

11. Spezifikation der graphischen Oberfläche

11.1 Die graphische interface

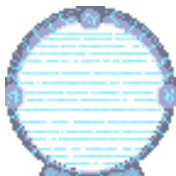
Icons:



Die Ufos die die Rohstoffe fördern können



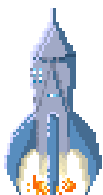
Die Asteroiden aus denen die Rohstoffen gefördert werden können



Die Teleport Portale durch denen die Spieler teleportieren können



Die Roboter die den Siedlern helfen



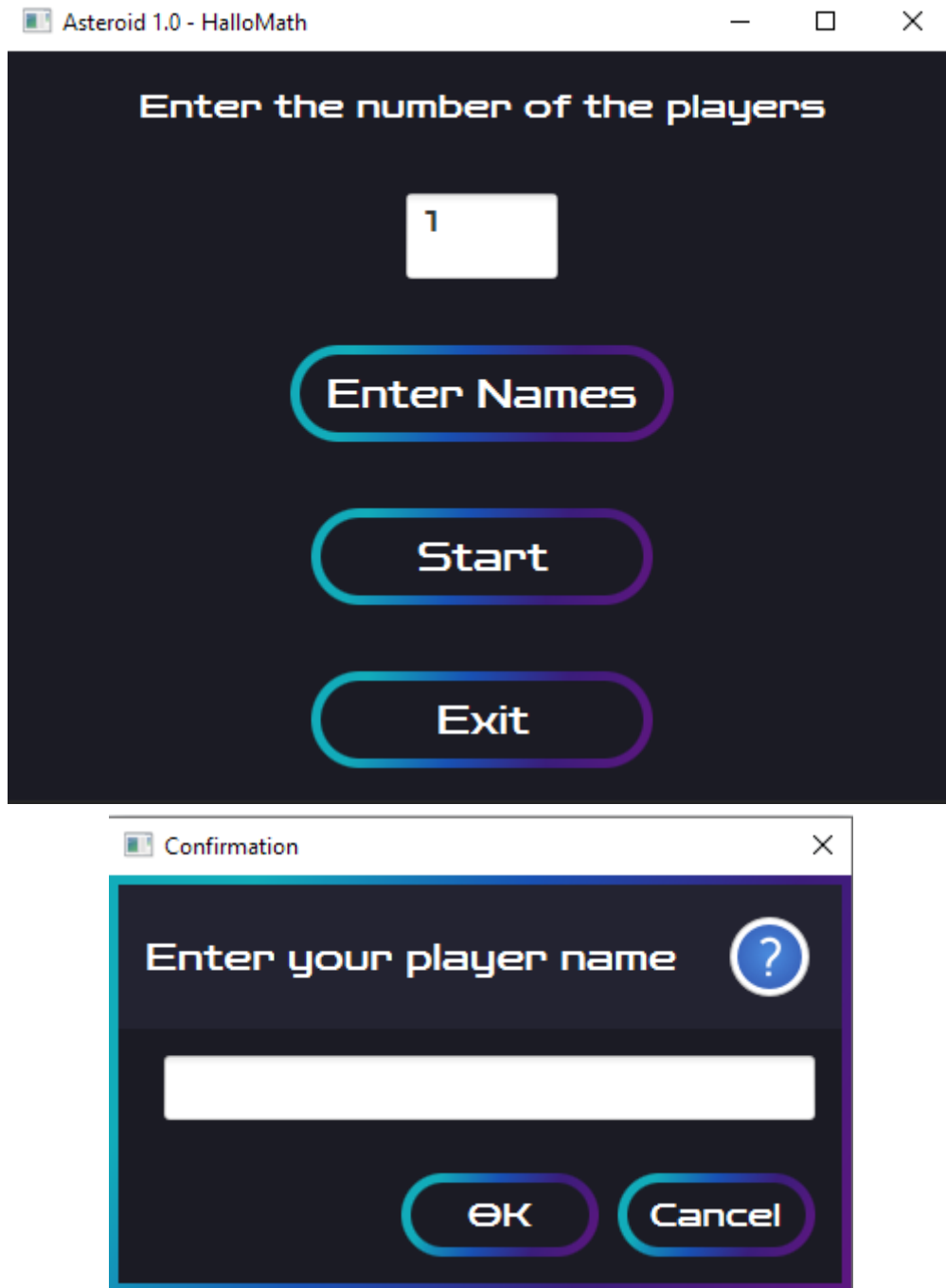
Die Raumschiffe der Siedler mit denen die fahren

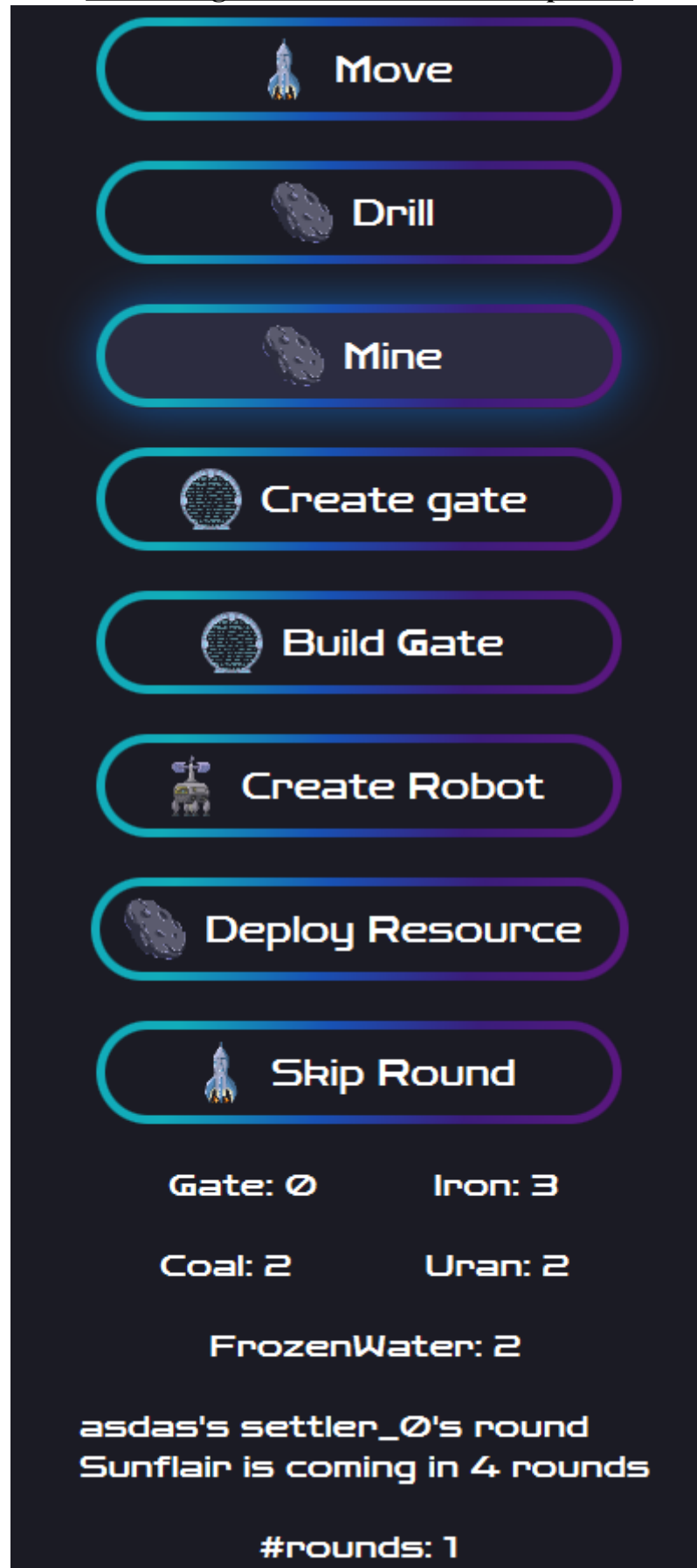


Die Sonne die ein Sonnenflair verursachen kann

Spielfeld:



Menüsystem am Anfang:

Rechts liegende Menü Panel beim Spielen:

11.2 Die Architektur der graphischen System

11.2.1 Funktionsweise der Oberfläche

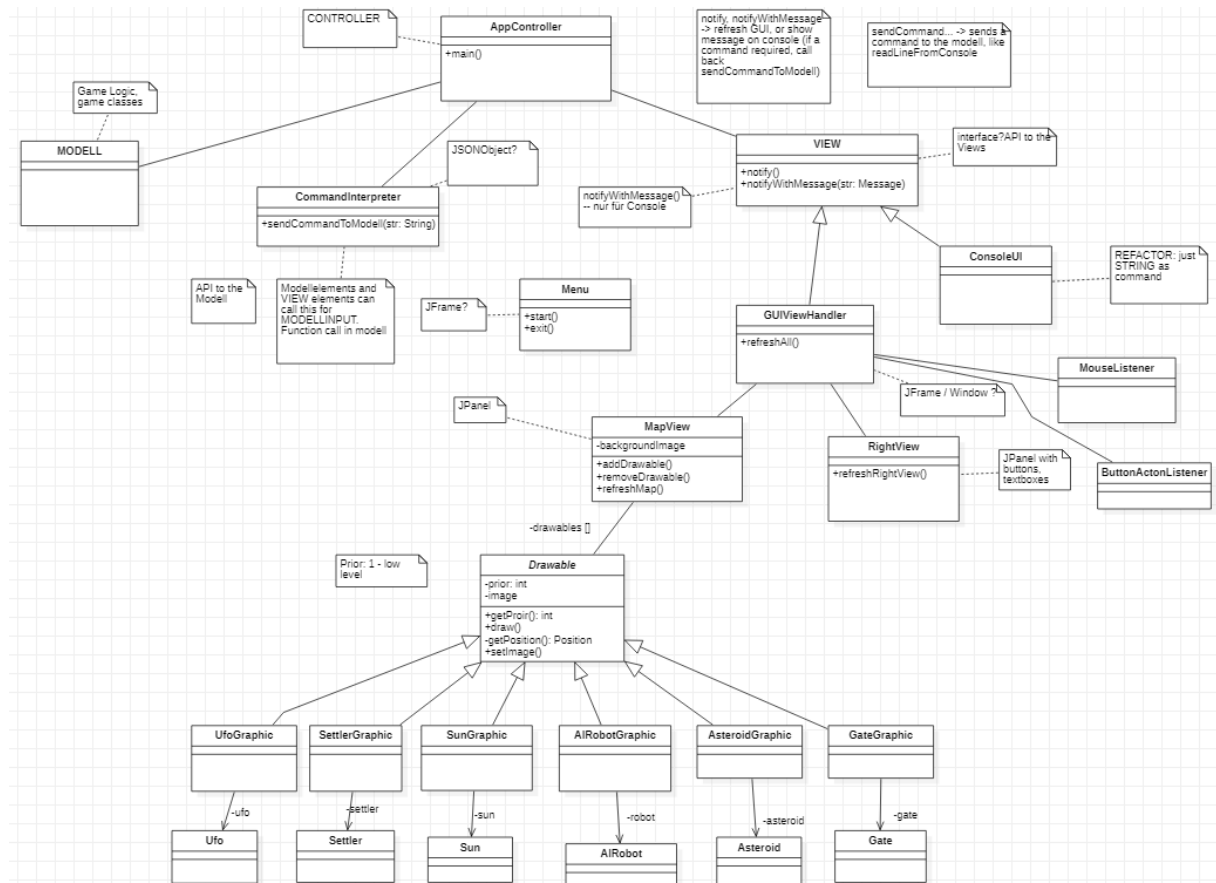
11.2.2 Die Klassenstruktur der Oberfläche

Das Architektur basiert auf dem MVC Pattern (Modell-View-Controller), wo das Spiellogik (Modell) durch das Controller erreichbar, und das Modell benachrichtigt das View über ihre Änderungen.

Durchschau:

- Modell: Das Spiel Logik, das wurde auf den vorigen, alten Klassendiagramme entwickelt. Es ist verantwortlich für die Datenbehandlung und Datenspeicherung, und das ganze Spiellogik.
- View: Verantwortlich für IO der Programm, behandelt das In und Output, informiert die Spieler über das Spielzustand.
- Controller: API zur Modell, dadurch werden die neue Befehle, Aufrufe von dem View zur Modell geschickt, und auf dem Modell interpretierbare Sprache übersetzt.

Unsere Modell folgt das gemischte (pull and push) Prinzip, weil durch View wird kontrollierfunktionalität aufgerufen (es führt weiter zum Modell), und das Modell benachrichtigt auch das View, das etwas erfrischen soll.



11.3 Auflistung der graphischen Objekte

11.3.1 View

- Verantwortung

Abstrakte Basisklasse, das Konsole und das Graphische Anzeige bringt zusammen, und aktualisiert.

- Methoden

- **+void notify():** Benachrichtigt über dem Änderung.
- **+void notifyWithMessage(string Message):** Benachrichtigt über dem Änderung mit Text.

11.3.2 GUIViewHandler

- **Verantwortung**

Eine Klasse, die für die Anzeige der grafischen Elemente verantwortlich ist.

- **Basisklassen**

View → GUIViewHandler

- **Methoden**

- **+void refreshAll()**: Aktualisiert jeden Teil der GUI, so dass der korrekte Status des Spiels angezeigt wird.

11.3.3 ConsoleUI

- **Verantwortung**

Es ist für das Erscheinungsbild der Konsole verantwortlich und spielt beim Testen eine wichtige Rolle.

- **Basisklassen**

View → ConsoleUI

- **Methoden**

11.3.4 RightView

- **Verantwortung**

Diese Klasse ist für die grafischen Komponenten auf der rechten Seite des Fensters verantwortlich. Es ist ein Panel, das Schaltflächen für Befehle enthält und den Status des Spiels ausschreibt.

- **Methoden**

+void **refreshRightView()**: Aktualisiert die rechten Seite der grafischen Oberfläche.

11.3.5 MapView

- **Verantwortung**

Diese Klasse ist für die grafischen Komponenten auf der linken Seite des Fensters verantwortlich. Es ist ein Panel, auf dem das Spielfeld ist.

- **Attribute**

-**backgroundImage**: Die Hintergrund mit anderen Bildern (UFOs, Asteroiden ..).

- **Methoden**

+void **refreshMap()**: Aktualisiert den Inhalt des Panels.

+void **removeDrawable()**: Fügt dem Pannel ein Element hinzu, das angezeigt werden soll.

+void **addDrawable()**: Entfernt ein Element, das für das Panel angezeigt werden soll

11.3.6 Drawable

- **Verantwortung**

Dies ist eine abstrakte Basisklasse zum Zeichnen jedes Spielobjekts, das Sie auf dem Spielfeld sehen.

- **Attribute**

-**image**: Enthält das dem Spielobjekt entsprechende Bild. So wird das Objekt auf dem Feld angezeigt.

- **int prior**: Diese Zahl stellt die Hierarchie der grafischen Objekte dar und zeigt an, ob ein Objekt ein anderes Objekt abdeckt oder von einem anderen abgedeckt wird.

- **Methoden**

+int **getPrior()**: Gibt prior zurück.

+**void draw()**: Zeichnet das Objekt, das die Klasse darstellt.

+**Position getPosition()**: Gibt die Position des Objektes zurück.

+**void setImage()**: Setzt das entsprechende Bild auf das Objekt.

11.3.7 UfoGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen die UFOs.

- **Basisklassen**

Drawable → **UfoGraphic**

- **Attribute**

- **Methoden**

11.3.8SettlerGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen der Settler.

- **Basisklassen**

Drawable → **SettlerGraphic**

- **Attribute**

- **Methoden**

11.3.9SunGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen die Sonne.

- **Basisklassen**

Drawable → **SunGraphic**

- **Attribute**

- **Methoden**

11.3.10 AIRobotGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen die AIRoboten.

- **Basisklassen**

Drawable → **AIRobotGraphic**

- **Attribute**

- **Methoden**

11.3.11 AsteroidGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen die Asteroiden.

- **Basisklassen**

Drawable → **AsteroidGraphic**

- **Attribute**

- **Methoden**

11.3.12 GateGraphic

- **Verantwortung**

Diese Klasse ist verantwortlich für das Erscheinen der Teleporter.

- **Basisklassen**

Drawable → **GateGraphic**

- **Attribute**

- **Methoden**

11.3.13 Menu

- **Verantwortung**

Es is eine JFrame und ist für die Anzeige der Homepage verantwortlich.

- **Basisklassen**

- **Attribute**

- **Methoden**

- + **void start():** Das Spiel beginnt mit dem Aufrufen diese Funktion.
- + **void exit():** Das Spiel beendet mit dem Aufrufen diese Funktion.

11.3.14 CommandInterpreter

- **Verantwortung**

- **Basisklassen**

Interfészek

- **Attribute**

- **Methoden**

+sendCommandToModel(str: String):

Tagebuch

Anfang	Dauer	Teilnehmer	Beschreibung
2021.04.17	2 Stunde	Alle	Besprechen der Realisationsmöglichkeiten der MVC Modell
2021.04.19	1 Stunde	Hrotkó	Bilder/Ikons modifizieren und ein Preview herstellen
2021.04.19.	1,5 Stunde	Pongrácz	MVC Planung, Diagramm herstellen
2021.04.20	2 Stunde	Alle	Dokumentation schreiben und Beschreibung
2021.04.20	45 Minuten	Domokos	Dokumentum Skeleton schreiben