# Teilnehmer/innen des Teams:

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

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „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 |  |
| C | Teamaufgabe 3:Abgabe Szenario (.zip) mit Inline-Dokumentation, Systemdokumentation (UML Klassen-, Sequenzdiagramm)Fachgespräch Projektabnahme | 02.02.2023 |  |
| C2 | Einzelaufgabe 4:Abgabe: Ausgefüllter Systemtest |  |  |

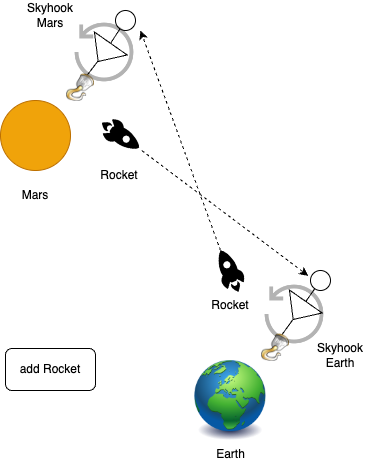
# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

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:

...

(Detaillierte Beschreibung der UseCases)

Legende:

...

## Ablauf

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

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

# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-B2.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

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

***M226B\_ Aufgabe\_3\_Szenario\_IhrName.zip***

## Statisches Design: Klassendiagramm

Folgend die statische Struktur des Szenarios

...

(UML Klassendiagramm mit Assoziationen und Kardinalitäten)

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

...

(Umstände / Anpassungen / Veränderungen)

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen detailliert beschrieben:

...

(Ausführliche Beschreibung der internen Funktionen   
oder Verweis zum Inline-Kommentar mit JavaDoc! (/\*\* @param @return \*\*/)

## Dynamische Struktur: Sequenzdiagramm

Ein zentraler Ablauf eines UseCases ist im Folgenden dargestellt:

...

(Darstellung eines zentralen Ablaufs mittels Sequenzdiagramm)

**Trace: ...**

*...*

# Bedienungsanleitung (Meilenstein C: individuelle Aufgabe 3)

...

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M226B\_LB2\_Testvorschrift\_MS-C2\_Name.docx***