# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  BI19bc | Team:  Daniel Furrer, Marc Geiger, Sebastian Müller |

# Anforderungsdefinition

|  |  |
| --- | --- |
| „Skyhook Simulation“ | |
| **Auftrag:**  (Allgemeine Beschreibung) | Nutzen: Mit der Simulation wollen wir die Therie des Skyhooks simulieren. Wie gut es funktioniert und ob es tatsächlich in der Realität eingesetzt werden könnte.**Szenario:** Die Simulation beinhaltet den Weltraum zwischen Erde und Mars. Die beiden Planeten bewegen sich dabei nicht um die Sonne. Zusätzlich sind bei beiden Planeten sogenannte Skyhooks, welche rotieren. Diese sollen Raumschiffe aufnehmen und zum anderen Planeten schleudern. Dabei verlieren sie an Schwung. Dieser Schwung muss wieder gut gemacht werden, indem vom Skyhook des anderen Planeten ein Raumschiff eintrifft, das vom Skyhook gefangen wird und mithilfe der Schwerkraft der Erde, dieses Raumschiff zur Erde gezogen wird. Am Punkt, wo das Raumschiff am wenigsten kinetische Energie zu Erde hat, wird es losgelassen. Dadurch erhält der Skyhook wieder an Schwung.  **Machbarkeitsabklärung:**   * Folgende Features sind vorab untersucht worden und ..... (Skizze / Mockup) |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden (Funktionalität):**   * Rotierende Skyhooks mit physikalischen Gesetzen * Skyhooks sollen Raumschiffe aufnehmen und abgeben. Dabei verlieren/gewinnen sie an kinetischer Energie * Skyhooks lassen Raumschiffe am höchsten und niedrigsten kinetischen Punkt los. * Skyhooks lassen Raumschiffe so los, dass Skyhook beim anderen Planeten dieses fangen kann. |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Kreativität)**   * Raumschiffe können Personen aufnehmen. * Raumschiffe mit Ladung sind schwerer und verbrauchen mehr kinetische Energie * Raumschiffs aufnehmen und abfeuern mittels der Skyhooks sind so optimiert, damit keine Zeit verschwendet wird. | |

## Planung LB2

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam BildungWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 15.12.2022 | 15.12.2022 |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Analyse, Design: Funktionsmodell, UseCase, GUI, Storyboard) | 12.01.2023 | 12.01.2023 |
| B2 | Teamaufgabe 2:Abgabe: Testvorschrift und Testfälle | 19.01.2023 | 19.01.2023 |
| C | Teamaufgabe 3:Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)Fachgespräch Projektabnahme | 02.02.2023 |  |
| C2 | Einzelaufgabe 4:Abgabe: Ausgefüllter Systemtest | 01.02.2023 | 01.02.2023 |

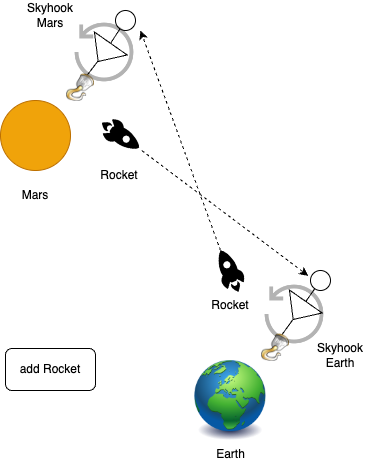
# Lösungsdesign

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Funktionsmodell

Im Folgenden sind die erwarteten Eingaben und Ausgaben beschrieben / dargestellt:

…

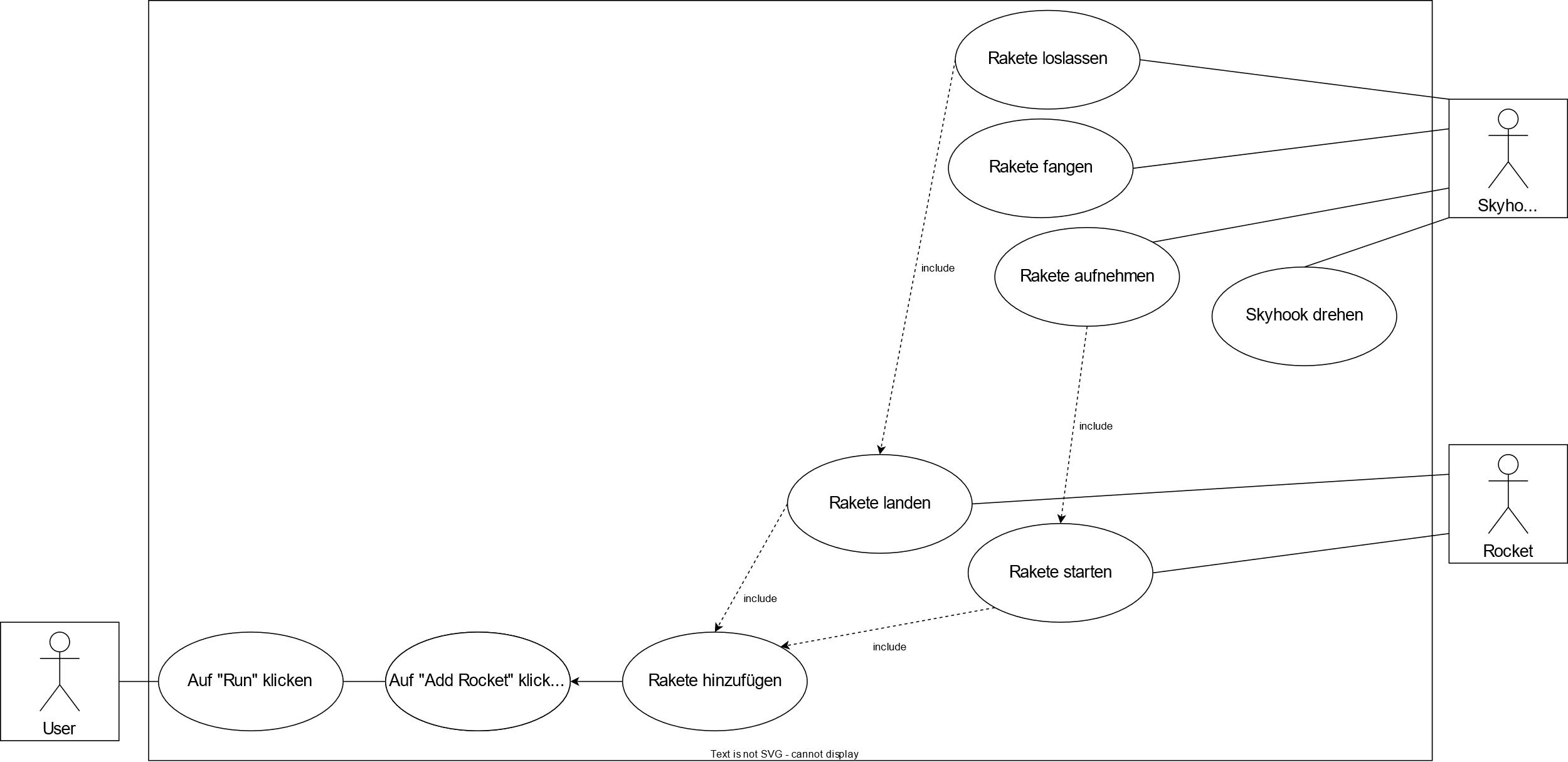


Erklärung:

* Wenn auf "add Rocket" geklickt wird, wird bei der Erde eine startende Rakete erzeugt
* Skyhook Earth nimmt Rakete auf.
* Wenn Rakete aufgenommen ist, schwingt sie mit.
* Am Punkt, wo die Rakete am meisten Schwung hat, wird sie losgelassen.
* Mit diesem Schwung fliegt sie bis zum anderen Planeten.
* Beim anderen Planeten angekommen wird die Rakete von Skyhook gefangen.
* Skyhook verlangsamt die Rakete und wirft sie am nähesten Punkt zum Planeten ab.
* Rakete landet und verschwindet
* Nach 5 sekunden startet sie wieder und macht das gleiche Prozedere in die andere Richtung

## Anwendungsfälle (UseCases)

Folgende Anwendungsfälle sind hier detailliert dokumentiert:



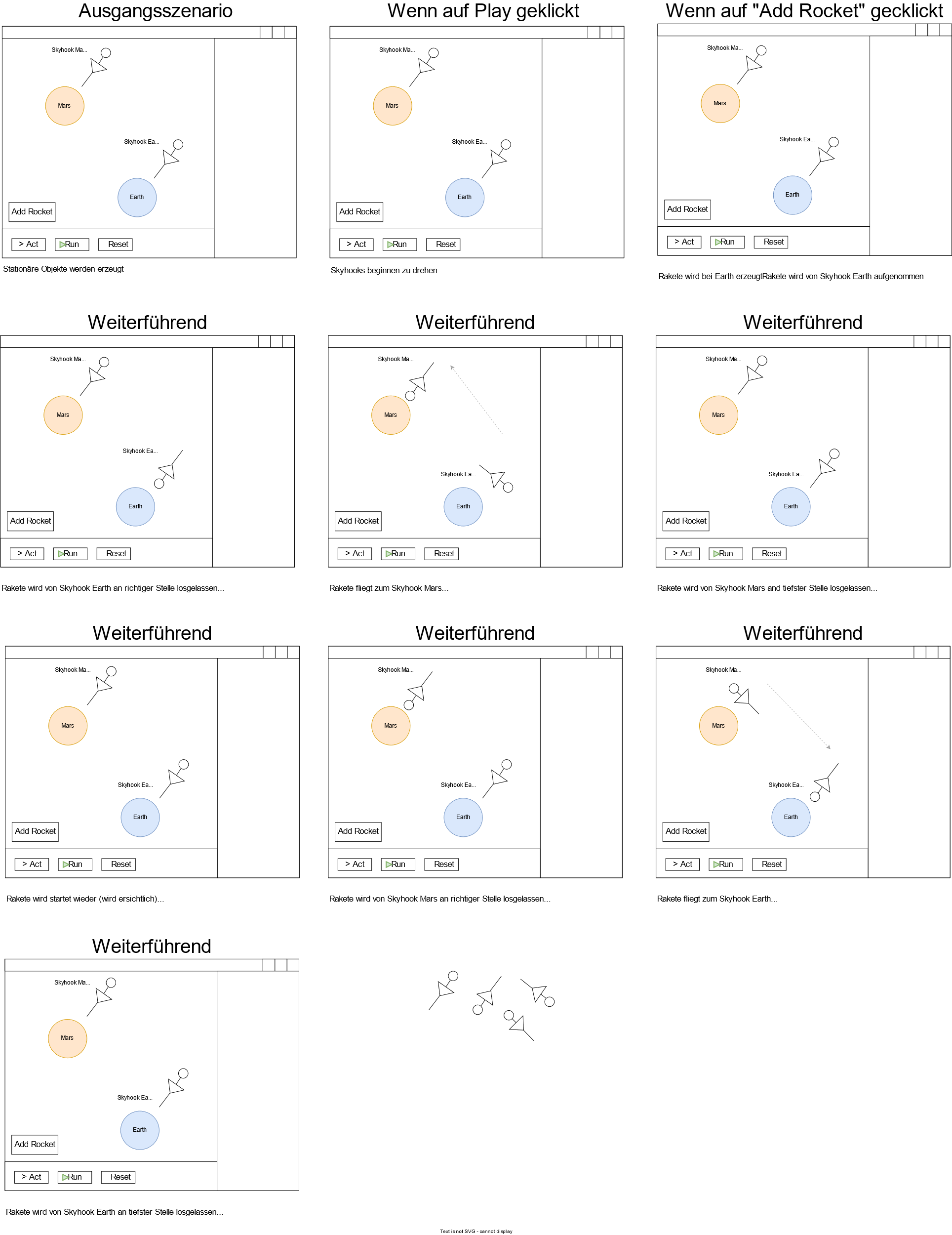
UseCases Beschreibung:

* **"Auf "Run" klicken":**  
  Der Benutzer, welcher mit der Software interagiert, klickt in Greenfoot auf "Run".
* **"Auf "Add Rocket" klicken":**Der Benutzer, welcher mit der Software interagiert, klickt auf "Add Rocket"
* **"Rakete hinzufügen":**Es wird eine Rakete auf dem Planeten *Earth* erzeugt.
* **"Rakete Starten":**Die Rakete startet.
* **"Rakete landen"**Die Rakete landet.
* **"Rakete Aufnehmen"**  
  Der Skyhook nimmt die Rakete nach dem Start auf und schwingt sie.
* **"Rakete loslassen"**Der Skyhook lässt die Rakete an einer optimalen Stelle los. Das kann kurz vor dem Flug zwischen zwei Planeten sein, oder kurz vor der Landung.
* **"Rakete fangen"**Der Skyhook, der die Rakete nicht geschleudert hat, fängt die Rakete an einem richtigen Zeitpunkt.
* **"Skyhook drehen"**Der Skyhook dreht sich jederzeit. Die Schnelligkeit der Drehung zeigt die kinetische Energie des Skyhooks.

## Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

Das Storyboard ist voller Auflösung in Github unter Modelle/Storyboard.drawio zu finden.



# Testvorschrift

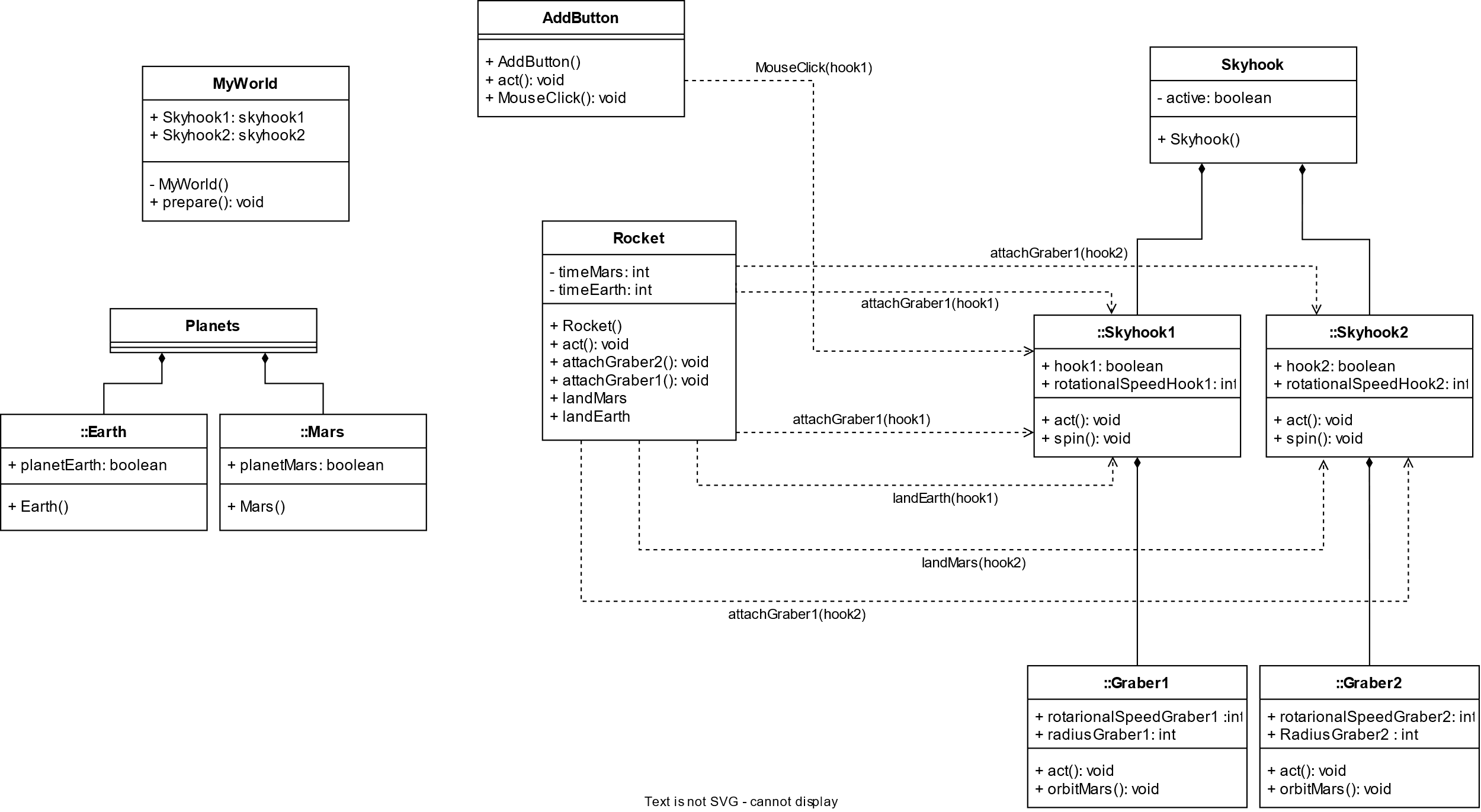
Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument:   
[***M226B\_Systemtests\_Vorlage.docx***](M226B_Systemtests_Vorlage.docx)

# Systemdokumentation

Das erstellte Java-Projekt (Greenfoor-Szenario) ist hier detailliert abgelegt:

[***https://github.com/daenzg/skyhook\_sim***](https://github.com/daenzg/skyhook_sim)

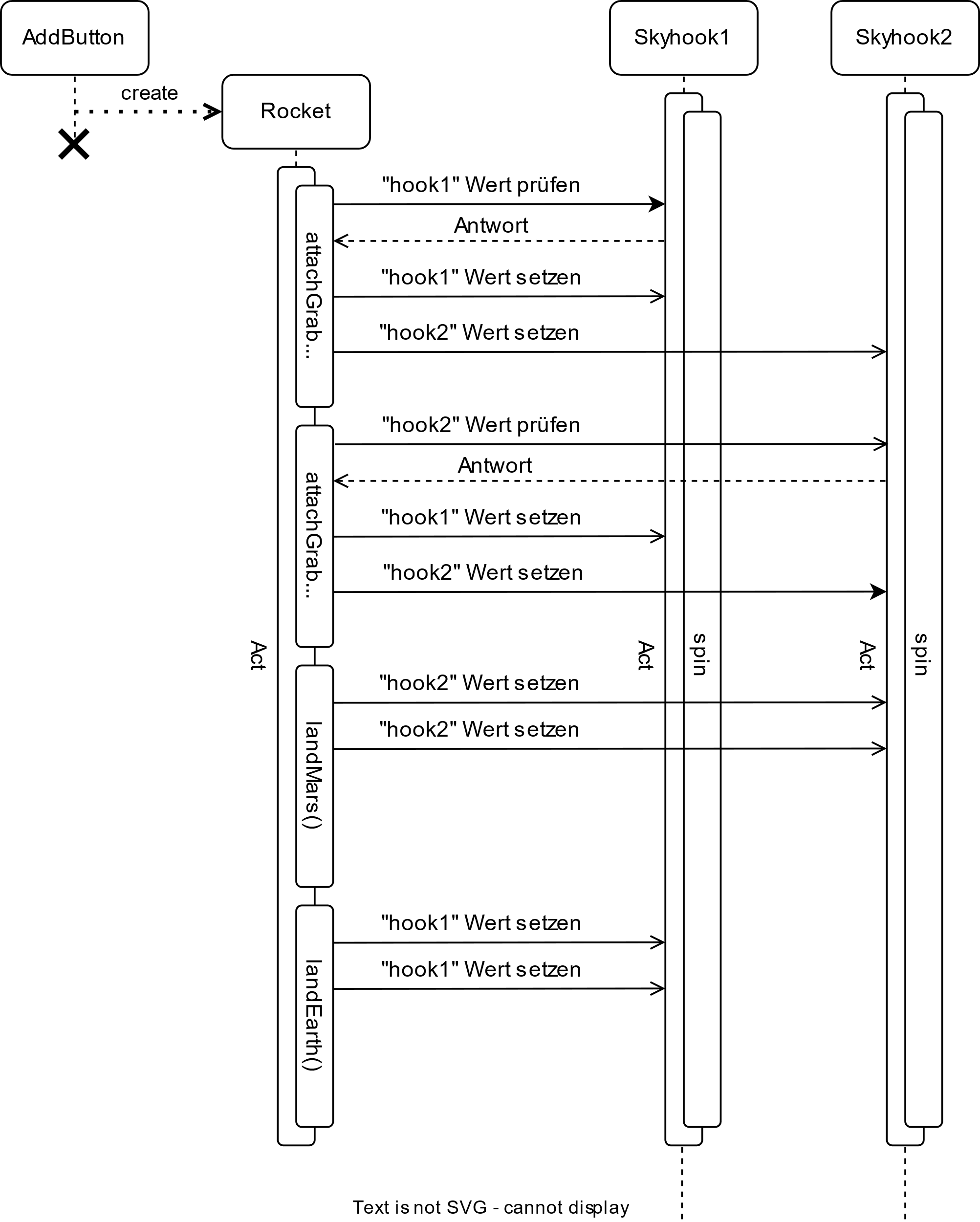
## Statisches Design: Klassendiagramm



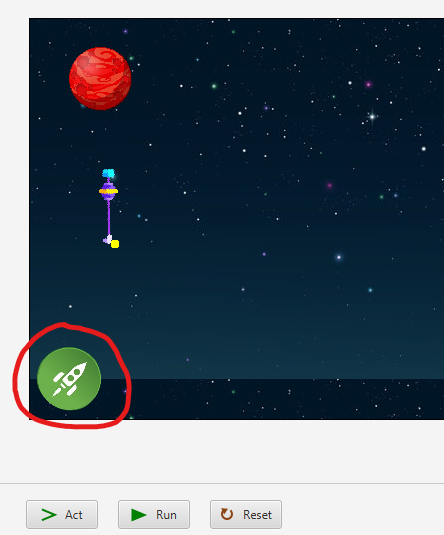
## Umfang / Abgrenzung / Änderungen gegenüber Design

Es wurden keine grundlegenden Änderungen vorgenommen.

## Dynamische Struktur: Sequenzdiagramm



# Bedienungsanleitung

1. Projekt von GitHub klonen:   
   git clone <https://github.com/daenzg/skyhook_sim.git>
2. Project.greenfoot öffnen.
3. Auf "Run" klicken.
4. Auf Knopf mit Rakete klicken, um Rakete hinzuzufügen. ->
5. Beobachten.

# Testprotokoll

Ausgefülltes Testprotokoll siehe Dokument:   
[***M226B\_Systemtests.docx***](M226B_Systemtests.docx)