

Floatin' Around

Projektgruppe: Karim Debbabi & Michael Rummer

Klasse: 1AHIF

Jahr: 2025



Betreuer: Bechtold David, Walch Sarah, Breuss Katharina

Kurzbeschreibung:

Raft ist ein Survival-Game, bei dem man möglichst lange überleben sollte. Der Spieler startet auf einem kleinen Floß mitten auf dem Ozean mit nur einer Angel und einem Haken. Ziel des Spieles ist, mit diesen Werkzeugen, Treibende Gegenstände wie Holz, Seile, Leder und Fische zu fangen, um das Floß zu erweitern, und um am Leben zu bleiben. Doch nehmt euch in acht vor dem gefräßigen Hai.

1 Inhaltsverzeichnis

Inhalt

1	Inhaltsverzeichnis	2
2	Projektzeitplan.....	3
3	Lastenheft (Kurzbeschreibung, Funktionsumfang, Skizzen)	4
4	Pflichtenheft	6
4.1	Interner Programmaufbau (Programmlogik).....	6
4.2	Umsetzungsdetails	7
4.3	Ergebnisse, Interpretation (Tests).....	7
5	Anleitung.....	8
5.1	Installationsanleitung.....	8
5.2	Bedienungsanleitung	8
6	Bekannte Bugs, Probleme.....	9
7	Erweiterungsmöglichkeiten	10
8	Info.....	11

2 Projektzeitplan

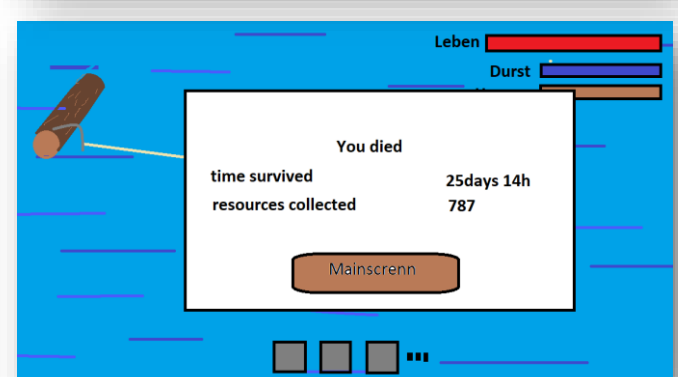
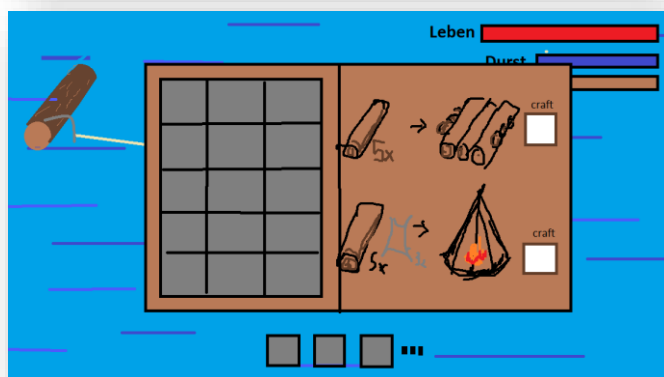
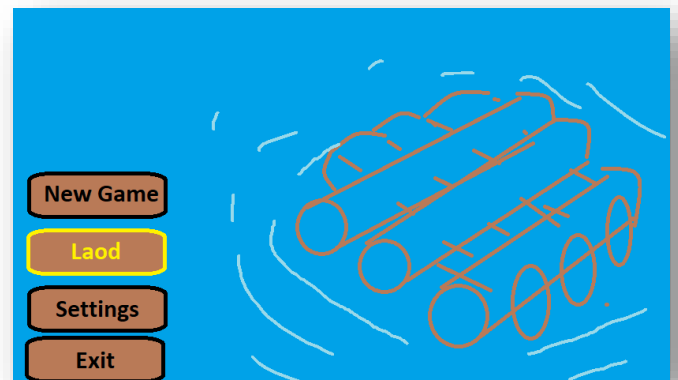
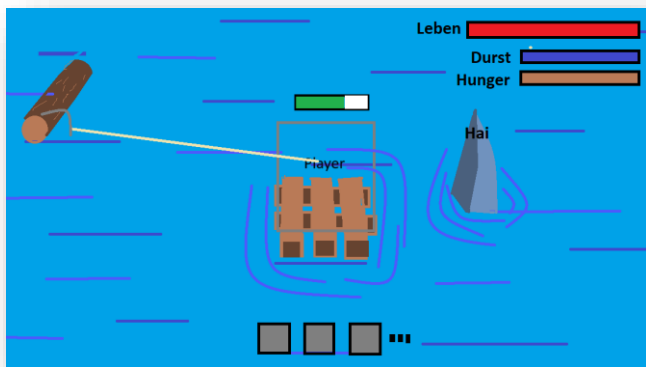
Datum	Aufgabe	Bearbeiter	Status (%)
12.05.2025	Startscreenfunktion	Michael	100%
12.05.2025	Wasser Erstellung	Karim	100%
15.05.2025	Player inkl. health/thirst/hunger Bar und Ideen für Hai	Michael	100%
16.05.2025	Raft teil platzieren	Karim	100%
23.05.2025	Objekte und Haken	Karim	100%
25.05.2025	Guide	Michael	100%
10.06.2025	Player inkl. Animationen	Michael	100%
12.05.2025	Hai	Michael	100%
15.06.2025	Inventar, Crafting, Item Auswahl	Karim	100%
16.05.2025	Speichern/Laden	Karim	0%

3 Lastenheft (Kurzbeschreibung, Funktionsumfang, Skizzen)

2.1. Kurzbeschreibung

Der Spieler startet auf einem kleinen Floß mitten im Ozean. Ziel ist es, so lange wie möglich zu überleben, indem man treibende Gegenstände aus dem Wasser sammelt, das Floß vergrößert und nützliche Gegenstände herstellt. Dabei muss man auf Hunger, Durst und Gefahren wie einen angreifenden Hai achten. Durch geschicktes Ressourcen-Management und geschicktes Bauen entwickelt sich das einfache Floß zu einer schwimmenden Überlebensbasis 😊.

2.2. Skizzen



2.25 Sc

2.3. Funktionsumfang

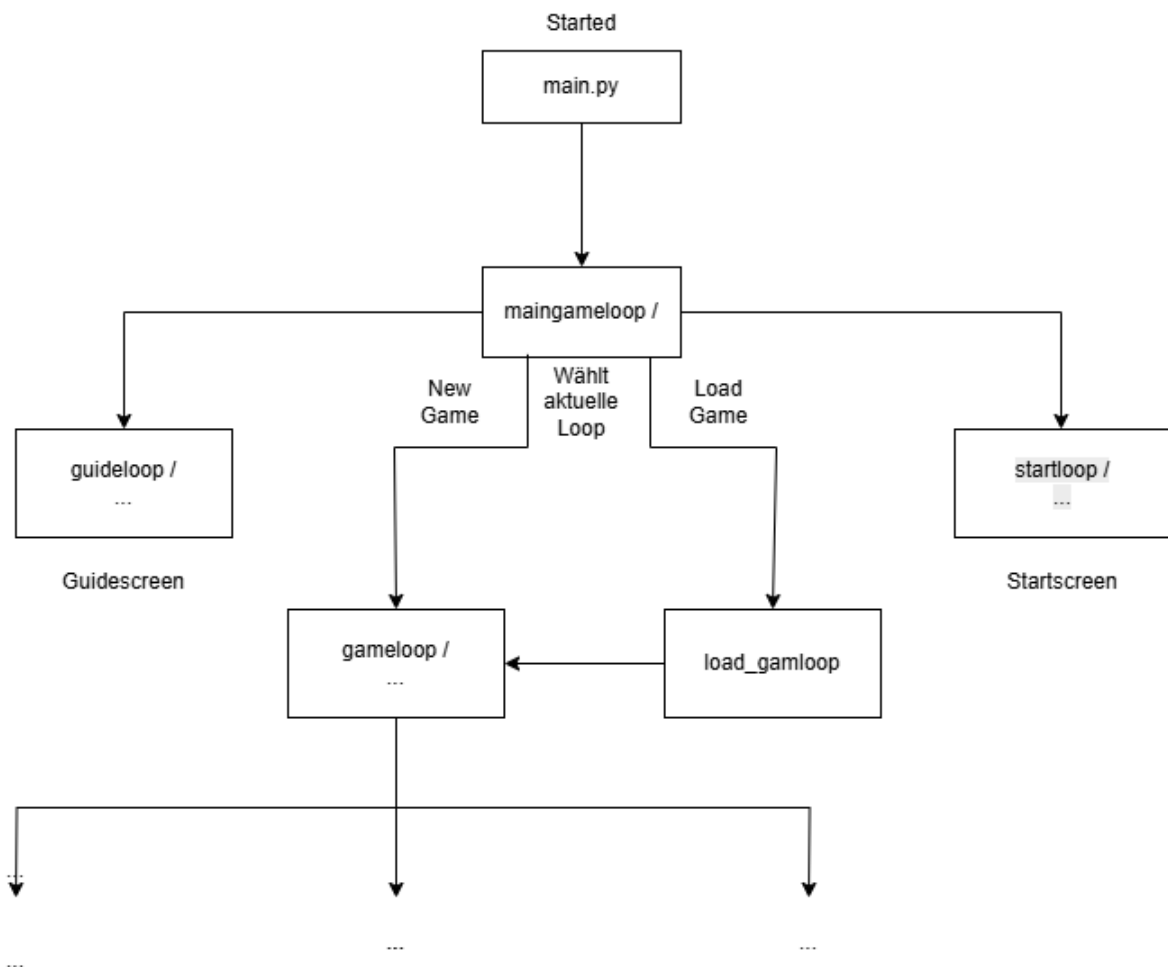
Alle Funktionen genau erklären.

Must-Haves und Nice-To-Haves beschreiben (Punkteliste). Must-Haves müssen umgesetzt werden.

Must have	Nice have
Startmenü (Start-Knopf, Laden-Knopf, Verlassen-Knopf)	Wetter (Stürme, Regen, Sonne)
End Menü	Tag/Nacht Zyklus (Jahreszeiten)
Floß (mit Vergrößerung) (nicht bewegbar)	Rucksack
1 funktionierender Spieler (mit Auswerfhaken)	achievements
Bewegendes Wasser mit Objekten wie Holz, Plastik, Stoff	
Hai, der um das Floß schwimmt und Spieler angreift	
Gesundheits- und Nahrungs Leiste	
Crafting von Gegenständen wie Waffen, Kleidung, Möbel (Lagerfeuer)	
Fishing Mechanik (Angel)	

4 Pflichtenheft

4.1 Interner Programmaufbau (Programmlogik)



Aufbau:

Main.py öffnet Maingameloop, welche alle Loops innerhalb vom Programm managend. Wenn das Spiel gestartet wird, wird die „Startscreenloop“ ausgeführt das Start Menü, wenn man nun auf einer dem Knopfe drückt, wird die Loop auf die dementsprechende Loop abgeändert.

Gameloop: Hauptspielschleife die die Unterordner öffnet für deren Funktionen je nach Hand Item oder Aktion

Laden: Kopie von Gameloop mit Lade-Veränderungen

Guide: Guideloop (eigenes File)

Exit: Beendet das Programm

4.2 Umsetzungsdetails

Maingameloop:

unterteilt das gesamte Spiel in Unter-Loops für bessere Ordnung.

Umsetzung vom Raft- & Building-Layer:

Zwei Matrizen speichern die platzierten Tiles. Diese zeigen auch die Hover-Funktion an: Wenn man mit der Maus über ein Teil hovers, wird angezeigt, ob es dort platziert werden kann oder nicht.

Hook:

Je nach Maus-Entfernung wird der Haken unterschiedlich weit ausgeworfen. Dies wird mit einem Balken visualisiert.

Fishing:

funktioniert ähnlich wie der Haken, hat jedoch eine bestimmte Wahrscheinlichkeit, einen Fisch zu fangen – abhängig von der Balkenlänge bzw. Kraft.

Hai:

greift nach einer zufälligen Zeit die Mitte an und spawnet dann zurück.

Startscreen:

überprüft die Button-Position in Bezug auf die Mausposition.

Pause- / Deathscreen:

wird mit einem if-Statement gesteuert.

Speichern & Laden:

speichert und lädt die Maps.

4.3 Ergebnisse, Interpretation (Tests)

Das Spiel läuft flüssig, außer wenn der Spieler im Inventar ist, weil dann eine große Fläche ausgegeben wird. Auch weil nicht alle blits auf den Screen angepasst sind braucht das Starten etwas länger

Schwachstellen

- Inventar
- Hintergrund Bild würde Lagen, wenn er verändert, wird keine Ahnung warum...
- Hai Attacke könnte ausgebaut werden

5 Anleitung

5.1 Installationsanleitung

pip install pygame, python

5.2 Bedienungsanleitung

Movement mit W/A/S/D, mit der Maus kann man unterschiedliche Sachen machen:

Wenn man mit der Leeren Hand auf ein Lagerfeuer klickt, kommt man in das Menü davon

Wenn der Hook ausgewählt ist, kann man diesen auswerfen,

Wenn man ein Raft Tail in der Hand hat,

Wenn man die Lederflasche in der Hand hat, kann man trinken mit links Klick,

Wenn man einen Fisch rechts klickt, isst man diesen

Inventar:

Öffnen mit Inventar mit links click und hold Items verschieben. Wenn man es in der Luft absetzt, dann einfach nochmal im Feld links klicken

Items zum weiter Crafting müssen in den 10 unteren Inventar Slots sein

6 Bekannte Bugs, Probleme

Merge Conflicts: Es gab unterandere Probleme beim Verbinden unserer verschiedenen Projekte via Git-Hub. Darunter zahlreiche Bugs und Errors. Außerdem wurden Bilder nicht immer übertragen was dazu geführt hat das einer den anderen immer um die Bilder schicken musste

Inventar: Das Inventar hat zahl reiche Fehler ausgelöst, da Leer-Items entstanden, welche man nicht bewegen konnte, außer dem die Click Hit box bei Crafting hat viele Fehler erbracht, weil diese manuell eingefügt worden mussten

Hintergrundbild: Lagt Ultra Krass, wenn ich das Hintergrund Bild um Pixel will. Ka warum, deswegen Pixel Fehler im Hintergrund

7 Erweiterungsmöglichkeiten

Tag/Nacht Zyklus, Wetter, Speichern/Laden, mehr Objekte

8 Info

- Der Zeitplan ist wöchentlich auszufüllen!
- Die Titelseite ist am Projektende auszudrucken („Hall of Fame“ „Hall of Shame“ 😊)
Sie muss auf **einer** Seite Platz finden.
- Endabgabe: **dieses** Dokument ausgedruckt, Projektverzeichnis per Teams abgeben
- Projektbenotung: Neben dem Endprodukt werden vor allem **der Projektfortgang, die Arbeitsweise und die Termintreue** benotet (keine Projekte, die in der letzten Nacht fertiggestellt werden!) Der Code soll möglichst übersichtlich gehalten werden (Einsatz von **Funktionen und Klassen**).

Viel Spaß und happy coding!

