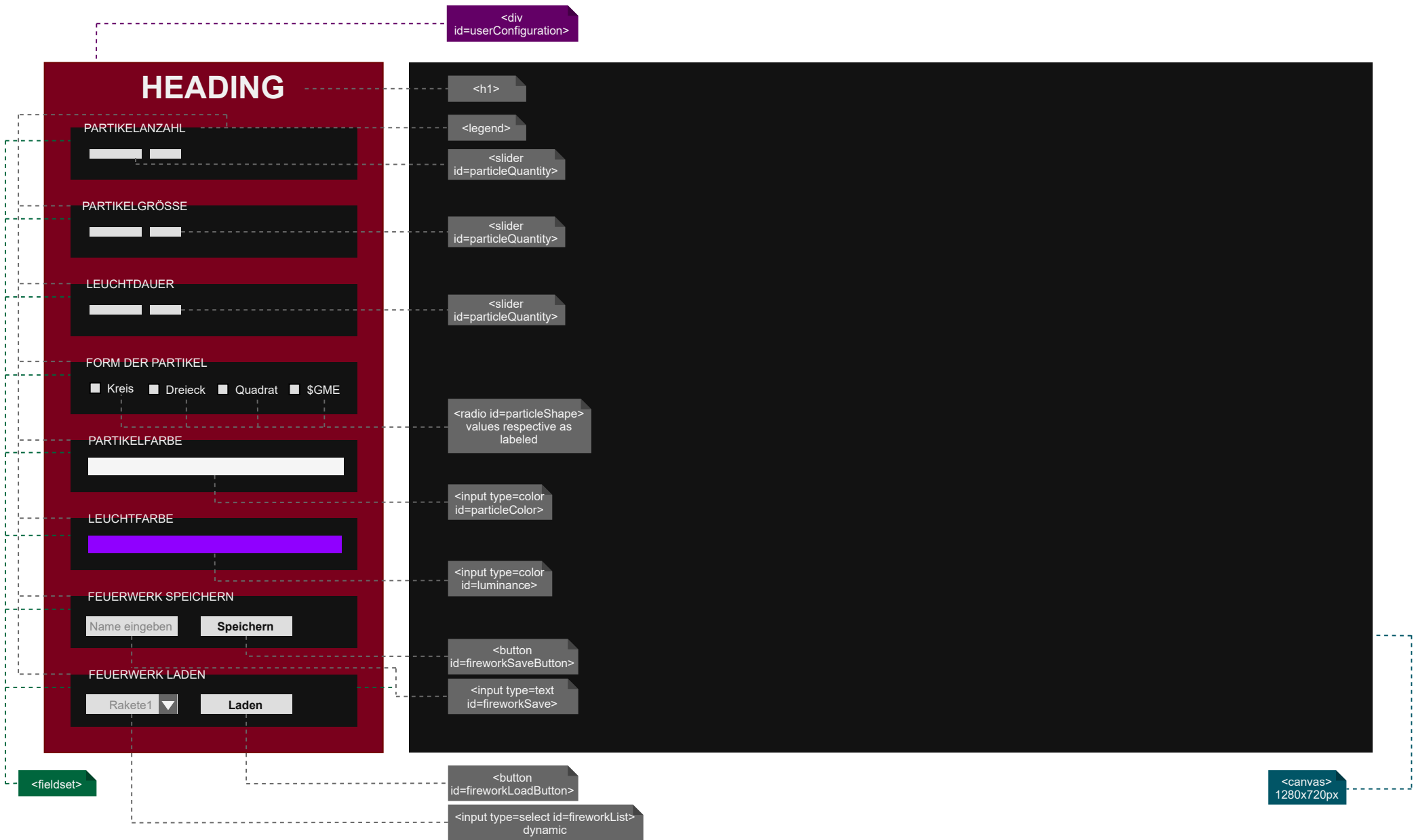
https://sebifyy.github.io/EIA2_Endabgabe-SebastianHoffmann/Index.html

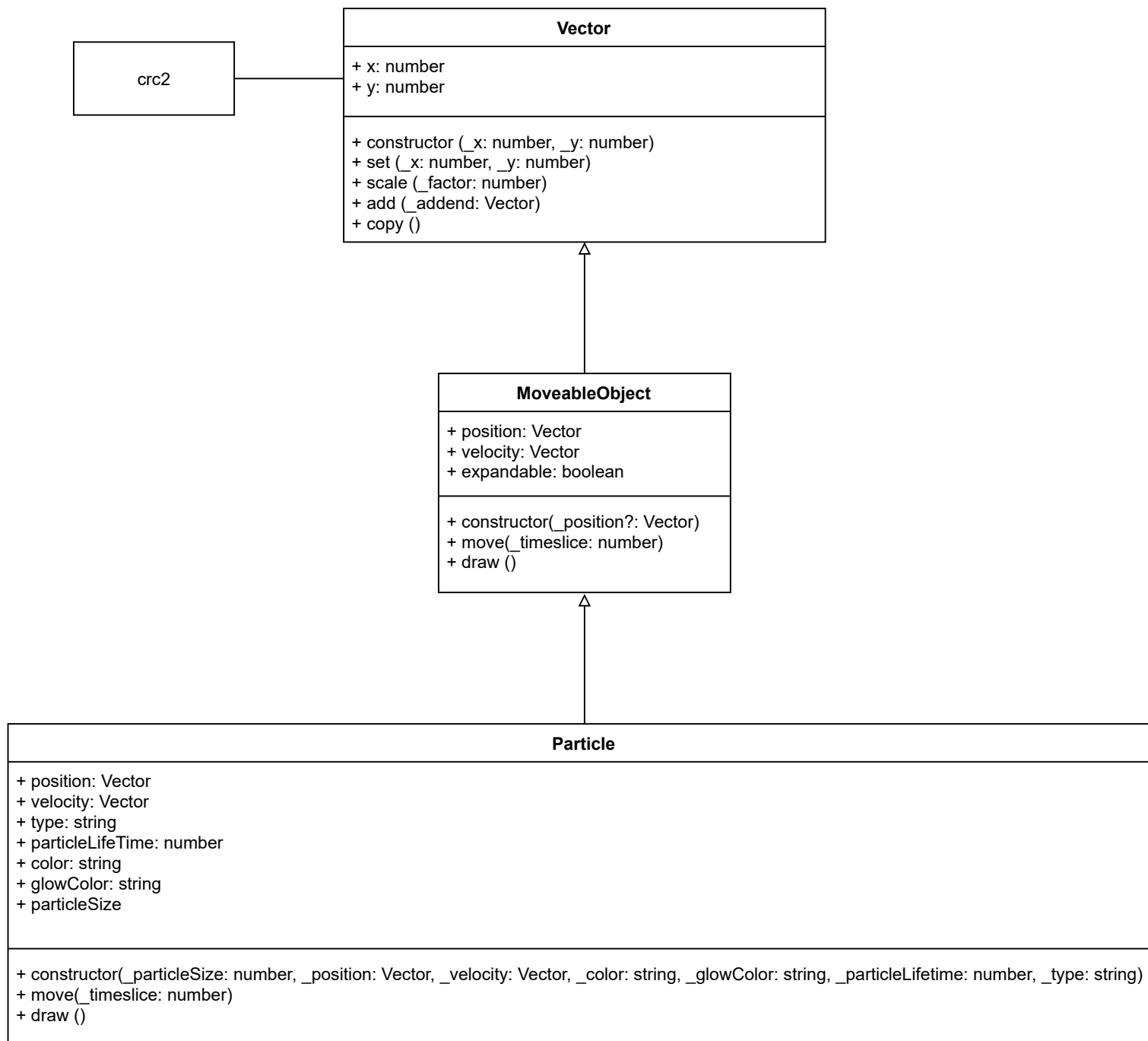
Github repository:

https://github.com/sebifyy/EIA2_Endabgabe-SebastianHoffmann

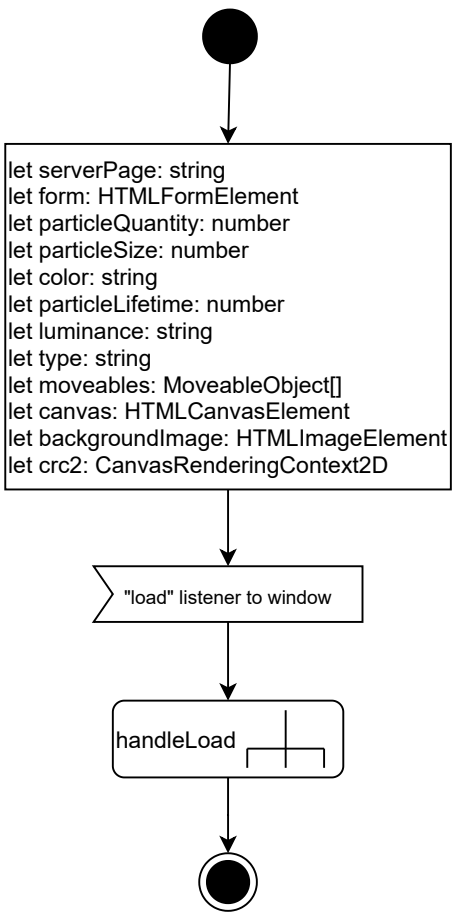


UI Scribble

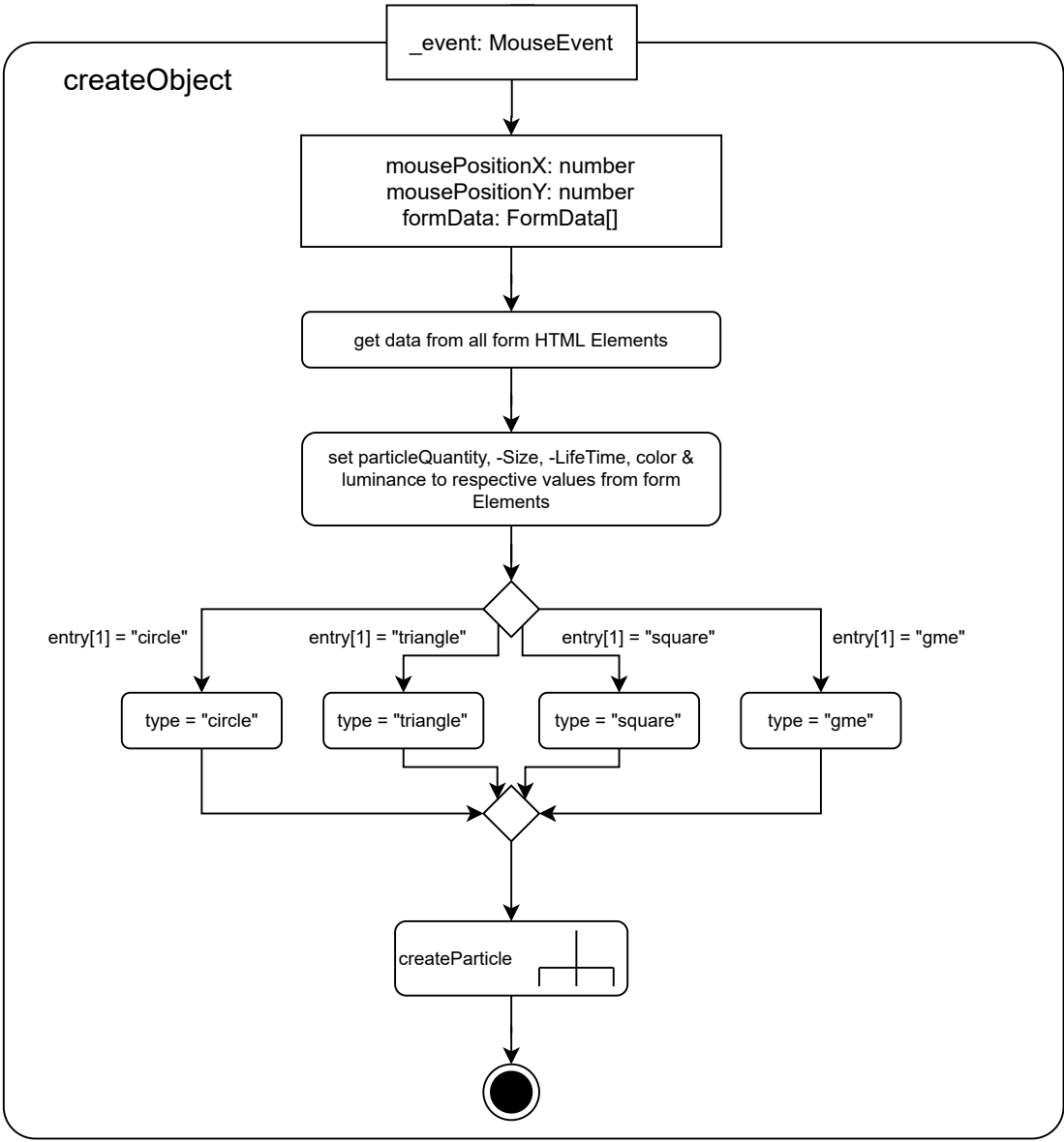
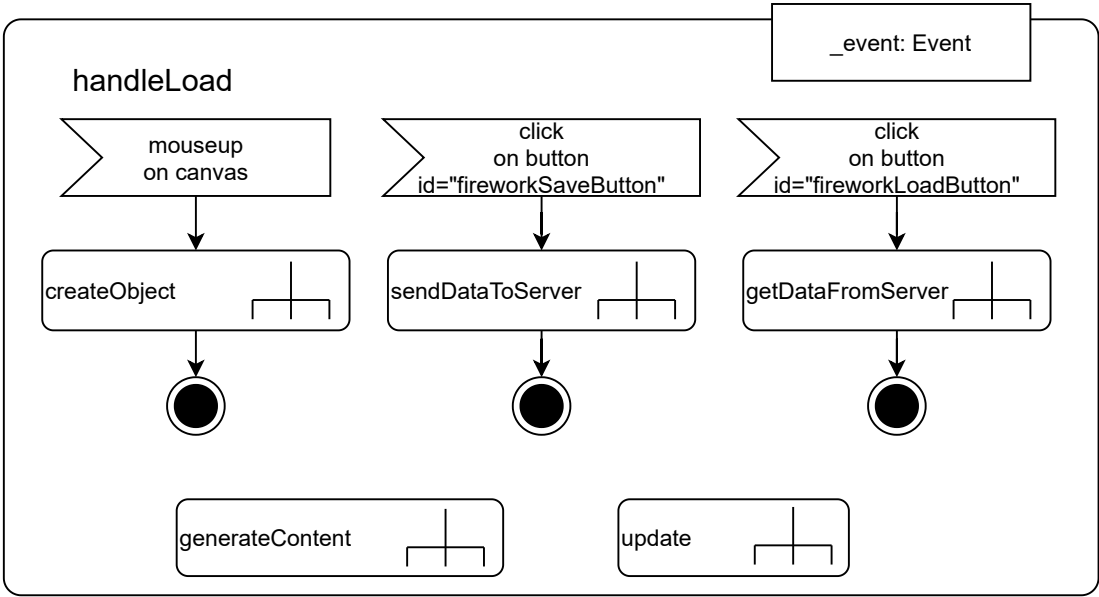


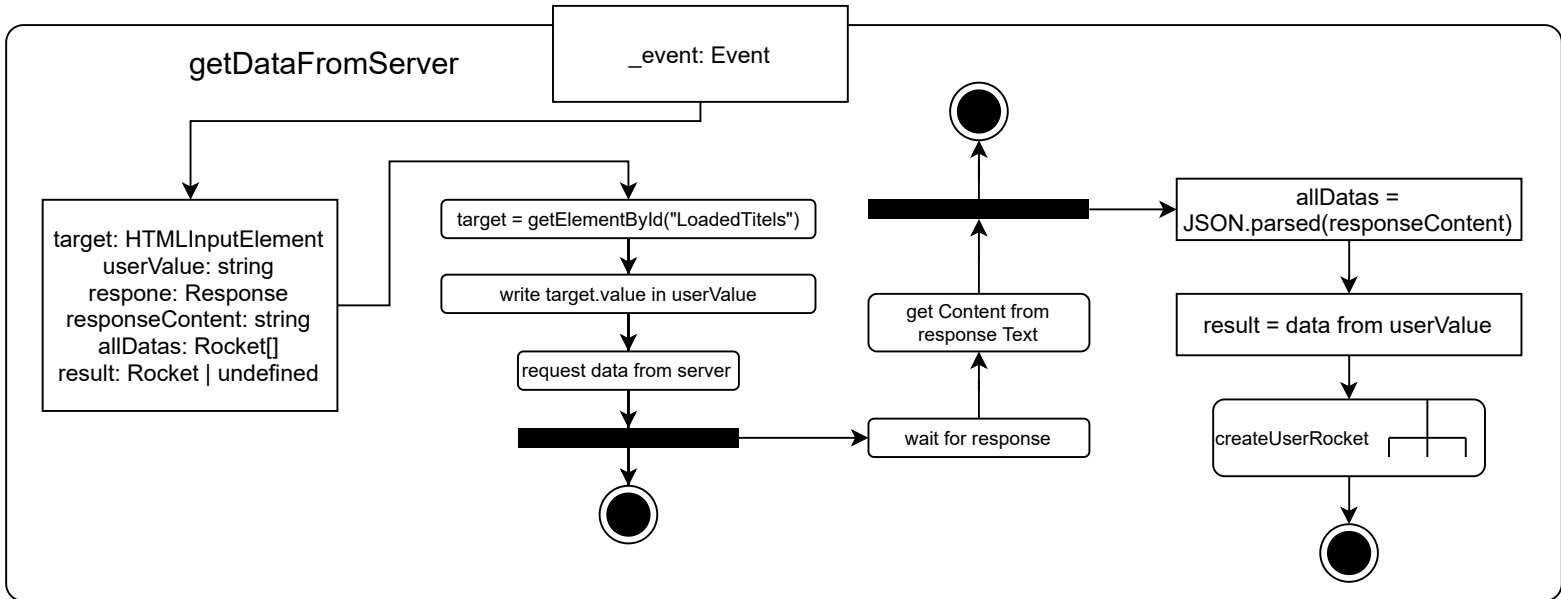
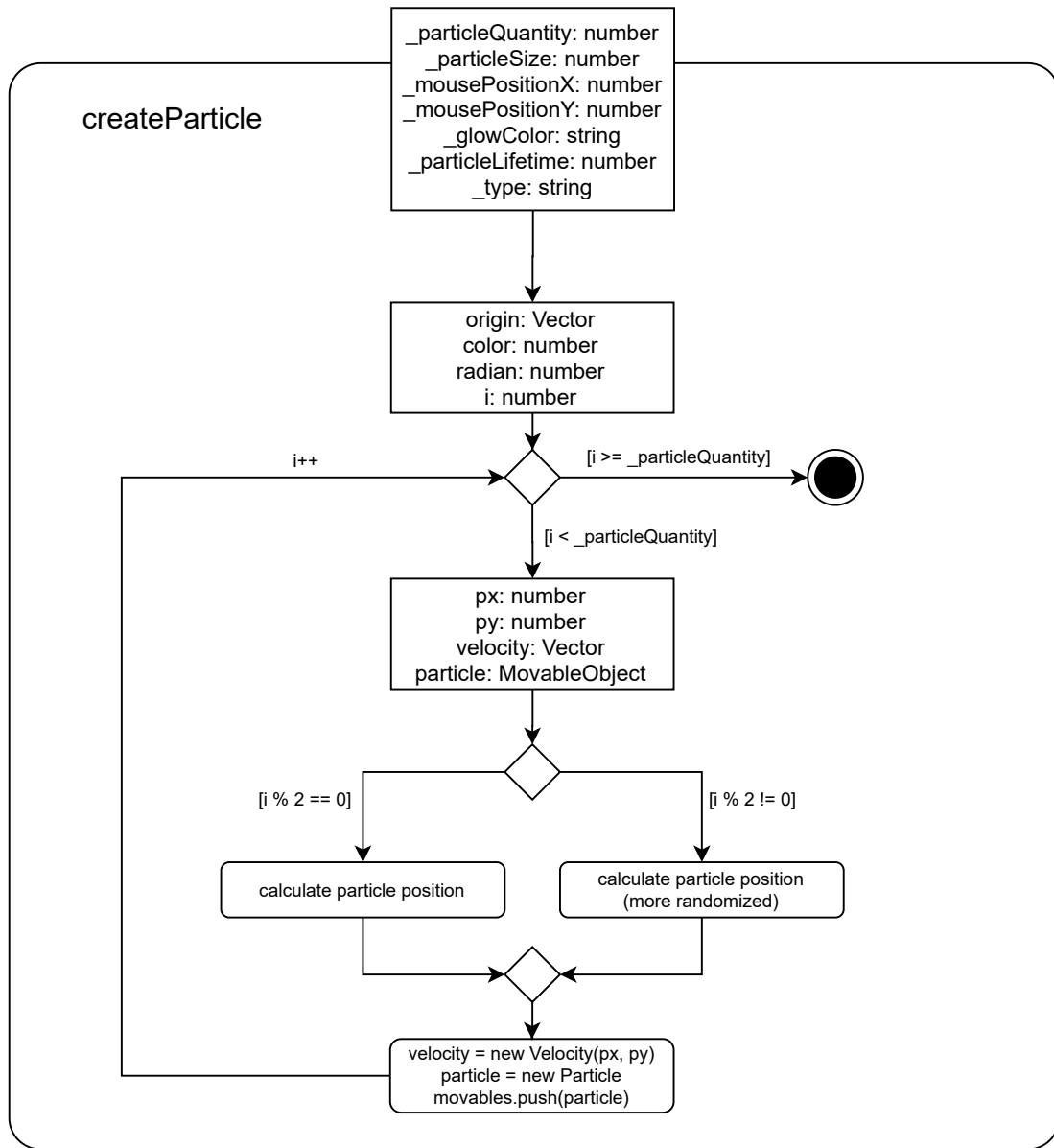


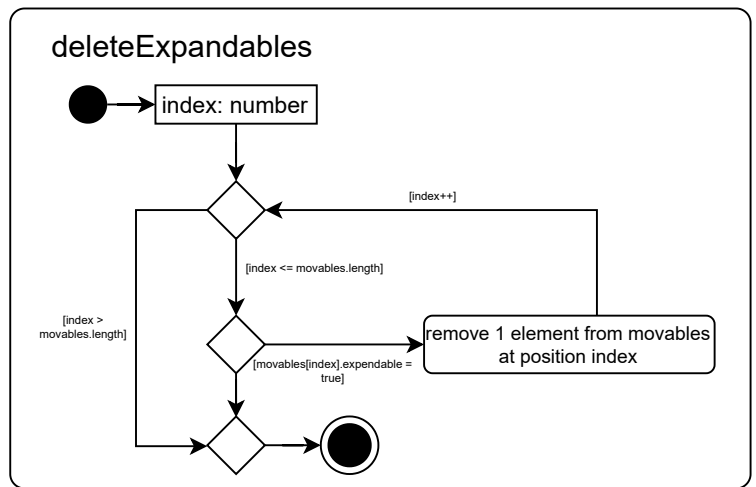
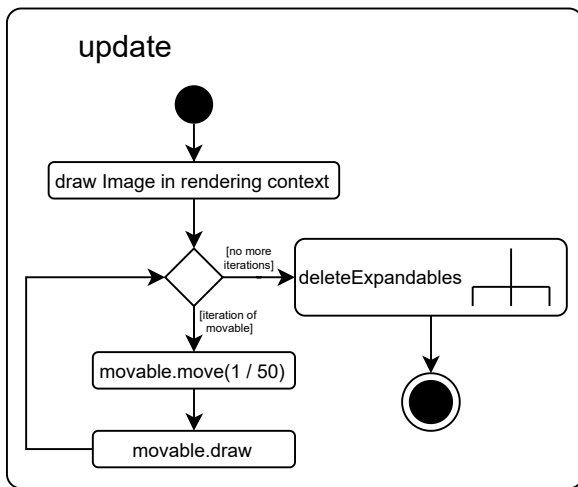
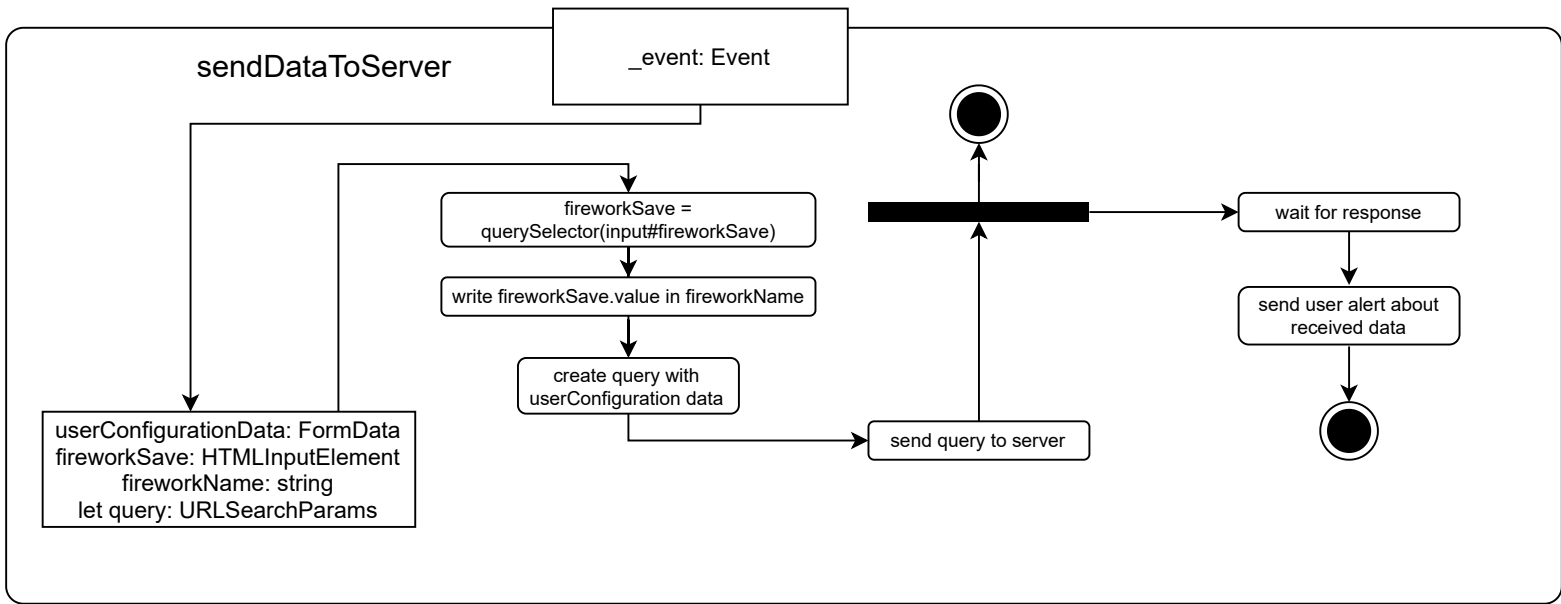
Activity Diagram - Main



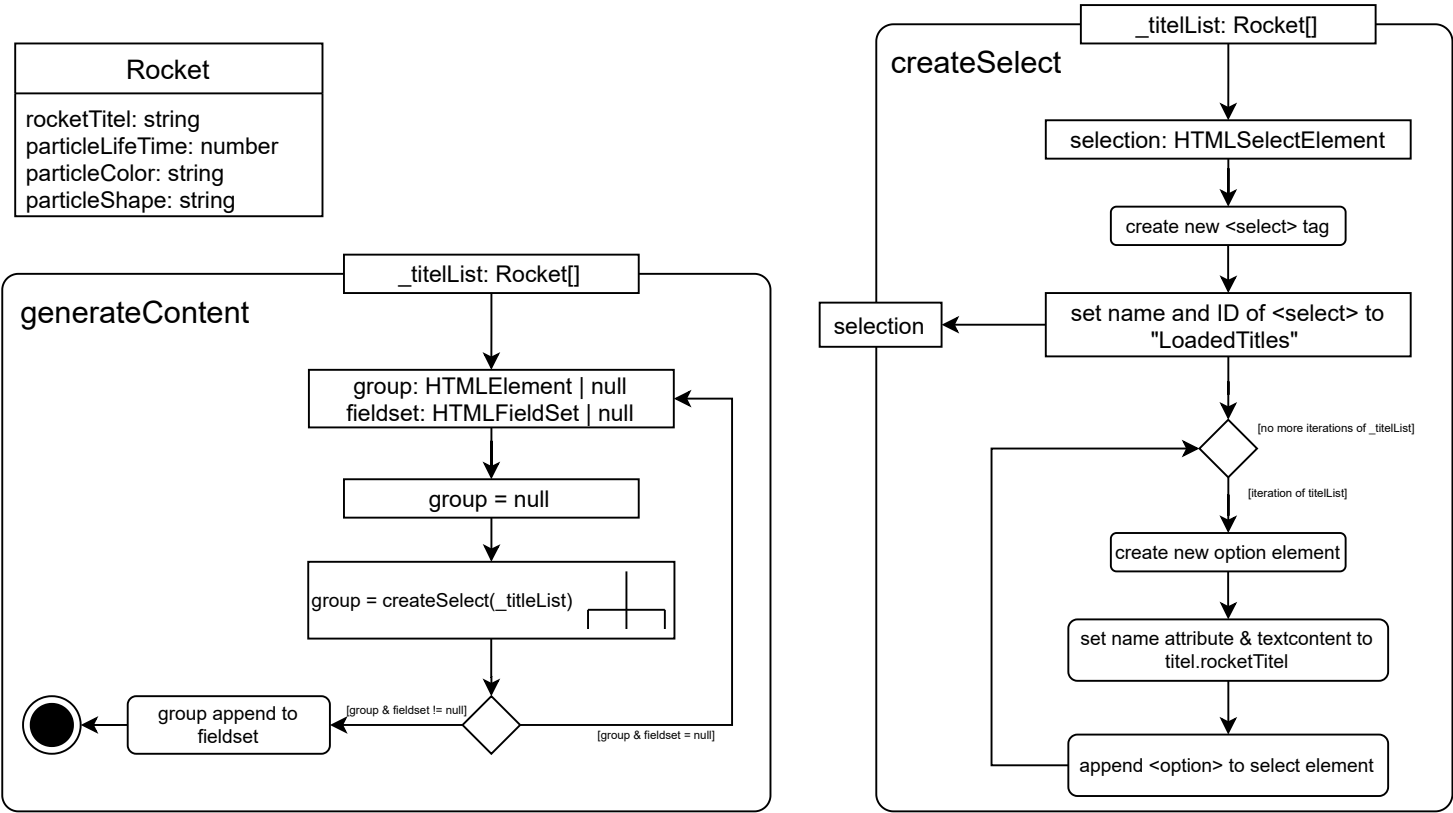
Activitydiagram - Main.ts





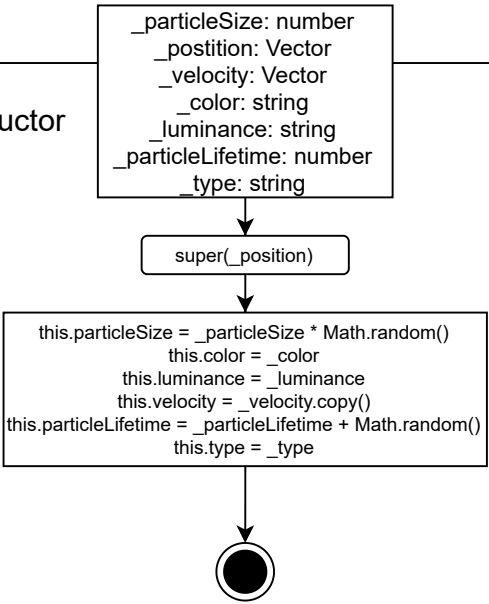


Activitydiagram - generateContent.ts

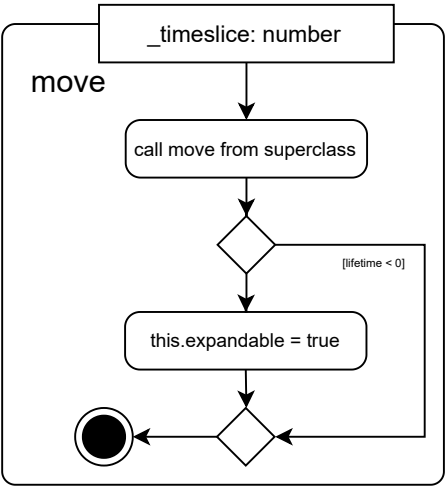


Activitydiagram - Particle.ts

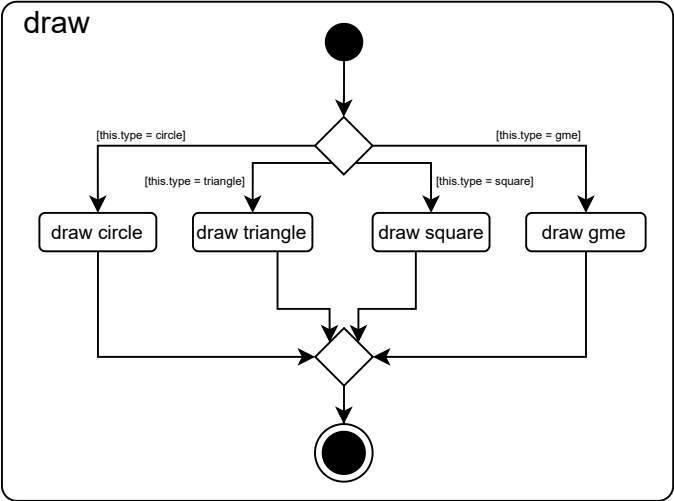
constructor



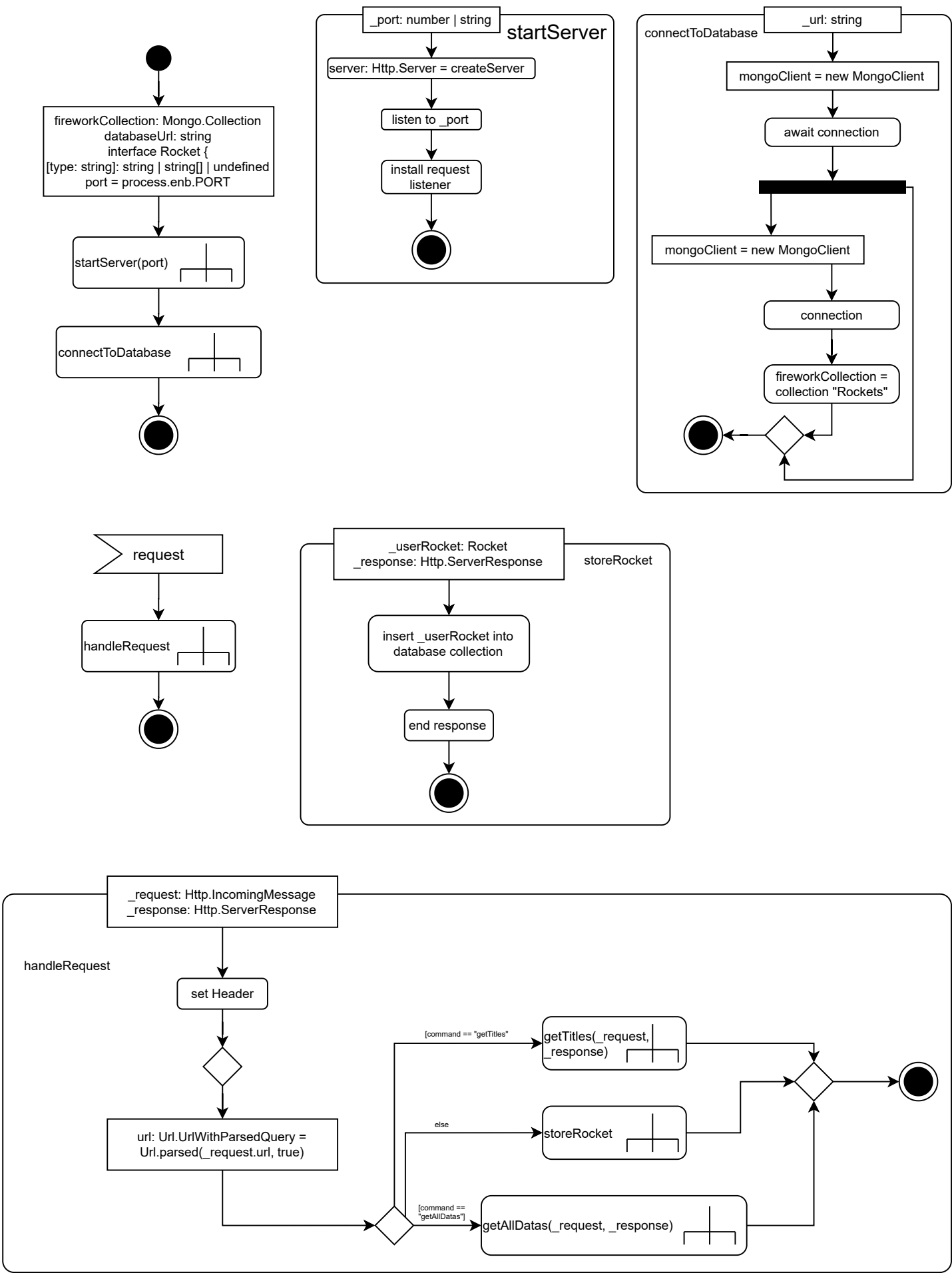
move

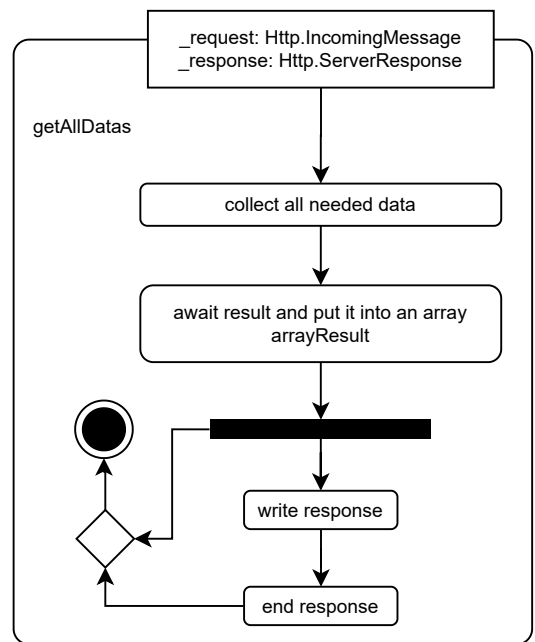
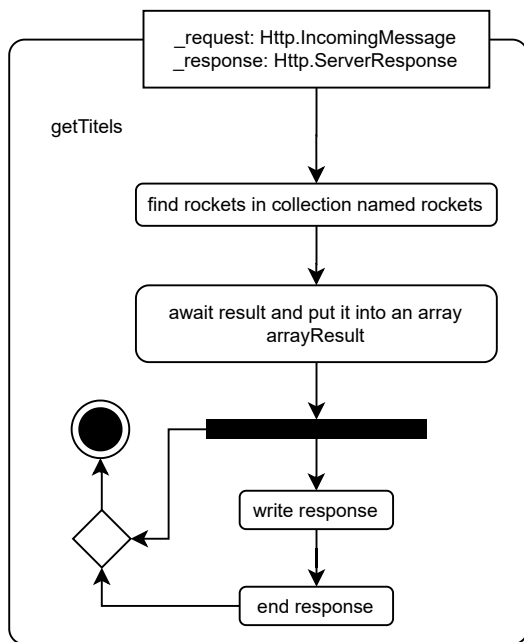


draw



Activitydiagram - Server.ts





Activitydiagram - Vector.ts

