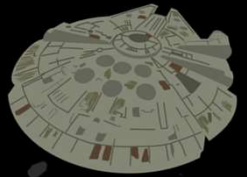


INVADING SPACE

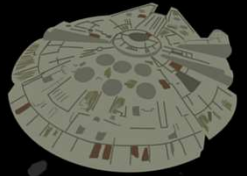
Patrick Bigge, Leon Henne, Maximilian Kinzler,
Jan Moormann und Jasmin Noll



Gliederung

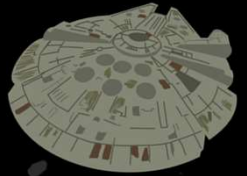


1. Anforderungen an das Programm
2. Organisation
3. Ordnerstruktur
4. Umsetzung
5. Aufbau
6. Programmstruktur
7. Herausforderungen
8. Ergebnis
9. Vorstellung des Spiels



Anforderungen an das Programm

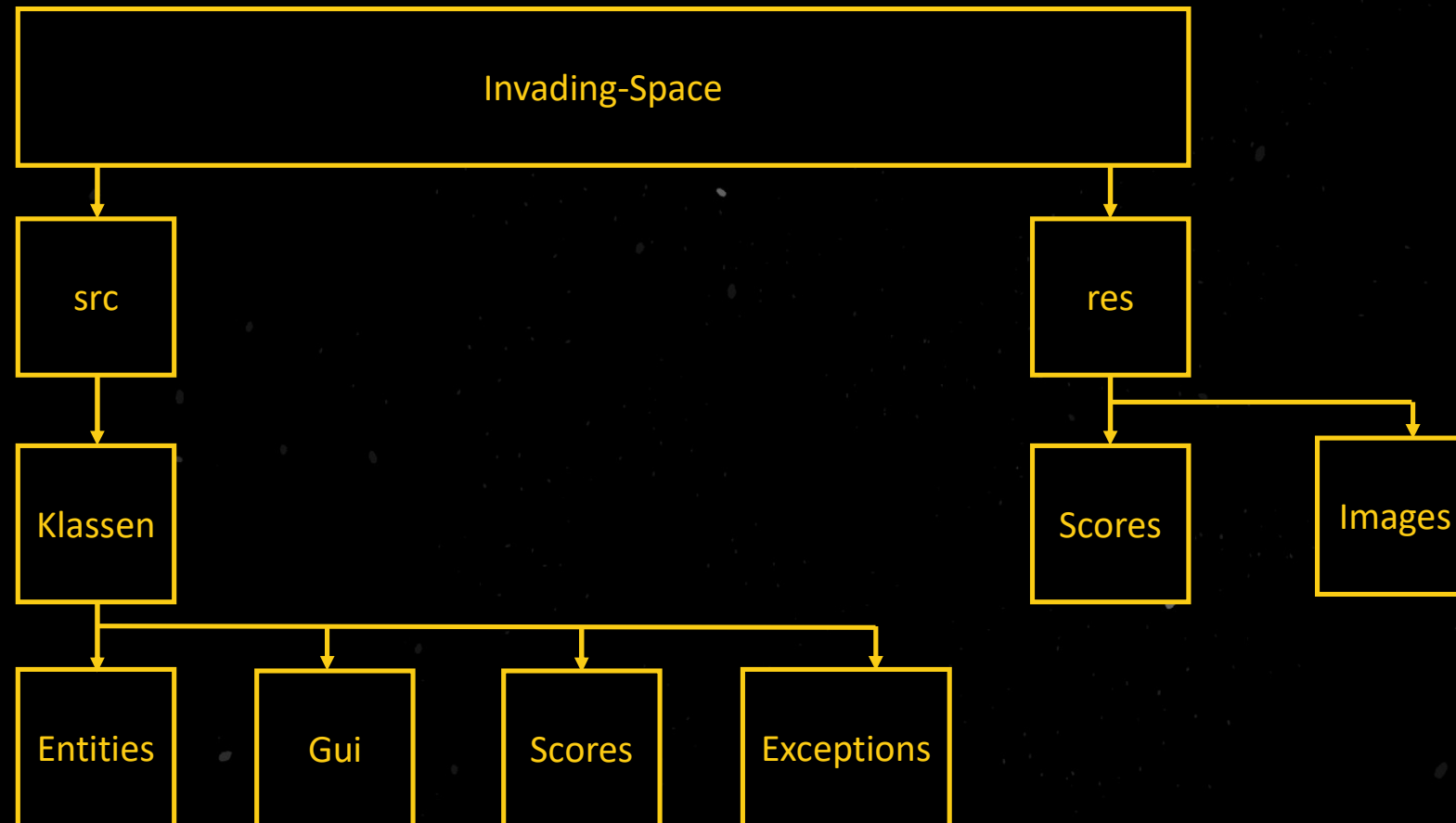
- JavaFX
- Auswahl des Spielmodus
- Bewegung der Objekte
 - Gegner
 - Raumschiff
- Schuss der Objekte
- Punkte für das Töten der Gegner
- Highscores
- Beenden des Spiels



Organisation

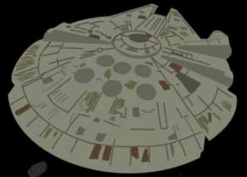
- Aufteilung der Aufgaben
- Nutzung von GitHub
- Pflichttermine
 - Aktueller Stand
 - Ausblick
- Zusätzlicher Termin zum Programmieren
- ToDo-Kommentare in IntelliJ

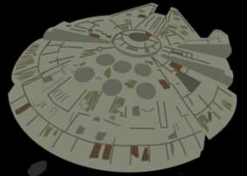
Ordnerstruktur



Umsetzung

- Externe JavaFX Bibliothek
- Aufbau einer Oberfläche
- Feste Fenstergröße
- Feste Koordinaten
- An Star Wars orientiert

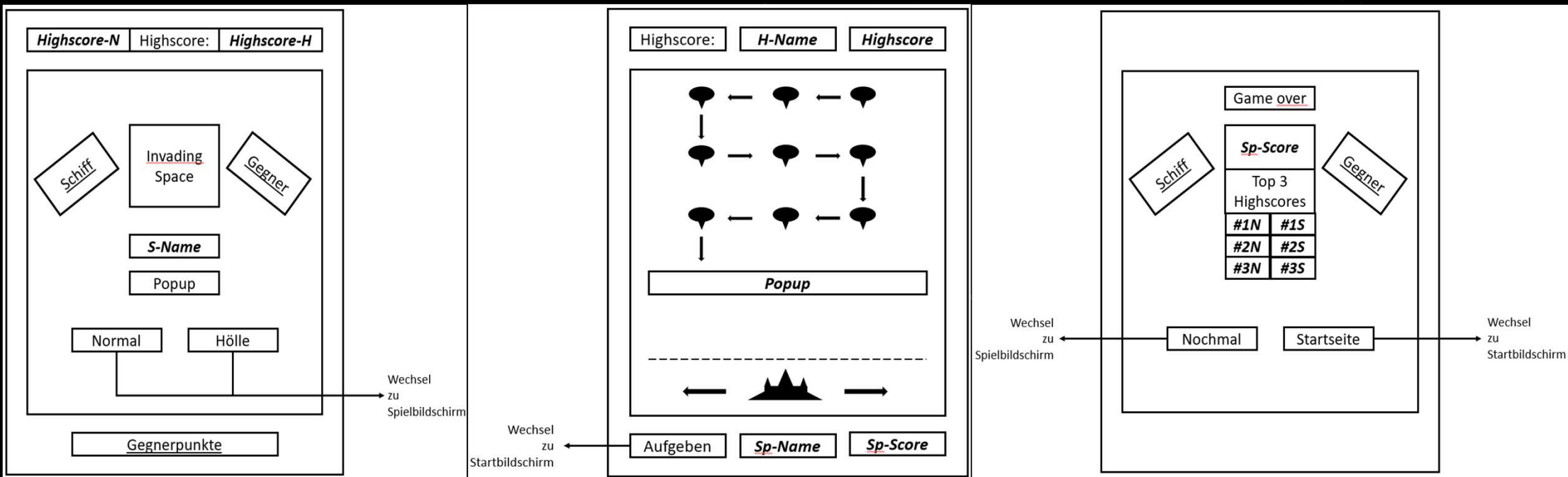




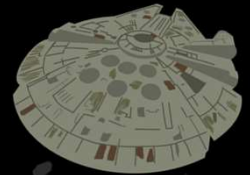
Programmstruktur

- GUI
- Framework
- Klassen der beweglichen Objekte
 - Bewegliche Objekte
 - Raumschiff
 - Gegner
 - Schuss
- Klasse ObjektSteuerung
- Spielablauf
- Spieler
- Punkteliste

GUI



GUI



The screenshot displays the JavaFX IDE interface for a game titled "invading Space". The central canvas features a black background with a yellow border. At the top, the word "Highscore" is written in yellow. Below it, the text "invading Space" is displayed in a larger yellow font. A text input field with the placeholder "Enter your Name" is centered on the canvas. The canvas is divided into three vertical sections labeled "Normal", "Hölle", and "Hölle" (repeated). The "Normal" section shows a green alien ship. The "Hölle" sections show blue alien ships. At the bottom, three yellow numbers "10", "20", and "50" are displayed, each next to a blue alien ship. The IDE interface includes a Library panel on the left with categories like Containers, Controls, and Shapes. The Inspector panel on the right shows the properties of the selected TextField component, including Prompt Text, Text, Font, Editable, and Node properties.

Library

Containers

- Accordion
- Accordion (empty)
- AnchorPane
- BorderPane

Controls

- Gluon
- Menu
- Miscellaneous
- Shapes
- Charts
- 3D

Document

- Hierarchy
- Controller

Controller class

Klassen.GUI.Startbildschirmcontroller

☐ Use fxroot construct

Assigned fxid

fxid	Component
lblSpace	Label
btn_hölle	Button
btn_normal	Button
lblInvader	Label
lblaberbesser	Label
txtNamensfeld	TextField
lblPopup	Label
bildBomber	ImageView
bildTiefighter	ImageView
bildGrossschiff	ImageView
bildMilenium	ImageView
bildTiefighter2	ImageView
bildBomber2	ImageView
bildGrossschiff2	ImageView
lbl10	Label

Inspector

Properties: TextField

Text

Prompt Text: Enter your Name

Text:

Font: System 23px

Specific

Editable: ☒

Node

Alignment: CENTER

Disable: ☐

Opacity: 1

Node Orientation: INHERIT

Visible: ☒

Focus Traversable: ☒

Cache Shape: ☒

Center Shape: ☒

Scale Shape: ☒

Opaque Insets: 0 0 0 0

Cursor: Inherited (Default)

Effect: +

JavaFX CSS

Style

-fx-background-color: "black"

Style Class

Stylesheets

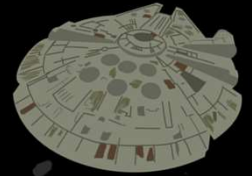
+

Id

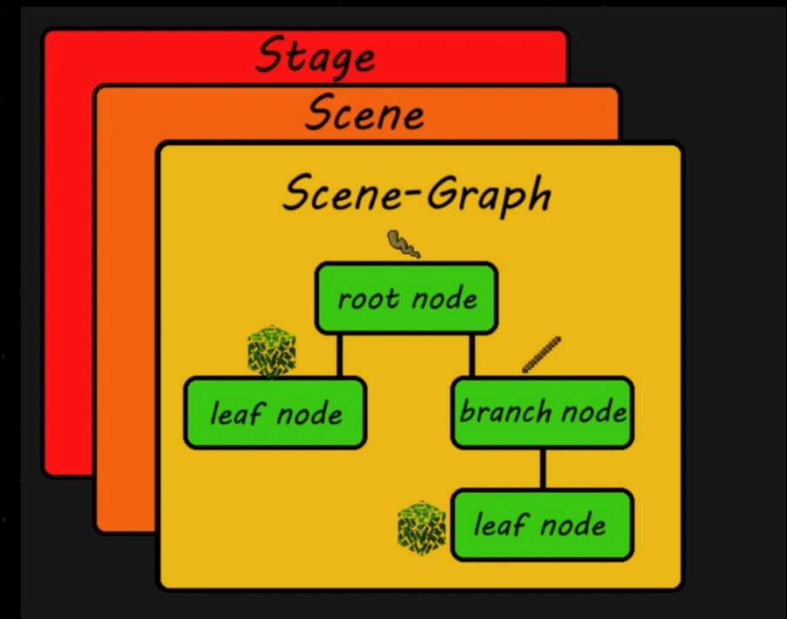
Layout: TextField

Code: TextField

Framework

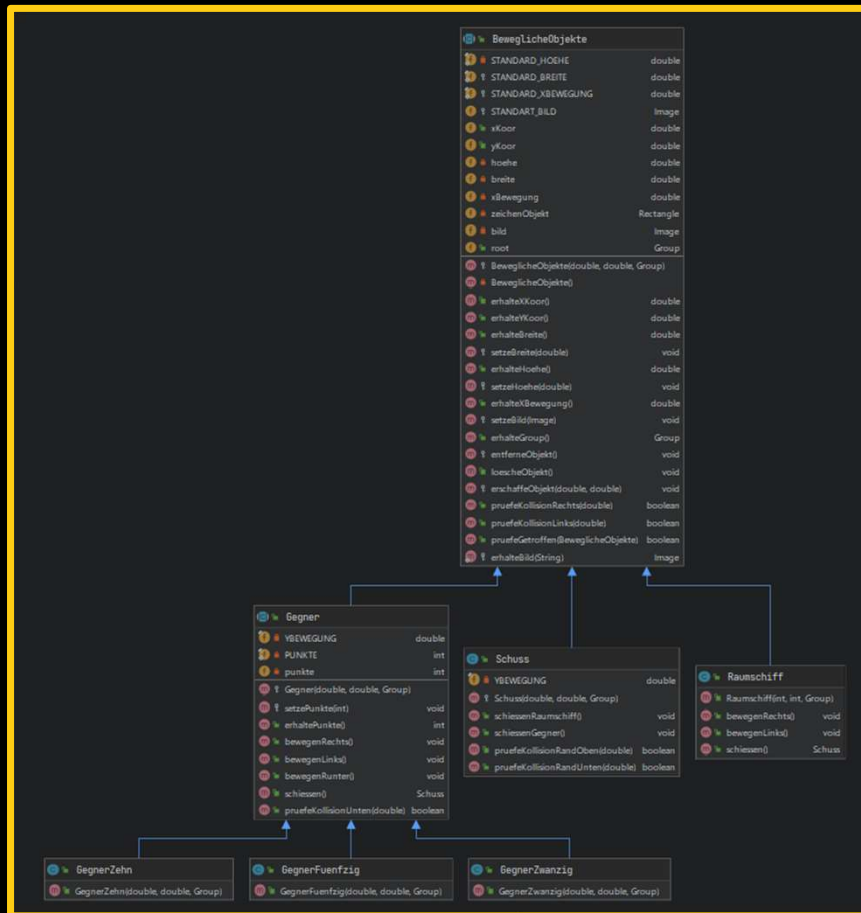


- Controllerklassen
- Import der FXML Dateien
- Zugriff auf GUI Elemente
 - Labels befüllen Werten
 - Tastatureingaben abfangen
 - Bewegen von Elementen
 - Bildwechsel durch Knöpfe
 - Formatieren von Schriften



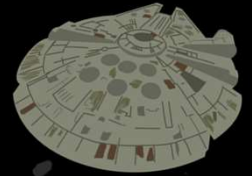


Klassen der beweglichen Objekte



- Generalisierungs-Spezialisierungs-Verbindung
- Oberklassen: BeweglicheObjekte und Gegner
- Unterklasse: Schuss, Raumschiff, Gegner

ObjektSteuerung



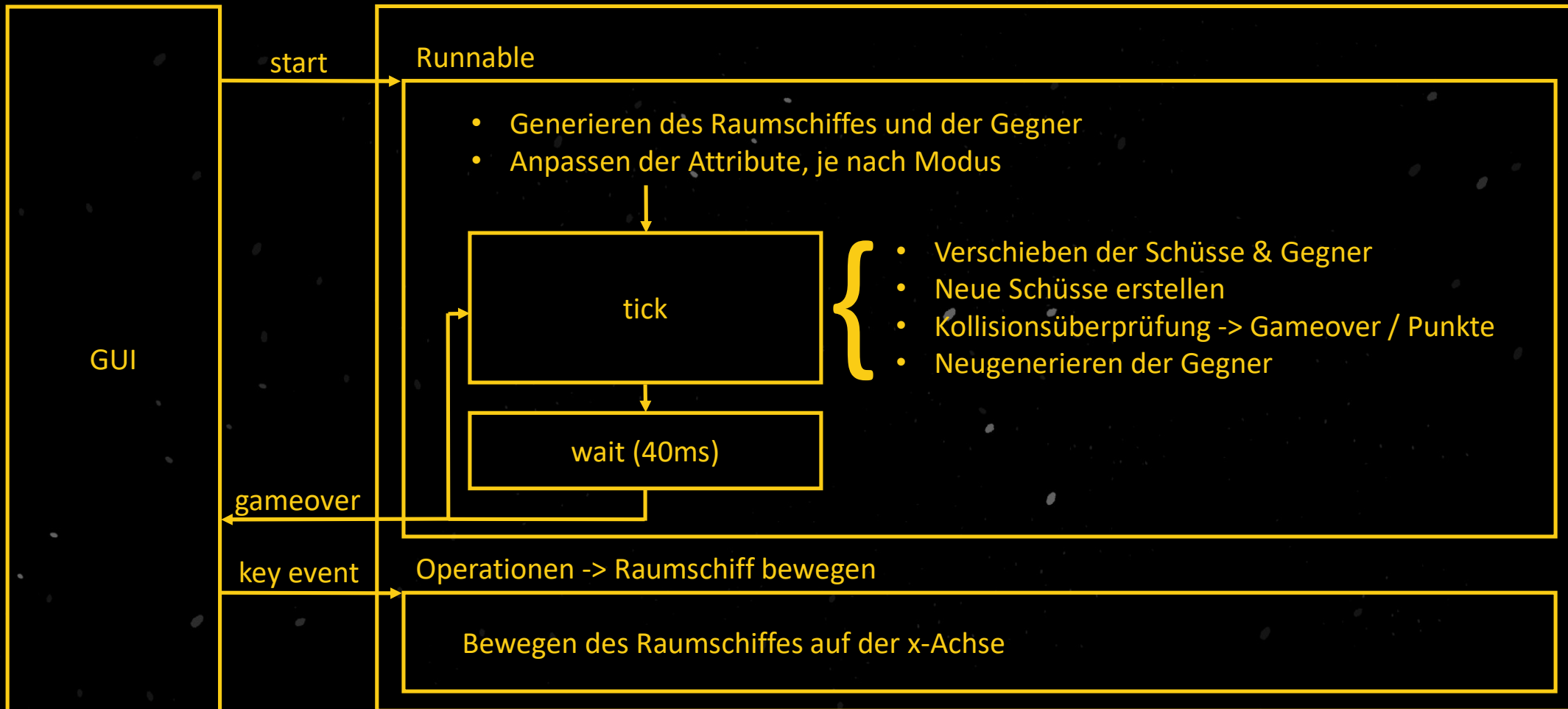
ObjektSteuerung		
richtung	xBewegung	
gegnerListe	ArrayList<Gegner>	
schuesseRaumschiff	ArrayList<Schuss>	
schuesseGegner	ArrayList<Schuss>	
RANDRECHTS	double	
RANDLINKS	double	
RANDUNTENGEGNER	double	
RANDUNTENSCHUSS	double	
RANDOBEN	double	
score	int	
ObjektSteuerung()		
erhalteScore()	int	
setzteRichtung(xBewegung)	void	
erhalteRandRechts()	double	
erhalteRandLinks()	double	
hinzufuegenSchussRaumschiff(Schuss)	void	
schuessenGegner(double)	void	
pruefeKollisionRand()	boolean	
bewegeGegner()	void	
bewegeSchussRaumschiff()	void	
bewegeSchussGegner(Raumschiff)	boolean	
neueWelleNotwendig()	boolean	
neueWelle(ArrayList<Gegner>)	void	
gameOver()	boolean	

- Verbindung zwischen Darstellung und Modell (MVP)
- Wird von der Klasse Spielablauf aufgerufen

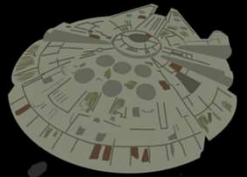
Spielablauf



Klasse Spielablauf (Thread)

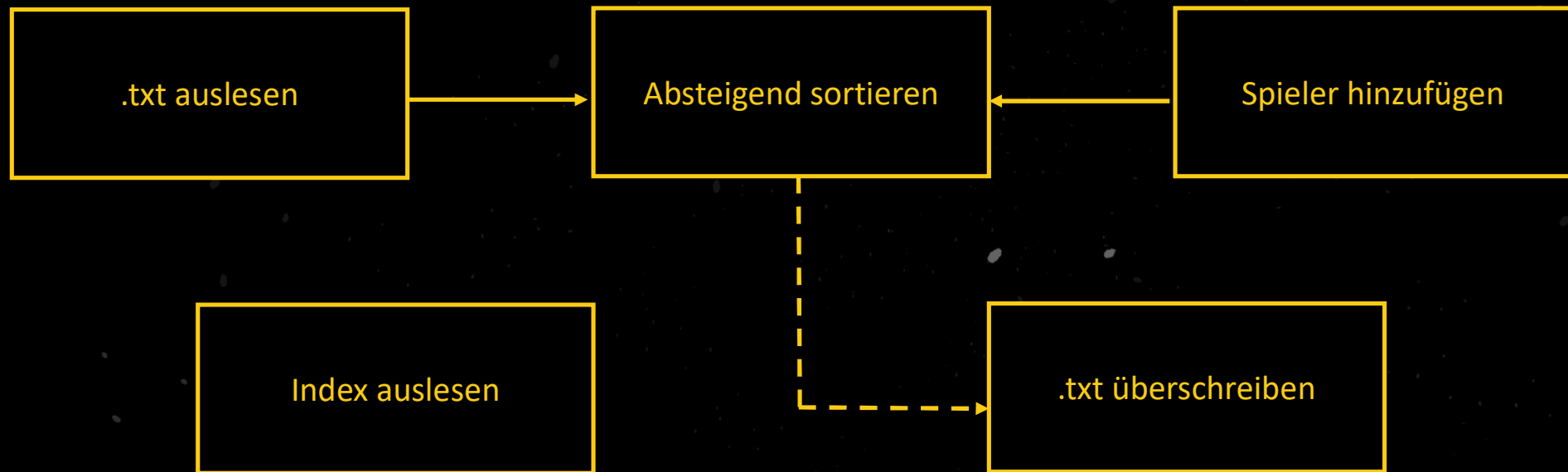
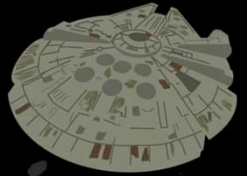


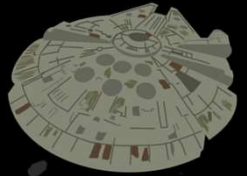
Spieler



- Spielername
- Erreichte Punktzahl
- [Spielername],[Punktzahl]

Punktliste





Herausforderungen

- Bewegliche Objekte auf Grafischen Flächen
- Spiel lief nicht flüssig
- Vergesslichkeit der Teammitglieder
- Benennung der Variablen und Methoden
- Programmierstil
- Deutsche Programmierung

Ergebnis



NORMAL **HÖLLE**
PATRICK **7050** **HIGHSCORE** **PATRICK** **690**

INVADING SPACE

PATRICK

NORMAL **HÖLLE**

10 **20** **50**

The image shows a selection screen for the game 'Invading Space'. At the top, it displays the player's name 'PATRICK' and a score of '7050' under the 'NORMAL' difficulty, and a 'HIGHSCORE' of '690' also for 'PATRICK' under the 'HÖLLE' difficulty. The title 'INVADING SPACE' is in the center. Below it is a button labeled 'PATRICK'. At the bottom, there are three difficulty levels: '10', '20', and '50', each with a corresponding ship icon.

HIGHSCORE: PATRICK 7050

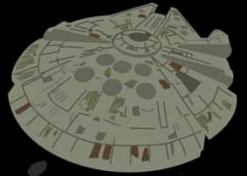
A formation of various spaceships arranged in four rows. The first row has 10 ships, the second and third rows have 12 ships each, and the fourth row has 10 ships. The ships are of different types, including fighters and larger vessels.

VERLASSEN

PATRICK

160

Ergebnis



HIGHSCORE: PATRICK 7050

WELLE 2 ERREICHT!

VERLASSEN PATRICK 1320

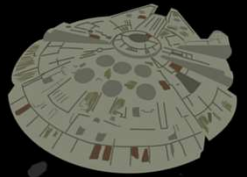
GAME OVER

PUNKTZAHL: 1320

TOP 3 SPIELER:

PATRICK	7050
JAN	2140
JASMIN	1410

ERNEUT SPIELEN **STARTSEITE**



Danke fürs Zuhören!

Habt ihr noch Fragen?