

# Heistrunner

**“Heistrunner” - ist ein First-Person Jump ‘n’ Run Game mit Action Elementen, bei dem die Spieler:innen Hürden durch Parkour-Tricks überwinden sollen.**

by Matteo Bentivegna

## Inhaltsverzeichnis

- [1. Game Design](#)
  - [1.1 Game Design Übersicht](#)
  - [1.2 Spielkonzept](#)
  - [1.3 Game Design Schema](#)
- [2. Technische Dokumentation](#)
  - [2.1 Technische Design Übersicht](#)
  - [3. Artwork](#)
    - [3.1 Themen-Recherche und Art-Bible](#)
    - [3.2 Konzept für Modelle und Animationen](#)
    - [Assets](#)

# **1. Game Design**

## **1.1 Game Design Übersicht**

Die Spieler:innen tauchen in die Rolle eines weltberühmten Diebes mit dem Spitznamen Joker. Er verfügt über übernatürliche akrobatische Fähigkeiten. Sein aktueller Auftrag führt ihn in die Metropole Safe City. In der Mitte der Stadt befindet sich ein riesiger Tresor, der Megasafe, welcher einen sehr seltenen Schatz beinhaltet. Diesen Schatz soll er für seinen Boss stehlen und sicher aus der Stadt bringen.

Bevor Joker den Heist startet, muss er jedoch noch einmal in einer Simulation trainieren. In der Simulation findet er Anweisungen, welche Skills er benötigt, um sein Ziel zu erreichen.

Joker startet in der Nähe des Tresors, auf dem Dach eines Hochhauses. Von dort aus muss er einen Weg weiter hoch finden. Der Tresor ist das größte Gebäude in der Stadt, was den Heist um einiges erschwert. Es finden sich jedoch ein paar Kräne, die einen Weg direkt zum Schloss des Tresors führen.

Auf dem ersten Gebäude findet Joker horizontal liegende Rohre, an denen er mit einem simplen Sprung vorbeikommt. Zwischen dem aktuellen Gebäude und dem nächsten ist eine relativ große Lücke. Diese kann Joker einfacher mit Hilfe von zwei Boxen überqueren, indem er einen höheren Startpunkt hat.

Das zweite Gebäude scheint auf den ersten Blick nichts besonderes zu sein. Doch auch hier findet sich ein Rohr, das einen kleinen Spalt zwischen Rohr und Dachboden hat. Groß genug für Joker, um darunter hindurch zu rutschen. Wenn Joker im Lauf anfängt zu rutschen, gewinnt er an Geschwindigkeit. Diese Geschwindigkeit kann nützlich sein, um noch größere Lücken zwischen den Hochhäusern zu überqueren.

Dies ist nun auch der Fall. Jedoch wird es mit einfachen Rutschen nicht möglich sein, diesen Spalt hinter sich zu lassen.

Da kommt nun die Dachschräge des Gebäudes ins Spiel. Rutscht Joker diese Dachschräge hinunter, wird er deutlich schneller und schafft damit den Sprung.

Zwei Hochhäuser stehen sich relativ nah. An den Wänden der Häuser befinden sich Werbetafeln, die Joker mit Hilfe eines Wandsprints nutzen kann, um auf die andere Seite zu kommen. Springt er zwischen den Werbetafeln hin und her, fällt er auch nicht in den Abgrund. Auf der anderen Seite angekommen findet man dieses Mal Kanten, die man zum Klettern auf dem Dach der Gebäude nutzen kann.

Oben angekommen, hängen Teile von Werbeplakaten an einen Kran. Alleine reichen diese jedoch nicht aus, um auf das nächste Gebäude, den Wolkenkratzer, zu kommen. An der Kante des Wolkenkratzers befindet sich ein Ring, an dem sich Joker mit seinem Wurfhaken hochziehen kann, wenn er den Sprung von den Werbeplakaten wagt.

Vom Wolkenkratzer aus kommt er nur über mehrere Objekte, die an zwei Kränen hängen an das Schloss des Tresors. Neben Ringen, Werbeplakaten und Kanten findet man nun auch Rohre, durch die man hindurch rutschen kann. Hier kommen alle Mechaniken, die Joker vorher gelernt hat, zusammen und bieten damit die letzte Herausforderung.

Am Ende angekommen, findet man eine goldene Glasflasche Kakao. Diese eingesammelt, beendet die Simulation und damit auch das Spiel.

## 1.2 Spielkonzept

*HINWEIS: Einige Interaktionen und Mechaniken wurden noch nicht zu Ende implementiert und werden in der jetzigen Iteration nicht vorhanden sein, sind aber für die Zukunft geplant. Diese wurden über das gesamte Dokument mit dem Kürzel **WIP(Work In Progress)** markiert.*

Ziel des Spiels ist es, den Tresor zu infiltrieren und den Schatz erfolgreich zu stehlen. In der ersten Iteration wird nur das “Tutorial” implementiert sein.

Die Szene ist minimalistisch aufgebaut. Sie beinhaltet an ausgewählten Stellen und Objekten kontrastreiche Farbmarkierungen, um die Spieler:innen zum Ziel zu leiten und um interagierbare Objekte zu markieren.

Zu interaktiven Objekten gehören:

**Werbeplakate**, an denen man entlang rennen kann.

**Kanten (an Wänden)**, die zum Klettern da sind.

**Senkrechte Rohre**, die denselben Nutzen wie Werbeplakate haben.

**Ringe**, die mit dem Wurfhaken abgeworfen werden können, um sich zu ihnen hochzuziehen.

**Rutschen**, zum Rutschen, um Geschwindigkeit aufzubauen.

**Schätze**, die eingesammelt werden können.

(**WIP: Power Ups**, die ein Fähigkeits-Upgrade geben.)

(**WIP: Leiter/Regenrinne**, die das senkrechte Klettern ermöglichen.)

(**WIP: Seile**, fürs senkrechte Klettern und Schwingen.)

(**WIP: patrouillierende Gegner**, die einen angreifen können.)

Die Hindernisse in den Level können hauptsächlich durch den üblichen Aktionen der Spieler:innen überwunden werden. Neben dem einfachen Laufen, Sprinten und Springen, auch die komplexeren Wand-Sprints, Slides und Widerhaken-Würfe. Diese müssen im späteren Spielverlauf miteinander kombiniert werden und verlangen schnelle Reaktionszeiten.

Die Hindernisse verlangen also, dass man sich alles zuvor gelernte merkt und geschickt im richtigen Moment anwendet.

Der Spielfortschritt wird durch die Umgebung dargestellt. Neben dem erfolgreichen Passieren der Hindernisse gibt es auch Objekte, die eingesammelt werden können, um ein Level abzuschließen oder ein Teil-Level freizuschalten.

## 1.3 Game Design Schema

Das erste Gebäude dient dazu, das Springen beizubringen, da dies eine der fundamentalsten Spielerbewegungen ist.

Auf dem zweiten Gebäude lernt man die erste komplexere Spielmechanik, das Sliden. Diese wird auch direkt mit dem Springen kombiniert. Das Sliden wird mit der Farbe Grün signalisiert.

Die zweite komplexere Spielmechanik wird auf dem 3. Gebäude beigebracht. Hier erkennt man, wie schon oben erwähnt, dass das Springen fundamental ist und in allen Bewegungen integriert wurde. Der Wand-Sprint wird nur an ausgewählten "Wänden" erlaubt. Diese werden Rot markiert.

Um auf das Dach der Hochhäuser zu kommen, muss man von Kante zu Kante springen/klettern. Diese Kanten befinden sich abwechselnd auf dem einen und dem anderen Hochhaus und haben die Farbe Orange.

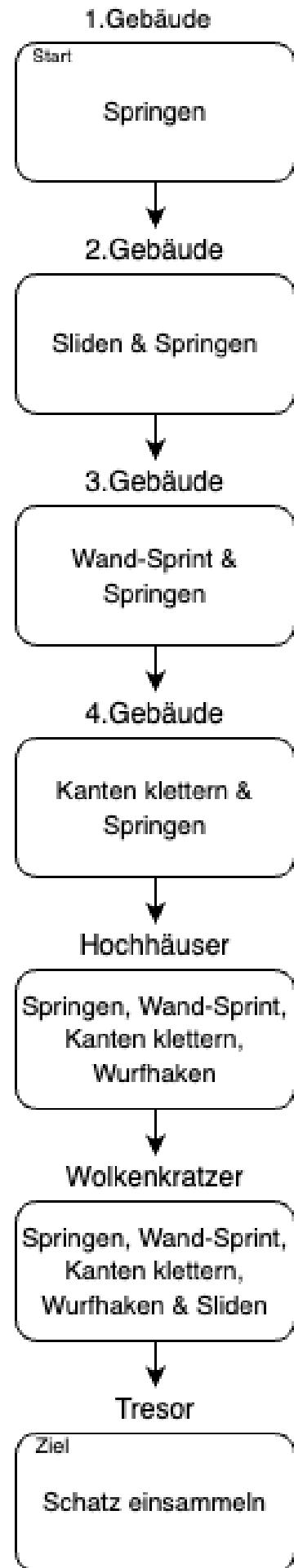
Nun lernt man die letzte komplexe Mechanik kennen. Um diese anwenden zu können, braucht man den Wand-Sprint, das Kanten-

Klettern und wieder das Springen. Erst wenn man alles hintereinander absolviert hat, kann man den Wurfhaken einsetzen. Das liegt daran, dass sich der "Ring" an dem sich der Wurfhaken festhaken kann, nur ab einer bestimmten Entfernung erreicht werden kann. Diese Ringe haben die Farbe Blau. Der UI-Cursor wechselt die Farbe von Weiß auf Grün, um zu signalisieren, dass man nah genug am Ring ist, um ihn zu erreichen.

Auf dem Wolkenkratzer angekommen, muss man alle bisher gelernten Mechaniken einsetzen, um an das Ziel zu kommen.

Dabei muss man einen langen Wand-Sprint an einen anderen Wolkenkratzer hinterlegen, um in Reichweite für einen Wurfhaken-Ring zu kommen. Dieser zieht einen an ein grünes Rohr, das an einem Kran hängt.

Das grüne Rohr gibt einem genug Momentum, um in die Reichweite eines weiteren Wurfhaken-Rings zu kommen, welcher die Spieler:innen zu einer Kante zieht. Von dort aus klettert man über zwei Kanten weiter bis man auf die Höhe mehrerer hängender Wände kommt. Mit einem Sprint gelangt man an das letzte Objekt des Krans. Ein rotes, senkrecht hängendes Rohr.



Springt man vom Rohr ab, reicht die Reichweite aus, um den letzten Wurfhaken-Ring des Levels zu erreichen. Dieser bringt einen auf die Plattform des Tresors, wo sich der Schatz befindet.

## **2. Technische Dokumentation**

### **2.1 Technische Design Übersicht**

Die wichtigsten Systemkomponenten sind die PlayerMovement Komponente, sowie die PlayerCamera, PlayerWallRunning, PlayerGrapple, PlayerLedge, PlayerRespawn Komponente.

Die PlayerMovement Komponente beinhaltet einen State-Manager, welcher die einzelnen States der Player Komponente setzt. Die allgemeine Geschwindigkeit, Gravitation, Kollision der Player Komponente wird in dem Skript behandelt.

Die PlayerCamera Komponente behandelt neben der einfachen Kamera Steuerung des Players auch die Field Of View Anpassung und die Ausrichtung der Kamera bei Wandsprints (Player Wall Running Script).

In den restlichen oben genannten Komponenten, existieren Sub-State-Manager, diese können den State-Manager anstoßen und den State dadurch aktualisieren.

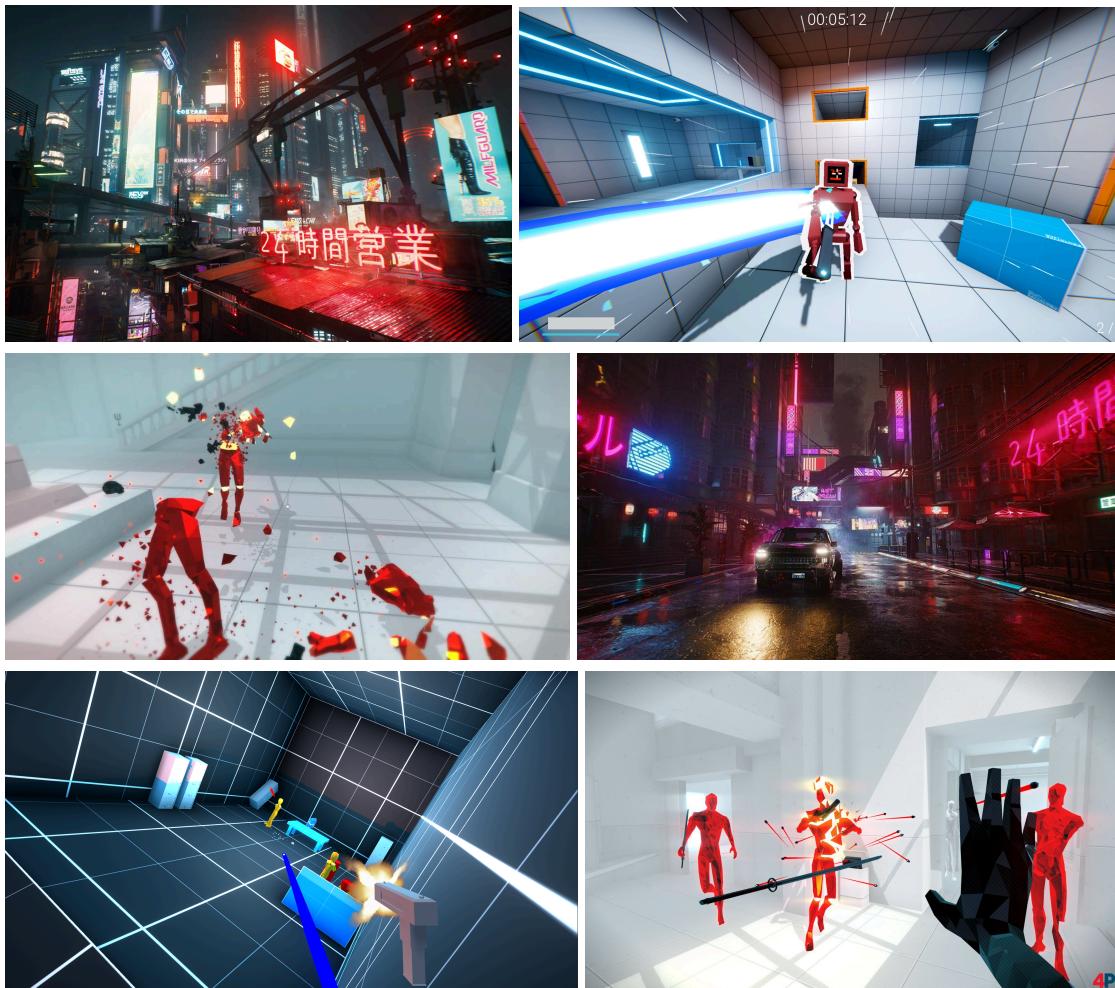
Das PlayerRespawn Script stoppt das Spiel und wartet auf den Input der Spieler:innen. Wird das Spiel fortgesetzt, setzt das Skript die Position der PlayerKomponente auf eine festgelegte Position. Die SpawnPosition wird im Laufe des Spiels immer wieder neu gesetzt, um eine Art Fortschritt zu simulieren.

Die PlayerGrapple Komponente stößt den Line Renderer an und übergibt ihm die Start- und Endposition, damit eine Linie gezogen werden kann, um einen Wurfhaken zu simulieren.

## 3. Artwork

### 3.1 Themen-Recherche und Art-Bible

Das Artwork des Spiels soll einen Neo-Retro-Look haben. Die Modelle werden mit wenigen Polygonen modelliert und mit kontrastreichen Farben versehen. Was gut anhand des Spiels "Karlsson" (siehe [https://www.youtube.com/watch?v=\\_npGfPDfByw](https://www.youtube.com/watch?v=_npGfPDfByw)) dargestellt werden kann.



Auf dem Moodboard sind Bilder aus den Spielen Karlsson, Superhot und Cyberpunk zu sehen

Fokus liegt auf den fluiden und schnellen Spielerbewegungen. Mit Hilfe der kontrastreichen Objekte sollen den Spieler:innen die wichtigsten Informationen in kürzester Zeit übermittelt werden, ohne dass sie von zu vielen Details abgelenkt werden. Hierfür kann man "Mirrors Edge" als gutes Beispiel ziehen (siehe <https://www.youtube.com/watch?v=2N1TJP1cxmo>).

## 3.2 Konzept für Modelle und Animationen

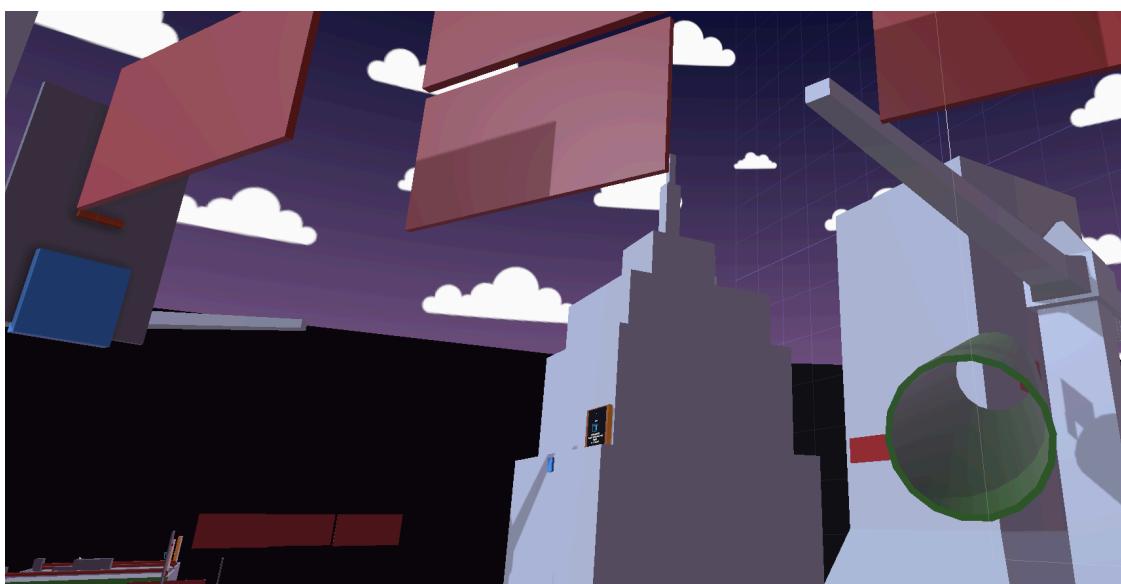
### Modelle und Umgebung

Die Modelle, mit denen die Spieler:innen interagieren beschränken sich größtenteils auf primitive Objekte wie Würfel, Zylinder und Ebenen.

Für die Stadt wurden jedoch ein paar Komplexere Objekte modelliert.



Der Tresor (Megasafe) soll herausstechen und sofort die Aufmerksamkeit auf sich ziehen. Dies wird durch seine massive Größe und den goldenen Glanz erreicht. Die Gebäude um ihn herum werden zwar vom Spieler genutzt, um sich dem Ziel zu nähern, sollen jedoch keinen großen Fokus bekommen.



Die Objekte, mit denen die Spieler:innen interagieren können, kriegen kontrastreiche Farben, um die grauen Gebäude zu komplementieren.

Für Animationen gibt es lediglich die “FadeIn” und “FadeOut” für Szenenwechsel.

## Assets

### Fonts

[Expose-Regular SDF](#)

[Persona5MenuFontPrototype-Regular](#)

### Asset Store

SimpleCumulus by [Borodar](#)

LeanTween by [Dented Pixel](#)

Gridbox Prototype Materials by [Ciathyza](#)