

# Jump 'n' Run – analog 'n' digital: Gemeinsam Hindernisse Überwinden

Materialien zur Online-Fachkonferenz „Digitalisierung als Chance:  
Milieuübergreifende Bildungskonzepte für kulturelle Teilhabe“

13. 1. 2021

## Das Projekt

<https://www.bremer-medienpaedagogik.de/ruder-blog.html>

Der Held macht seinen Weg laufend und springend durch eine mit Hindernissen gepflasterte Welt. Schafft er ein Hindernis nicht auf Anhieb, probiert er es einfach so lange, bis es klappt. Diesem uralten Muster folgen Jump 'n' Run Spiele und Parkour als digitale und analoge Variante. Beiden liegt eine resiliente und integrative Lebenseinstellung zugrunde: *Jeder fängt klein an; nach und nach traue ich mir immer mehr zu; es ist okay, wenn ich etwas nicht auf Anhieb schaffe, dann probiere ich es einfach nochmal; Menschen können Sachen unterschiedlich gut.* Die beiden Varianten gegenüberzustellen, aber auch immer wieder zu schauen, wie sie sich ergänzen können, war Grundidee dieses Projektes.

Das Projekt wurde 2019 mit dem Medienkompetenzpreis „Das Ruder“ ausgezeichnet und von der Bremischen Landesmedienanstalt gefördert (<https://www.bremische-landesmedienanstalt.de/medienkompetenz-preis>). Es umfasste einen Workshop in den Herbstferien und wöchentliche Termine bis zu den Weihnachtsferien. Die Termine haben im Digital Impact Lab (<https://impact-lab.eu/>), einem experimentellen Lernort für digitale Aktivitäten, stattgefunden, das als Projektpartner auch einen Praktikanten zur Unterstützung abgestellt hat. Die 9 Jugendlichen haben meist in Zweierteams gearbeitet, genutzt wurden Bloxels und Scratch zur Gestaltung der eigenen Games, wofür 3 iPads (TB) und mehrere Desktop-PCs des Labs zur Verfügung standen. Ein professioneller Parkourtrainer hat die Sporteinheiten angeleitet. Neben Jump 'n' Run gab es einige weitere analoge Umsetzungen von Games. Im Herbstworkshop sind 3 Bloxelsspiele\* und 2 Scratchspiele umgesetzt worden. Danach haben die Jungs noch verschiedene Sachen ausprobiert, es sind aber keine weiteren Spiele mehr fertig geworden.

Ursprünglich als Projekt der kreativen Mediennutzung im Stadtteil Gröpelingen gedacht, kam nach Werbung in sozialen Netzwerken doch knapp die Hälfte der Teilnehmer aus anderen Stadtteilen. Das hat funktioniert, weil

- 1) es gemeinsame Interessen und Ziele gab: Gaming, Spiel selbst gestalten und Parkour ausprobieren,
- 2) Ort und Thema für alle mehr oder weniger gleich neu waren (kein Heimvorteil),
- 3) ein großer Fundus an geteilten Medienerfahrungen (z.B. Games/Filme) vorhanden ist => Gesprächsthema,
- 4) beim Spielen\* persönliche Qualitäten (Spielerfahrung, Ideen, Humor, Hilfsbereitschaft) mehr als z.B. sozioökonomische Faktoren zählen.



\*Genutzt haben wir im Projekt den alten Bloxels Builder als App auf iPads. Da die Spiele nicht mehr veröffentlicht werden können, wurden im Anschluss 3 Let's Plays gedreht, in denen die Spiele gezeigt werden. Sie sind auf der Seite <https://www.bremer-medienpaedagogik.de/ruder-blog.html> zu sehen.

\*Hier verwende ich bewusst das deutsche Wort Spielen, weil es weiter gefasst ist als „Gaming“.

# Steckbrief Parkour

Anfang des Jahrtausends in Frankreich entwickelt. Grundidee: Effiziente Fortbewegung im (städtischen) Raum. Die Hindernisse beim Lauf werden springend, aber auch balancierend und kletternd überwunden. Fitness, Körperbeherrschung und die Fähigkeit zur Kontrolle der Situation werden trainiert. Dafür wird eine realistische Einschätzung der eigenen Fähigkeiten (z.B. „so weit kann ich schon springen“ „soviel Kraft habe ich noch“) und der räumlichen Gegebenheiten gebraucht. Bei Parkour geht es auch um die Auseinandersetzung mit den eigenen Ängsten (z.B. sich zu verletzen). Zwar werden im Internet gerne spektakuläre Runs gezeigt, trotzdem ist Parkour keine Sportart für leichtsinnige Draufgänger, sondern erfordert Geduld, hartes Training und risikobewusste Besonnenheit.

- wettbewerbsfrei
- Trainiert werden vor allem verschiedene Sprung- (zum Überwinden von Weite) und Klettertechniken (zum Überwinden von Höhe) sowie das Abrollen. Balancieren (Gleichgewicht) und Hangeln (Arm-/Handkraft) sind ebenfalls wichtige Elemente des Trainings.
- Durch das Aufsuchen verschiedener Spots (Trainingsorte) wird das flexible Umgehen mit räumlichen Situationen geübt.
- Schritt-für-Schritt Verschieben der eigenen körperlichen wie mentalen Grenzen
- Werte: Respekt, Anerkennung der Trainingsleistung, Motivation, Frustrationstoleranz, Hilfsbereitschaft, Verantwortungsübernahme, Rücksicht
- starke, internationale Community

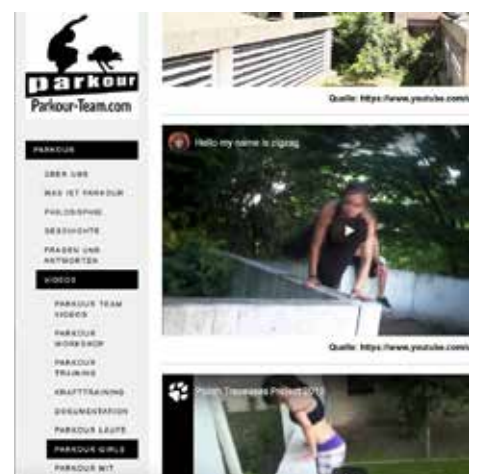
Im Prinzip kann man Parkour allein trainieren. Ich empfehle aber eine Gruppe mit Trainer, auf jeden Fall für den Anfang. In Bremen und anderen Städten gibt es zum Beispiel das Parkour-Team. Auf dessen Seite findet man u.a. weiterführende Informationen zur Geschichte und Philosophie von Parkour sowie Videos, die einen guten Einblick geben.

<https://www.parkour-team.com/>

Auch auf der Seite <https://parkourgenerations.com/> findet man ebenfalls viele Informationen dazu, was Parkour ausmacht. Englisch.

Wer es allein probieren möchte, kann sich z.B. auf dem Youtubekanal *urbanamadei* Anregungen holen.

Beispiel: <https://www.youtube.com/watch?v=oQuaHtzEu3o>



# Steckbrief Bloxels

Grundidee: Auf  $13 \times 13 = 169$  Feldern wird das ganze Spiel gebaut: Welt, Hintergründe, Inventar, Spielcharakter und Gegner, Texturen. Dazu gibt es ein Steckboard, das nicht zwingend nötig, aber gut zu haben ist; vor allem bei Teamarbeit ist es hilfreich, weil damit Sachen parallel gestaltet werden können (z.B. die Welt direkt am digitalen Gerät und Figuren am Board). Im Konstruktionsbereich definieren die Farben der Felder unterschiedliche Funktionalitäten, z.B. Untergrund, Gefahrenstelle, bewegliche Gegenstände oder Gegner, ansonsten das Aussehen.

- Ursprünglich App *Bloxels Builder* für Tablets  
=> keine Internetverbindung nötig, Daten lokal gespeichert mit „kindlicher“ 2D-Optik im Builder, deutlich Board orientiert  
Sprache: Deutsch



- browserbasiert => geräte- und systemübergreifend nutzbar
- Internetverbindung nötig
- Spielen (in Arcade) ohne Account möglich, sonst benötigt
- Verschiedene kostenpflichtige Accountformen:
  - Edu (Lehrkraft, Schüler\*in)
  - Privat (bei Kauf von Board kostenlos)
- Spiele können in der Arcade veröffentlicht werden
- „angesagte“ Retro-Optik im Builder
- nur 2D-Spiele möglich
- Sprache: Englisch
- Schießen-Button ersetzt durch Greiffunktion => Bauen, Umgestalten etc. möglich. (=> weniger Jump 'n' Run, mehr Aufgaben, Erkunden, Storytelling etc.) Es gibt eine Reihe von „Power-Ups“.
- Kein Import eigener Dateien außer Sound möglich.

<https://edu.bloxelsbuilder.com>



Oben: das alte Bloxels

Das neue Bloxels:



# Steckbrief Scratch

Grundidee: Programmiersprache mit übersichtlichem, grafischem Interface, dadurch tauglich für Kinder. Objekte, z.B. Spielfiguren oder Inventar, werden mit Hilfe von Skripten mit Attributen und Fähigkeiten ausgestattet (= objektbasierte Programmierung). Große stilistische und inhaltliche Offenheit erzeugt immense Vielfalt an individuell gestalteten Projekten. Kollaborativer Ansatz, es wird viel Wert auf Austausch gelegt.

- 2 Varianten: browserbasiert und stationär
- Browserbasierte Variante benötigt Internetverbindung und Account, sonst gleich.
- kostenlos
- Spielen und Einsehen der Projekte im Browser ohne Account möglich, Ändern nur mit Account oder App.
- Bibliothek mit Hintergründen, Sounds und Sprites
- Gestaltung von eigenen Sprites, Hintergründe oder Sound in Scratch direkt möglich oder Import von eigenen Dateien (Pixel- oder Vektorbasierte Dateien)
- große, unterstützende Community
- Veröffentlichte Projekte sind offen für alle => einsehen=lernen und ändern/weiterentwickeln möglich und gewünscht.
- nur 2-D- Projekte möglich
- Sprache: Deutsch



Der Ansatz von Scratch, das von der Lifelong-Kindergarten-Gruppe am MIT Media Lab des MIT (Buchempfehlung: „Lifelong Kindergarten“ von Mitchel Resnick, ISBN 978-3-946829-35-5) entwickelt und gepflegt wird, ist folgender: Im Rahmen eines selbstgewählten Projektes beschäftige ich mich spielerisch mit der Übertragung gewünschter Features in eine Programmierung und lerne dabei die zugrundeliegenden Strukturen (z.B. Wenn-dann-Beziehungen), verschiedene Funktionen (z.B. Variablen) und Strategien (z.B. Debuggen) beim Programmieren.

<https://scratch.mit.edu/>

Beispiel Jump 'n' Run (nicht aus dem Projekt):

<https://scratch.mit.edu/projects/467829086/> Snowman Rush

# Spiele selber gestalten

**Bloxels** stellt mit seiner Game-Engine festgelegte Fähigkeiten und Attribute zur Verfügung, die u.a. über Funktionsblöcke zugewiesen werden und nur in definiertem Rahmen modifiziert werden können. Beispiel: Der Spielcharakter kann voreingestellt laufen, springen und Dinge greifen. Einstellen kann man, wie schnell er laufen oder wie hoch er maximal springen kann – DASS er laufen kann, muss man aber nicht programmieren.

Indem es mir die eigentliche Programmierung abnimmt, legt Bloxels den Fokus auf die Gestaltung einer eigenen Spielwelt. Ganz im Stil eines 8-Bit Games gehalten, lassen sich mit Bloxels ausschließlich zweidimensionale Spiele gestalten, die aber durchaus eine 3-D-Optik haben können. Kennzeichnend ist die Verwendung von Sprites, d.h. Modulen, die mehrfach (z.B. Gegnerfiguren) bzw. repetierend (z.B. Mauerstücke o.ä. für Unter-/Hintergründe) eingesetzt werden. Alle Teile (bis auf Hintergrund) können animiert werden. Der Charakter verfügt zusätzlich über mehrere animierbare Zustände: Ruhe, laufen, springen. Mit Hilfe von Sprechblasen, die mit eigenem Text gefüllt werden können, ist es möglich, eine Spielgeschichte zu erzählen, Hinweise oder Hilfestellungen zu geben oder Aufgaben zu stellen.

All das macht Bloxels zu einem vielseitigen Tool, das trotzdem einfach in der Handhabung ist. Die Benutzerführung ist übersichtlich und erschließt sich auch schon Grundschulkindern problemlos, obwohl der Builder in der neuen Version auf Englisch ist. Das eigene Spiel ist spielbar, sobald die Funktionsblöcke platziert sind. Die Verschönerung ist optional, d.h. man kommt schnell zu einem spielbaren Ergebnis, kann sich aber auch in die Spielgestaltung versenken und lange tüfteln. Änderungen sind einfach und schnell gemacht, z.B. indem der Funktionsblock verschoben oder gelöscht wird. Das führt zu einer leichtfüßigen, spielerischen Herangehensweise. Da ständig ausprobiert werden muss, ob die Ideen in der Umsetzung taugen, wechseln die Nutzer ständig vom Builder- in den Spielmodus und haben beim Testen=Spiele Spaß.

**Scratch** hingegen ist ein allgemeines Programmierool. Hier sind die Möglichkeiten deutlich größer als bei Bloxels, es gibt aber keine vordefinierten Attribute oder Fähigkeiten, alles ist offen und muss eigenhändig programmiert werden. Das heißt beispielsweise, dass man sich für ein Jump 'n' Run mit der banal erscheinenden Frage auseinandersetzen muss, wie eine Figur auf einem Untergrund laufen kann, d.h. wie man die gewünschte physikalische Fähigkeit in eine Programmierung übersetzt\*. Zwar liefert Scratch mit seiner grafischen Oberfläche die Bausteine dazu in übersichtlicher Form, auch die Verknüpfung ist einfach durch Zusammenstecken, aber die grundsätzliche Programmierlogik muss verstanden werden. Dadurch dass nichts vorgegeben ist, sind die Möglichkeiten enorm, aber die Arbeit, bis etwas funktioniert wie gewünscht, ist kleinschrittig und braucht eine gute Struktur. Leichtfüßig wird Scratch erst nach intensiver Beschäftigung.

\*Ein Jump 'n' Run erfordert daher fortgeschrittene Programmierkenntnisse – mit entsprechender Unterstützung ist das aber durchaus im Kontext eines Projektes wie „Jump 'n' Run – analog 'n' digital“ machbar.

Noch ein paar Wort zu anderen Tools...

**Unity** ist eine Programmiersprache, mit der professionell Games gestaltet werden können. Da jedes Mehr an Möglichkeiten im gestaltbaren Bereich mit einer Erhöhung der Komplexität einhergeht, ist Unity nochmal eine Nummer anspruchsvoller als Scratch. Unity ist im Grunde nur geeignet für Menschen, die richtig Spaß am Programmieren und darin auch schon Erfahrung haben. Dafür gibt es ein Tutorial für Platformer (2-dimensionale Jump 'n' Runs) .

<https://unity.com/products/unity-platform>

Der **Mario Maker 2** funktioniert in vielerlei Hinsicht ähnlich wie Bloxels. Für Kinder sind die Figuren und Funktionen vertraut. Erwachsene finden das Gestalten auf dem kleinen Display der Nintendo Switch oft enerzierend, Kinder sind da robuster.

<https://supermariomaker.nintendo.com/>

Mit der App **Draw your Game** gibt es eine schlichte und ebenfalls analog-digitale Möglichkeit zum Selbergestalten von Jump 'n' Runs. Hier wird die Spielwelt gezeichnet und in die App gescannt. Taugt prima als Einstieg, ist aber so schlicht gehalten, dass es schnell an Reiz verliert.

<https://draw-your-game.com/>