

Jump and Hook

Renne, springe und hangle dich zur nächsten Plattform. Benutze den Greifhaken, um dich zur nächsten Plattform zu ziehen, Objekte für Minigames aufzuheben oder aus dem Weg zu räumen.

- [Pages-Version](#)
- [Repository](#)
- [Zum den Designdokument als PDF](#), sonst auch hier in der Readme.md
- [Zu den Skripten](#), und [main.ts](#)
- [Zip-Version herunterladen](#)

Anleitung zum Starten

- Repository clonen und index.html über einen Live-Server starten
- Pages-Version öffnen
- Play drücken :)

Steuerung

- WASD: Bewegt den Avatar
- Maus: Kamerasteuerung(First-Person), um auf Objekte zu zielen
- Linke Maustaste: Objekte mit dem Haken wegstoßen
- Rechte Maustaste: Zieht den Avatar zu statischen objekten oder dynamische Objekte zum Avatar
- E: Nahe Objekte aufheben/fallenlassen
- Leertaste: Springen
- Shift: Sprinten

Den Greifhaken während des Sprintens im Sprung zu verwenden vereinfacht das Erreichen der nächsten Plattform. Der Greifhaken trifft nur Objekte in der Mitte des Bildschirms in einem bestimmtem Radius.

Checkliste für Leistungsnachweis

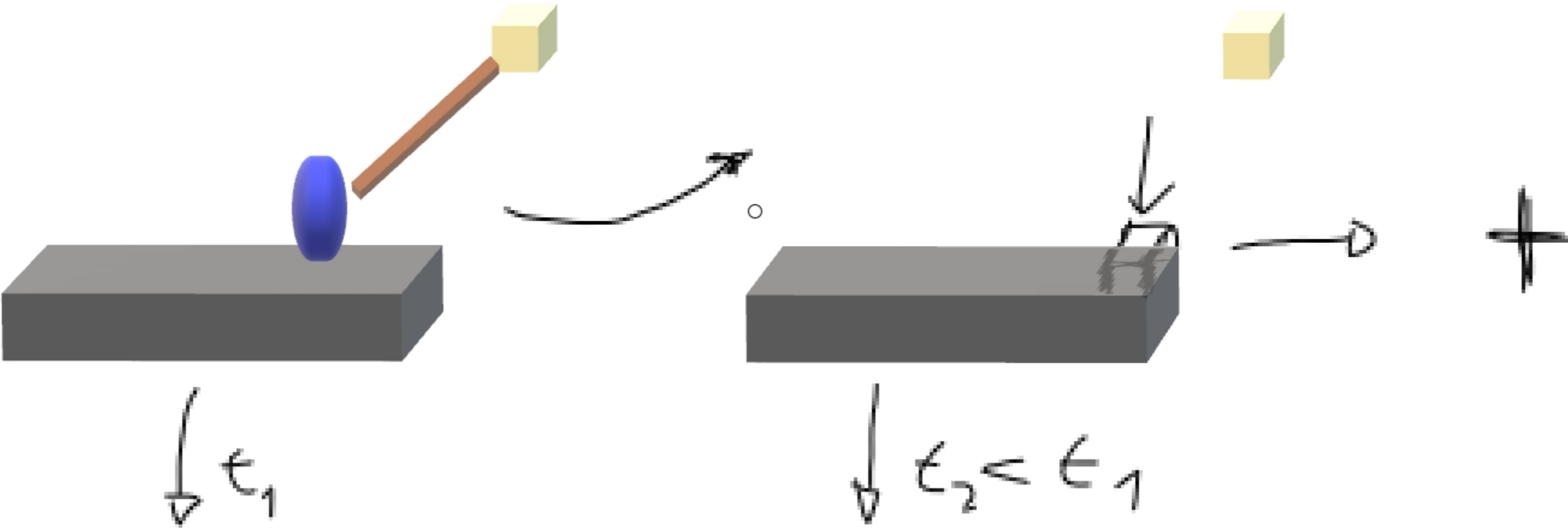
© Fabian Kowatsch, HFU

Nr	Bezeichnung	Inhalt
	Titel	Jump and Hook
	Name	Fabian Kowatsch
	Matrikelnummer	263775
1	Nutzerinteraktion	Der Nutzer kann mithilfe von einer Maus und einem Keyboard mit der Applikation interagieren. Der Avatar lässt sich mit WASD bewegen und die Kamera lässt sich mit der Maus rotieren(First-Person). Mit der linken Maustaste kann man Objekte (auch aufgehobene) mit dem Greifhaken wegstoßen. Mit der rechten Maustaste kann man sich zu statischen Objekten heranziehen oder dynamische Objekte an sich heranziehen. Mit E kann man Gegenstände aufheben und fallenlassen. Mit der Leertaste kann man springen und mit Shift sprinten.
2	Objektinteraktion	Der Avatar interagiert hauptsächlich mithilfe des Greifhakens mit Objekten (S.O.). So kann er sich von Plattform zu Plattform schwingen oder Objekte für Minigames aufheben und abstoßen, sowie Hindernisse aus dem Weg räumen.
3	Objektanzahl variabel	Mit dem Erreichen des Endes einer Plattform wird, sofern die nötigen Hindernisse oder Minigames überwunden wurden, eine neue Plattform erzeugt. Diese kann eine von 4 möglichen Hindernissen haben, manche davon mit variabler Anzahl.
4	Szenenhierarchie	Alle Plattformen sowie der Avatar sind Kinder eines einzelnen Knoten. An der Plattform ist jeweils ein Knoten, über den man sich zur nächsten Schwingen kann, sowie ein Knoten, an dem alle Hindernisse hängen. Der Avatar hat einen Kameraknoten. Am Kameraknoten ist auch der Knoten für den Greifhaken sowie viele Audioknoten befestigt. Der Knoten für den Greifhaken hat sowohl Audioknoten als auch Knoten für die jeweiligen Meshes als Kinder.
5	Sound	Es gibt eine Hintergrundmusik sowie ein atmosphärisches Hintergrundgeräusch. Zudem wird bei jeder Kollision ein Sound abgespielt. Beim Benutzen des Greifhakens mit linker oder rechter Maustaste werden jeweils die passenden Sounds abgespielt. Beim Bestehen des Ball-Minispiels erfolgt ein Ton zur Bestätigung.
6	GUI	Im GUI ist während des Spielens die Anzahl der abgeschlossenen Plattformen und die verbleibende Zeit für die aktuelle Plattform zu sehen. Vor dem Programmstart kann man im Menü das Spiel über eine Taste starten und die Soundlautstärke mit einem Schieberegler einstellen. Außerdem sind der lokale Highscore und der Score vom letzten Versuch zu sehen.
7	Externe Daten	Die Spielergeschwindigkeit, die Sprungstärke sowie andere Kräfte, die Zeit bis die erste Plattform sinkt, die Zeit, die pro bestandener Plattform von der Anfangszeit abgezogen wird sowie Einstellungen zur Musik oder zum Debug-Modus werden in einer externen Datei config.json gespeichert. Einige Daten werden zudem aus der Scene.json geladen.
8	Skriptkomponenten	Alle zur Plattform gehörenden Objekte haben ihre Skriptkomponente, z.B. die Plattform an sich, das Ballminigame, die sich an der y-Achse drehende Falle sowie weitere Hindernisse und Gegenstände.
9	Klassen	Der Avatar und der Greifhaken werden von f.Node abgeleitet, um einfacher miteinander kommunizieren zu können ohne umständliche Verweise. Zudem gibt es weitere Klassen für das UI, Utility oder das Game an sich, welches in der main.ts erstellt wird.

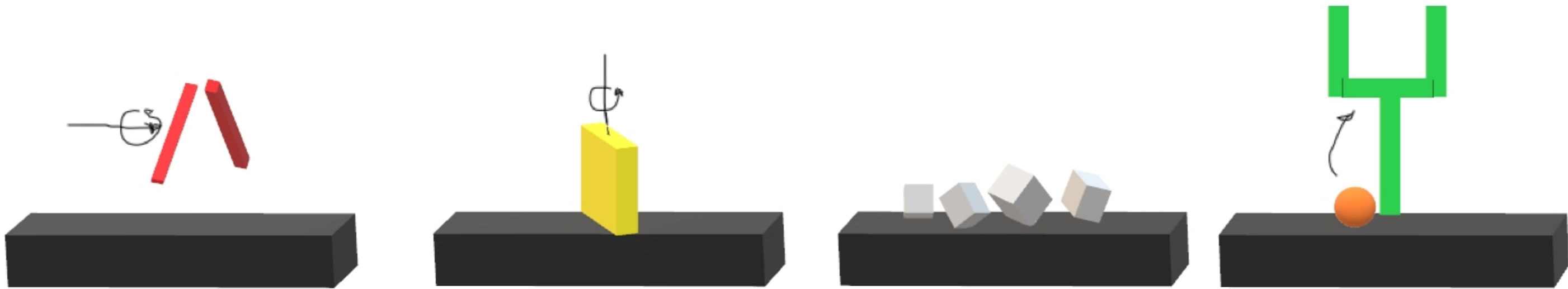
Nr	Bezeichnung	Inhalt
10	Maße & Positionen	Der Avatar startet an $Z = X = 0$ und ist 1 Einheiten hoch, von ihm gehen alle anderen Größen aus. Die erste Plattform befindet sich am Ursprung.
11	Event-System	Es werden hauptsächlich Physic-Events genutzt, um Kollisionen festzustellen oder das Erstellen neuer Plattformen durch das betreten eines Triggers auszulösen. Zudem hören alle Skriptkomponenten darauf, ob sie an einen Knoten gehftetet werden, da erst dann auf den Knoten zugegriffen werden kann. Außerdem gibt es ein eigenes Event, das beim bestehen des Minigames gefeuert wird und das spawnen einer neuen Plattform erlaubt.

Skizzen

Spielprinzip



Hindernisse



Szenenhierarchie

