

Das Vergnügen der Computerspiele



César Pimenta und Nico Schwab

Projektarbeit 9. Klasse

Volksschule Lenk

3. Mai 2021

Inhalt

Das Vergnügen der Computerspiele	1
Einleitung.....	3
Geschichte der Computerspiele	4
Der Anfang	4
Erste Schritte der Konsolen.....	4
Sega Mega Drive	5
Nintendo 64.....	5
Verkauf.....	6
Ablauf eines Computerspiels	6
Was ist Programmierung?	7
Wie soll ich anfangen?.....	7
Pixelart.....	7
2D Unity.....	8
Visual Studio.....	10
Arbeitsschritte.....	11
Planung	11
Umsetzung	11
Grafik.....	12
Programmierung.....	14
Algorithmus	16
Exportieren	17
Fazit.....	18
Arbeitsjournal.....	19
Abbildungsverzeichnis	23
Internetquellen.....	24

Einleitung

Computerspiele gehören zu den grössten Faszinationen mehrerer Generationen. Unsere Generation liebt Computerspiele, aber wir sind nicht die Ersten. Die ganze Geschichte hat schon in den 1940er Jahren angefangen und die Entwicklung ist noch nicht abgeschlossen. In der folgenden Arbeit wird der Weg beschrieben, wie man ein 2D «Jump and Run»-Spiel programmieren kann. Dafür werden zuerst die wesentlichen Punkte der Computerspiele geschildert. Es wird erklärt, was genau Programmierung ist und mit Beispielen veranschaulicht. Die Arbeitsschritte werden genau erklärt, so dass jede Person versteht, was wir gemacht haben. Unser Ziel ist es, ein Spiel zu programmieren, das nicht kompliziert ist und ohne zu stocken läuft. Es ist nicht die Absicht, etwas möglichst Neues und Kreatives zu entwickeln. Wir beschränken uns darauf, alles selbst zu programmieren und Probleme, die dabei auftreten, selbstständig zu lösen.

Geschichte der Computerspiele

Der Anfang

Estle Rey und Thomas T. Goldsmith Jr. haben 1946 zusammen das erste Computerspiel für einen Röhrenrechner entwickelt. Am 25. Januar 1947 haben sie es zum Patent angemeldet und das Patent wurde dann am 14. Dezember 1948 erteilt. Mit Hilfe eines NIMROD-Computers wurde 1951 das mathematische Nim-Spiel präsentiert und 1952 kam das Tic-Tac-Toe Spiel «OXO». 1958 hat ein amerikanischer Physiker namens William Higinbotham das Spiel «Tennis for Two» entwickelt. Dieses Spiel wird als das erste richtige Computerspiel angesehen. Damals war die Entwicklung eines Computerspiels sehr von der technischen Entwicklung der Computertechnologie abhängig. Deswegen wurden die ersten Computerspiele von Studenten und Wissenschaftlern auf Universitäten entwickelt, weil man dort mit Grossrechnern arbeiten konnte.¹²

Erste Schritte der Konsolen

1984 war das Jahr der japanischen Videospielehersteller Sega und Nintendo. Dank der Krise konnten sie ohne Konkurrenz in den US-Markt einsteigen. Mit Spielfiguren wie "Sonic the Hedgehog" und "Super Mario" zogen Sega und Nintendo ins pop-kulturelle Gedächtnis der Gesellschaft ein und waren somit auch in einem Konkurrenzkampf um die Gunst der Spieler. Sega brachte 1989 auch schon ihre erste Konsole auf den Markt, die Mega Drive, welche die Generation der 16-Bit-Geräte einläutete. 1991 kam auch schon Nintendo mit dem Super Nintendo nach. Die Spiele waren allerdings alle noch in 2D, Ausflüge in den dreidimensionalen Raum waren erst 1993 mit Hilfe eines

¹ Quelle: https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

² Quelle: <https://www.game.de/spielgeschichte/>

Super FX-Chips möglich. Nur ein Jahr später brachte Sony die Playstation auf den Markt und eröffnete somit die Generation der 32-Bit-Konsolen. Der Unterschied zwischen 32 und 64 Bit ist, dass die eine Konsole 32 Bit produziert. Dank dem neuen Speichermedium CD waren gerenderte Videosequenzen und filmreife Soundtracks möglich. Die einzige Konkurrenz damals bestand aus dem SEGA Saturn und der 64-Bit-Konsole Nintendo 64.³⁴

Sega Mega Drive



Abbildung 1: Sega Mega Drive,
https://www.google.ch/search?q=sega+mega+drive&sxsrf=ALeKk02i4zAqueGCoWvQCJBxWTcog-nKZ3Q:1618846276337&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjZuPyp0lrwAhXKIqQKHT-wgBVkQ_AUoAXoECAE-QAw&biw=1920&bih=937#imgrc=cPfbjyMvgU8IGM

Nintendo 64



Abbildung 2: Nintendo 64,
[https://www.google.ch/search?q=nintendo+64&tbm=isch&ved=2ahUKEwjJqNeq0lrwAhUXgaQKHY4FCtgQ2-cCegQIABAA&oq=Ninten&gs_lcp=CgNpbWcQARgAMgQIIXAnMgcl-ABCxAXBDMgclABCxAXBDMgQIABBDMgQI-ABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDUMLjCViA6glg9vIJaABw](https://www.google.ch/search?q=nintendo+64&tbm=isch&ved=2ahUKEwjJqNeq0lrwAhUXgaQKHY4FCtgQ2-cCegQIABAA&oq=Ninten&gs_lcp=CgNpbWcQARgAMgQIIXAnMgcl-ABCxAXBDMgclABCxAXBDMgQIABBDMgQI-ABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDUMLjCViA6glg9vIJaABw)

³Quelle: https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

⁴ Quelle: <https://www.game.de/spielgeschichte/>

Verkauf

Auch wenn der physische Kauf von Spielen noch möglich ist, bewegt sich seit 2013 der Markt zunehmend auf ein Downloadgeschäft hin, besonders der Online-Verkauf von virtuellen Gütern und Zusatzinhalten. Onlinehändler wie Steam, Origin etc. ermöglichen diesen Kauf.⁵⁶ Dabei geraten Spiele in einer CD-Version immer mehr in den Hintergrund.

Ablauf eines Computerspiels

1. Idee und Konzeptentwicklung
2. Finanzierung und Kostenplanung
3. Produktionsphase
 - a) Levels, Missionen, Drehbuch (Charaktere und Story), Spieldesign (Skizzen, Spielkonzept und Steuerung)
 - b) Künstliche Intelligenz, Physik- und Grafik-Engine (virtuelle Umgebung), Programmierung (Spielmechanik und Steuerung)
 - c) Figuren, Objekte, Musik Sprache, Sound und Benutzeroberfläche erstellen
 - d) Entwicklung von Features und Download-Erweiterungen
 - e) PR und Marketing
 - f) Testphase und Fehlerbehebung
 - g) 3D Head Scanning und Motion Capturing
4. Herstellung, Verkauf und Werbung
5. Veröffentlichen von Patches, Multiplayer-Modus-Aktualisierungen und DLCs⁷

⁵ Quelle: https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

⁶Quelle: <https://www.game.de/spielgeschichte/>

⁷ Quelle: <https://spielkultur.ea.de/themen/gesellschaft-und-kultur/ein-computerspiel-entsteht-von-der-idee-bis-zum-spielvergnugen/>

Was ist Programmierung?

Beim Programmieren ist das Ziel, ein Programm herzustellen, das der Computer versteht und so auch ausführen kann. Man programmiert Spiele, Apps, Webseiten, E-Books, Roboter-Steuerungen und auch Autopiloten bei Flugzeugen. Dies sind alles Softwares, die durch das Programmieren erstellt werden.⁸

Wie soll ich anfangen?

Man kann auf verschiedene Arten programmieren. Für Website-Design und Programmierung reicht CSS und HTML für den Anfang. Mit Java oder Python kann man Webseiten und Apps interaktiv machen. Dann gibt es noch vereinfachte Programmiersprachen wie JavaScript oder TypeScript, die für Anfänger hilfreich sein können. Man kann aber als Anfänger auch mit einer normalen Programmiersprache anfangen. Im nächsten Abschnitt zeigen wir die Tools, die man als Anfänger beim Programmieren kennen sollte.⁹

Pixelart

Pixelart ist eine Stilart in der Computerkunst. Sie verwendet Rastergrafiken. Das Stilmittel von beschränkten Auflösungsvermögen bei Bildschirmen wird bewusst eingesetzt. Pixel ist die Abkürzung für Picture Element. Auf Deutsch bedeutet dies Bildteil. Ein Pixel ist die kleinste Einheit eines Bildes, die auf einem Monitor dargestellt werden kann. Alle Bilder bestehen aus ganz vielen kleinen Pixeln. Wenn man auf eine

⁸ Quelle: <https://de.wikipedia.org/wiki/Programmierung>

⁹ Quelle: <https://de.wikipedia.org/wiki/Programmierung>

Webseite geht, werden alle Pixel geladen. Aufgrund der geladenen Pixel kann man nachschauen, ob ein Nutzer die Webseite benutzt hat. Die Abkürzung für Pixel ist px.¹⁰



Abbildung 3: Beispiel für Pixel Art Bild: Pixel Art Mario, https://www.google.ch/search?q=pixel+Art+mario&tbm=isch&ved=2ahUKEwjhwqDX0Yr-wAhWO0YUKHXEVBEoQ2-cCegQIABAA&oeq=pixel+Art+mario&gs_lcp=CgNpbWcQAzIECCMQJzICCAAYAggAMgIIADICCAAYAggAMgIIADICCAAYAggAMgIIADoECAAAQ1D-LVjXNmCWomgAcAB4AYABywGIAfs

2D Unity

2D Unity ist eine Plattform, auf der man 2D Spiele gestalten und optimieren kann, indem man z.B. verschiedene Hindernisse einfügt. 2D Unity ermöglicht uns auch unser Spiel auf allen kommenden und führenden mobilen Betriebssystemen bereitzustellen. 2D Unity beschleunigt den Entwicklungsprozess. Viele Spiele, die mit 2D Unity optimiert worden sind, haben schon einen kommerziellen Erfolg erzielt. Wir haben uns für diese Plattform entschieden, weil wir erstens schon beide etwas Erfahrung mit dieser Plattform haben und sie kostenlos ist.¹¹

¹⁰ Quelle: <https://de.wikipedia.org/wiki/Pixel-Art>

¹¹ Quelle: <https://unity3d.com/de/2d/solution-guide#:~:text=Unity%20ist%20eine%20vollst%C3%A4ndige%20Plattform%20f%C3%BCr%20D.,und%20kommerzielle%20Erfolge%20zu%20erzielen.>

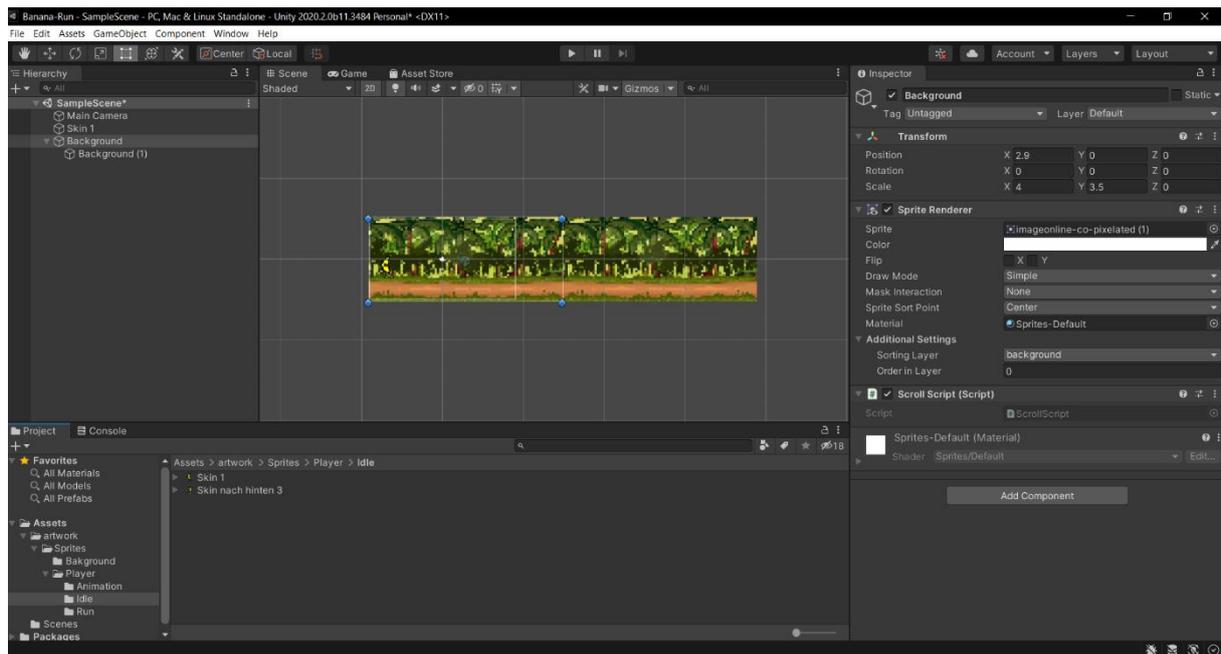


Abbildung 4: 2D Unity Screenshot

Hier sieht man einen Screenshot von 2D Unity. Hier haben wir an unserem Spiel gearbeitet. In der Mitte ist die Position unseres Elementes zu sehen. In unserem Fall der Hintergrund, die Skripte usw. und rechts haben wir den sogenannten Inspector. Dort können wir beispielsweise mit Zahlen die genaue Position unseres Hintergrundes festlegen. Dort sind alle Informationen und man kann dort verschiedene Sachen bearbeiten, unter anderem hat man die Funktion «realistische Physik». Diese dient für die Schwerkraft des Spielers. Dies bedeutet, dass wenn man z.B. springt, dass die Figur auch wieder angezogen wird und wieder auf dem Boden landet. Wenn man diese Funktion nicht eingestellt hätte, würde der Spieler einfach rumschweben. Unten sieht man die Dateien, die man im 2D Unity darin hat, wie die Dateien der Figur, also alle Animationen. Unter anderem sind da auch die Skripte abgespeichert.

Visual Studio

Visual Studio ist eine Entwicklungsumgebung für verschiedene Hochsprachen. Das Programm ist von Microsoft und ist kostenlos. Mit Visual Studio kann man in verschiedenen Sprachen für verschiedene Zwecke programmieren. Wir kreieren unser Spiel im 2D Unity, aber ohne Visual Studio wäre dies nicht möglich, da wir die Skripte im Visual Studio machen und die dann im 2D Unity einfügen. Ein Skript ist eine Textdatei, in der wir programmieren und so eine bestimmte Logik herstellen. Eine einzige Bewegung wie z.B. ein Sprung nimmt schon ein ganzes Skript ein.¹²

```
1  using UnityEngine;
2
3  public class Movement : MonoBehaviour
4  {
5      public float Movementspeed = 9;
6      public float JumpForce = 7;
7
8      private Rigidbody2D _rigidbody;
9
10     private void Start()
11     {
12         _rigidbody = GetComponent<Rigidbody2D>();
13     }
14
15
16     private void Update()
17     {
18         var movement = Input.GetAxis("Horizontal");
19         transform.position += new Vector3(movement, 0, 0) * Time.deltaTime * Movementspeed;
20
21         if (Input.GetButtonDown("Jump") && Mathf.Abs(_rigidbody.velocity.y) < 0.001f)
22         {
23             _rigidbody.AddForce(new Vector2(0, JumpForce), ForceMode2D.Impulse);
24         }
25     }
26
27
28 }
```

Abbildung 5: Skript Screenshot

In diesem Skript haben wir die Bewegung des Spielers programmiert. Die Laufgeschwindigkeit, die Sprunghöhe und die Sprungstärke sind auf diesem Abschnitt zu sehen.

¹² Quelle: https://de.wikipedia.org/wiki/Visual_Studio

Arbeitsschritte

Planung

Zuerst haben wir uns überlegt, was wir für ein Spiel machen wollen, mit welchen Programmen und wie wir es umsetzen wollen. Wir wollen ein Spiel programmieren und es auf eine Website hochladen, die wir selbst erstellen. Unser Ziel ist, dabei von Anfang an kein Geld auszugeben. Uns war bewusst, dass wir keine genügend guten Programme zu Verfügung haben würden, um eine hochwertige Grafik zu gestalten. Aus diesem Grund haben wir uns für Pixelart-Grafik entschieden, da man diese mit kostenlosen Programmen hochwertig gestalten kann. Nach langer Recherche haben wir für die Grafik das Programm Gimp ausgewählt und für die Programmierung des Gameplays entschieden wir uns für 2D Unity.

Umsetzung

Als wir wussten, welche Programme wir verwenden werden, haben wir angefangen zu planen. Wir haben die Arbeit aufgeteilt und aufgeschrieben, von wann bis wann wir Zeit haben. So eine Zeitplanung war gut, damit wir nicht anfangen Sachen zu verschieben und so unter Zeitdruck kommen würden. Nico hat eine Umfrage gemacht, damit wir die Meinung der Leute zu Computerspielen rausfinden. Währenddessen habe ich mit der Recherche angefangen. Mein erstes Thema war die Geschichte der Computerspiele. Wir wollten herausfinden, wie eigentlich Computerspiele entstanden sind, wer auf die Idee kam und wie die Idee umgesetzt wurde. Ich habe meine Informationen mit Nico ausgetauscht und wir waren beide sehr erstaunt, da wir gar nicht wussten, wie es eigentlich entstanden ist. Mehr über die Geschichte der Computerspiele herauszufinden, fand ich persönlich sehr spannend und hat mir sehr viel Spass gemacht. In der Geschichte der Computerspiele spielen Konsolen auch eine sehr grosse Rolle, deswegen habe ich auch eine Recherche zu der Entwicklung der Konsolen gemacht. Die Entwicklung von Gameboys zu richtigen Konsolen ist sehr interessant und es war ein richtiger Konkurrenzkampf zwischen Konsolen wie Nintendo und Sega Mega Drive.

Währenddessen hatte Nico bereits mit dem Erstellen der Website angefangen. Die Website haben wir mit CSS und HTML mit Visual Studio Code programmiert. Visual Studio Code ist ein Programm, in dem man mit CSS und HTML arbeiten kann. Uns war bewusst, dass viele Probleme auftauchen würden, da Probleme und Fehler beheben zum Programmieren dazugehören. Da tauchte das erste Problem auch schon auf. Natürlich wollten wir gemeinsam gleichzeitig an derselben Datei arbeiten, aber das konnten wir nicht. Wir wussten, dass es funktionieren würde. Das war auch einer der Gründe, wieso wir dieses Programm ausgesucht hatten. Wir recherchierten also sofort, wie wir das Problem lösen könnten. Da sind wir auf Visual Studio Live Share gestossen. Ein Add-On, das wir beide für Visual Studio Code herunterladen mussten. Ein Add-On ist ein Erweiterungstool für ein beliebiges Programm. Beim Programmieren der Website hatten wir keine grossen Schwierigkeiten und haben alle Befehle für HTML und CSS gefunden. Als die Website erstellt und nutzungsbereit war, hat sich Nico um das Design der Website gekümmert. Zuerst haben wir zusammen besprochen, in welche Richtung unser Design gehen soll. Wir haben uns für ein Dark Mode Design entschieden, das heisst dunkler Hintergrund und helle Schrift. Währenddessen habe ich mit einer nächsten Recherche angefangen. Da wir ein Computerspiel erstellen, dachten wir, dass es gut wäre, wenn man auch weiss, wie der Ablauf zum Erstellen eines Computerspieles vor sich geht. Zuerst wollte ich einen normalen Fliesstext schreiben, dann ist mir aber klar geworden, dass es viel übersichtlicher wäre, wenn ich es nummerieren würde. Es war ein eher komplexeres Thema, aber ich konnte die Informationen gut finden. Es war spannend, dass dieser Ablauf ziemlich passend im Vergleich zu unserem war, obwohl wir uns gar nicht darauf konzentriert haben. Wenn man ein Computerspiel programmieren will, fanden wir heraus, dass der Ablauf immer identisch ist.

Grafik

Jetzt hatten wir schon einen Teil der Recherche und wir hatten unsere Website fertig. Der nächste Schritt war, die Grafik für das Spiel herzustellen. Hier hatten wir am meisten Probleme und mussten sehr viel Zeit investieren. Wir hatten eigentlich die Idee, die

Grafik mit einem neuen Photoshop Programm herzustellen. Man hat eigentlich auch die Möglichkeit, das Programm gratis auszuprobieren und das während einem Monat. Aus einem unbekanntem Grund konnten wir es doch nicht kostenlos testen und da unser Ziel war, ein Spiel ohne Geld auszugeben zu programmieren, haben wir nach anderen Programmen recherchiert. Da haben wir wertvolle Zeit verloren. Wegen der geringen Auswahl haben wir uns für das Gimp Programm entschieden. Das Gimp Programm ist sehr simpel gehalten und so hatten wir alles, um eine anständige Pixel Art Grafik zu gestalten. Jetzt konnten wir endlich mit dem Gestalten loslegen.

Wie immer haben wir uns vor dem Start zuerst die Arbeit aufgeteilt, damit wir nicht durcheinanderkommen. Wir haben uns dafür entschieden, dass Nico sich um die Spielfiguren/Charakter kümmert und ich mich um den Hintergrund kümmern soll. Wir haben lange und intensiv daran gearbeitet. Nico hat verschiedene Figuren gestaltet und ich verschiedene Hintergründe. Wir haben uns viele 2D Jump and Runs Hintergründe angeschaut und uns so inspirieren lassen. Wir hatten zuerst die Idee, einen Wald als Hintergrund zu gestalten. Das habe ich auch gemacht. Danach habe ich noch mit einem Schneegebiet angefangen, aber mittendrin abgebrochen, da wir schnell erkannten, dass es nicht zu unseren Spielfiguren passen würde. Nico kam dann auf die Idee, einen «Banana Skin» zu gestalten. Zuerst haben wir darüber gelacht, aber als er dann angefangen hatte, sah es gar nicht mal so schlecht aus und nachdem ich auch noch mal ein paar Detailideen gegeben hatte, hatten wir einen ziemlich guten und lustigen Skin zusammengestellt. Wir entschieden uns, ihn erstmal zu behalten. Da waren wir noch lange nicht fertig und an dem Zeitpunkt hatten wir uns auch noch nicht fest für den «Banana Skin» entschieden. Nico hat weiterhin Skin Entwürfe gestaltet und ich habe einen weiteren Hintergrund gestaltet, den ich auch fertigstellen konnte.

Man könnte denken, weil wir in Pixel Art Grafik gestalten, dass es nicht so lange braucht, aber es ist eben gerade nicht so, da wir, wenn wir eine Figur gestaltet haben, jeden einzelnen Pixel setzen mussten und dazu konnten wir nicht einfach ein paar Farben nehmen, wir mussten zu jeder einzelnen Farbe, die wir gebraucht haben, viele verschiedene Farbtöne verwenden, damit es nach etwas aussieht. Daher hat die Arbeit mit der Grafik am längsten gebraucht. Wir kamen immer wieder auf den «Banana

Skin» zurück und merkten mit der Zeit, dass er wahrscheinlich unser Skin werden würde. Ich war gerade daran, einen eher strandmässigen Hintergrund mit Palmen und so weiter zu gestalten sowie im Hintergrund noch einen gelben Himmel (bzw. der Hintergrund des Hintergrunds). Unser Hintergrund passte perfekt zu unserer Banane. Nun hatten wir die Grafik und wollten diese jetzt ins 2D Unity einfügen, damit wir mit ihr auch arbeiten können. An diesem Punkt hatten wir unser nächstes Problem. Die gesamte Grafik, also Charakter und Hintergrund, war sehr unscharf. Das Problem war, dass die Pixel nicht mit der normalen Grösse vom 2D Unity gepasst hat.

Wir haben uns zu zweit direkt an das Problem gemacht und haben dies gelöst, indem wir im Gimp, in dem wir die Figuren hatten, die Datei der Figuren nicht einfach genommen und ins 2D Unity eingefügt haben, sondern mit dem Snippingtool ein Screenshot im Gimp gemacht haben und dieses zurechtschnitten. Nun hatten wir eine neue Datei mit denselben Figuren darin. Diese Datei hatte nun die passende Pixelgrösse, um sie ins 2D Unity einzufügen.

Programmierung

Zu unserer Recherche gehören alle Sachen, mit denen wir arbeiten, zu erklären. Da wir mit 2D Unity gearbeitet haben, gehörte natürlich auch eine Recherche zu diesem Programm dazu. 2D Unity war das Programm, mit dem wir alles überhaupt umsetzen konnten. Obwohl wir mit Gimp bzw. der Grafik am meisten Zeit verbracht haben, hätten wir alles, das wir gemacht haben ohne 2D Unity gar nicht gebrauchen können. Wir haben uns schon ganz am Anfang beim Planen genug Zeit fürs Programmieren eingeteilt, weil man das Programmieren einfach nicht unterschätzen darf. Bei den Bewegungen der Figur wurde es komplex. Wir haben uns damals gefreut, als wir es geschafft haben, dass unsere Figur springen konnte. Wenn man davon hört, dann klingt das eigentlich nach sehr wenig, da es ja "nur" eine Bewegung ist. In Wirklichkeit ist es eine ganze Seite mit Befehlen und Codes. Für uns war das Programmieren der Prozess, bei dem wir am genauesten arbeiten mussten. Denn nur für so eine kleine Bewegung, wie z.B. einen Sprung, sind so viele Details dahinter, wie die Sprunghöhe, die

Sprungstärke und vieles Mehr. Dazu muss man sehr genau arbeiten, da jedes kleine Zeichen, Punkt und sogar ein Leerschlag stimmen muss. Wenn dabei eine Kleinigkeit nicht stimmt, funktioniert gar nichts mehr und wenn mal ein Fehler passiert, wird der nicht angezeigt. Man muss selbst dahinter gehen und den Fehler herausfinden. Allein für die Spielfigur mussten wir den Sprung, die Bewegung nach links und die Bewegung nach rechts programmieren. Zudem mussten wir bedenken, dass die Figur nicht einfach schweben darf.

Die Figur musste für jeden Schritt, den sie macht, die Animation wechseln. Beim Herstellen des Designs haben wir daran gedacht. Wir haben die Figur normal, mit dem linken Fuss nach oben, mit dem rechten Fuss nach oben und dies alles nochmal gespiegelt designt, im Fall, dass man in die andere Richtung läuft. Danach mussten wir die Bewegung des Hintergrundes programmieren. Das heisst, dass sich der Hintergrund nach links bewegt, wenn sich die Figur nach rechts bewegt. Das gibt den Effekt, dass man sich nach vorne oder nach hinten bewegt.

Da ich zu diesem Zeitpunkt mehr an der Dokumentation gearbeitet habe, bzw. korrigiert, überarbeitet und weitergeschrieben habe, ging Nico schon mal ohne mich an die Sache heran. Nico hat nach den wichtigsten Befehlen recherchiert und danach schon angefangen. Währenddessen hatte ich unsere Dokumentation überarbeitet und konnte deswegen danach Nico helfen. Nach dem Programmieren stimmte noch nicht alles. Wir haben zweimal den gleichen Hintergrund eingefügt und die Programmierung sagte sozusagen, dass der hintere Hintergrund immer nach vorne geschoben wird, wenn der Spieler vorne ankommt. Also der hintere Hintergrund wird vorne angehängt und der, der vorne war, wird zum hinteren. Das geht dann unendlich lang, bis der Spieler stirbt. Die Probleme waren, dass z.B. der Hintergrund zu spät erschien oder dass er nicht schön eingefügt wurde. Wir wollten aber, dass man gar nicht bemerkt, dass der Hintergrund immer eingefügt wird. Man sollte einfach unendlich laufen können und nichts bemerken. Diese kleinen Fehler konnten wir zum Glück mit eigenem Wissen lösen.

Algorithmus

Mittlerweile waren wir schon sehr nah an unserem Ziel angekommen. Nur kam jetzt eine richtige Herausforderung. Wenn man in unserem Spiel nur endlos laufen würde, wäre es langweilig und es gäbe kein Ziel. Also mussten wir jetzt Hindernisse einbauen. Die sollten immer unterschiedlich im Spiel erscheinen, aber in einer bestimmten Reihenfolge und genau dafür brauchten wir einen Algorithmus. Uns war bewusst, dass unser Wissen nicht ausreicht, um einen kompletten Algorithmus zu coden/programmieren. Wir haben nach Informationen, Einleitung usw. recherchiert. Leider erfolglos. Daher kamen wir auf die Idee, einen Kollegen von uns zu fragen, der schon älter ist als wir und viel mehr Erfahrung mit coden/programmieren hat. Wir haben mit ihm Kontakt aufgenommen und ihm ausführlich erklärt, was wir machen und was unser Ziel sei. Wir haben mit ihm einen Termin vereinbart und er hat sich für uns Zeit genommen. Er hat sich das ganze mal angeschaut, aber leider konnte er uns nicht helfen, da wir mit der Programmiersprache C-Shark programmierten und er seine Kenntnisse mit Java hat. Java und C-Shark sind ähnlich, aber leider unterscheiden sie sich doch noch ein Stückchen. Jetzt waren wir wieder auf uns allein gestellt. Wir haben wieder recherchiert und haben auf YouTube einen eher kleineren, unbekannteren, deutschen Kanal gefunden. Auf diesem Kanal haben wir ein Video gefunden, in dem erklärt wurde, wie man das programmieren kann.

Wir haben uns das Video gemeinsam angeschaut und konnten dank der Einleitung den Algorithmus innerhalb einer Stunde programmieren. Es war nicht einfach normales Coden in einem Skript, sondern es war eigentlich ein vereinfachtes Coden, indem man Blöcke zusammensetzen musste und in diesen Blocks verschiedene Angaben setzen konnte. Danach haben wir noch ein paar Kleinigkeiten wie Tempo, Sprunghöhe usw. weiter verbessert und noch programmiert, dass das Startmenu wieder erscheint, wenn der Spieler stirbt.



Exportieren

Jetzt waren wir schon fast am Ziel angekommen. Es fehlte nur noch die Veröffentlichung unseres Spiels. Dafür mussten wir unser Spiel vom 2D Unity auf unsere Webseite exportieren. Zuerst haben wir das Spiel zu einer Datei exportiert. Wir schauten nach, was die Datei für einen Code hat, sozusagen eine Identifizierung, damit der Rechner weiss, dass es diese Datei ist. Mit diesem Code konnten wir im Skript der Website die Datei bzw. unser Spiel auf diese Webseite hochladen. Danach mussten wir noch alles skalieren, beispielsweise musste die Grösse eingestellt werden. Unser Spiel war nun fertig und veröffentlicht.

Fazit

Wenn wir jetzt zurückblicken, müssen wir sagen, dass wir wirklich viel gemacht haben und sehr viel Aufwand in unser Projekt reingesteckt haben. Wir waren ein gutes Team und die Zusammenarbeit hat immer hervorragend geklappt. Wir sind oft über Hindernisse gestolpert, aber haben die immer ohne Unterstützung gelöst. Am Schluss haben wir unser Ziel erreicht. Wir konnten ein Spiel programmieren, das unsere Generation, aber auch andere faszinieren könnte.

Wenn man unser Spiel ohne Hintergrundwissen anschaut, dann denkt man wahrscheinlich, dass da nicht so viel Arbeit dahintersteckt. Auch wenn wir ein sehr kleines und simples Spiel erschaffen haben, sind wir zufrieden, da wir uns mit einem sehr komplexen Thema befasst haben und alles selbstständig gecodet haben. Uns war bewusst, dass uns wahrscheinlich niemand helfen könnte, da wir niemanden kennen, der sich mit unserer Programmiersprache auskennt. Manchmal tauchten schon fast zu viele Probleme auf und das hat schon das eine oder andere Mal unsere Motivation beeinträchtigt, aber wir haben trotzdem immer zusammen eine Lösung gefunden. Wir sind der Meinung, dass sich diese Arbeit gelohnt hat und sind stolz auf unser Ergebnis.

Arbeitsjournal

Datum	Dauer	Ausgeführte Arbeiten	Probleme Lösungen	Weiteres Vorgehen
11.11.2020	90min	Heute haben wir unsere Vereinbarung abgeschlossen und unser Inhaltsverzeichnis fertig gemacht.	Wir können jetzt gleichzeitig an einem Dokument arbeiten. Mithilfe von Google Docs.	Nico wird eine Umfrage machen und César wird mit der Recherche anfangen.
18.11.2020	90min	Heute haben wir die Umfrage und die Recherche angefangen.	Wir hatten keine Probleme.	Umfrage und Geschichte bis 25.11.2020 fertig machen.
24.11.2020	40min	Wir haben die Umfrage und die Recherche fertig gemacht.	Wir hatten keine Probleme.	Informieren mit welchem Programm wir die Website erstellen.
25.11.2020	90min	Recherchiert und informiert mit welchem Programm wir die Website erstellen.	Wir hatten keine Probleme.	Website erstellen
2.12.2020	90min	Ablauf eines Computerspieles fertig gemacht und angefangen mit Website erstellen.	Wir hatten keine Probleme.	Website erstellen
9.12.2020	90min	Wir haben an der Website gearbeitet.	César hat auf der Plattform noch keine Rechte, um zu schreiben.	César Rechte geben damit beide gleichzeitig im Visual Studio Code arbeiten können.

13.01.2021	90min	Live Share Problem gelöst und Website Grundriss fertig gemacht.	Wir hatten keine Probleme.	Die Grafik für unser Spiel erstellen.
20.01.2021	90min	Heute haben wir uns über Photoshop informiert.	Wir hatten keine Probleme.	Grafik Design Programm auswählen.
27.01.2021	90min	Grafik Programm Gimp gefunden und mit Charakter und Hintergrund Design angefangen.	Wir hatten keine Probleme.	Charakter und Hintergrund designen.
03.02.2021	90min	Wir haben uns für Pixel Art design entschieden und mit dem Charakterdesign begonnen.	Wir hatten keine Probleme.	Charakter weiter designen und mit Hintergrund anfangen.
10.02.2021	90min	Wir haben unseren Charakter weiter designet und mit dem Hintergrund angefangen.	Wir hatten keine Probleme.	Charakter fertig machen und den Hintergrund designen.
17.02.2021	90min	Text zu Programmierung fertig geschrieben und Banane fertig gestaltet.	Wir hatten keine Probleme-	Text zu Pixel Art fertig schreiben.
24.02.2021	90min	Pixel Art Text fertig geschrieben und der Hintergrund ist fast fertig.	Wir hatten keine Probleme	Komplette Grafik abschliessen innerhalb der Sportferien.
09.03.2021	60min	Grafik abgeschlossen.	Wir hatten keine Probleme.	Wir werden mit 2D Unity starten.

10.03.2021	90min	Abgesprochen, 2D Unity Text geschrieben, Gameplay Test Setup angefangen.	Wir hatten keine Probleme.	Recherche weiter machen und Grafik ins 2D Unity einfügen.
17.03.2021	90min	Figuren scharf gemacht und Recherche weitergemacht.	Unsere Figuren waren im 2D Unity unscharf.	Dokumentation weiterschreiben und weiter am Gameplay arbeiten.
24.03.2021	180min	Dokumentation weitergeschrieben und programmieren, dass sich die Figur sich nach links und rechts bewegen kann.	Wir hatten keine Probleme.	Springen programmieren und Dokumentation weiter schreiben
30.03.2021	120min	Dokumentation weitergeschrieben.	Wir hatten keine Probleme.	Gameplay weiter programmieren und Dokumentation weiterschreiben.
31.03.2021	90min	Wir haben die Bewegungen der Figur fertig programmiert.	Wir hatten keine Probleme.	Dokumentation weiterschreiben und die Animation der Figur programmieren.
07.04.2021	90min	Dokumentation geschrieben und angefangen mit der Animation der Figur programmiert.	Wir hatten keine Probleme.	Dokumentation weiterschreiben die fortlaufende Bewegung des Hintergrundes programmieren.
13.04.2021	180min	Dokumentation geschrieben und recherchiert, wie wir unser Problem mit der Bewegung des Hintergrundes lösen.	Bewegung unseres Hintergrundes. Noch nicht gelöst.	Wir müssen unser Problem lösen, das Spiel abschliessen und die Dokumentation weiterschreiben.

14.04.2021	90min	Wir haben den Hintergrund zum Laufen gebracht und mit der Charakter Animation angefangen.	Wir haben das Problem mit unserem Hintergrund gelöst. Der Hintergrund bewegt sich jetzt nach links.	Die Laufanimation fertig machen und danach den Algorithmus machen.
16.04.2021	120min	Recherche zu Visual Studio Code geschrieben und Algorithmus angefangen zu programmieren.	Wir hatten keine Probleme.	Startbildschirm erstellen und Algorithmus programmieren.
19.04.2021	150min	Dokumentation korrigiert und Startbildschirm erstellt.	Startbildschirm hat zuerst nicht funktioniert.	Algorithmus fertig programmieren.
21.04.2021	90min	Wir haben Infos zur Programmierung von einem Algorithmus recherchiert und damit angefangen.	Wir hatten keine Probleme.	Algorithmus von den Hindernissen programmieren und das Spiel hochladen.
23.04.2021	150min	Wir haben den Algorithmus der Hindernisse programmiert und das Spiel auf unsere Webseite hochgeladen.	Wir hatten keine Probleme.	Dokumentation fertig überarbeiten.
26.04.2021	80min	Dokumentation fertig überarbeitet.	Wir hatten keine Probleme.	Dokumentation abgeben

Abbildungsverzeichnis

Abbildung 1: Sega Mega Drive,
https://www.google.ch/search?q=sega+mega+drive&sxsrf=ALeKk02i4zAqueGCoWvQCJBxWTcognKZ3Q:1618846276337&source=Inms&tbm=isch&sa=X&ved=2ahUKEwjZuPyp0lrwAhXKlqQKHTwgBVkQ_AUoAXoECAEQAw&biw=1920&bih=937#imgrc=cPfbjyMvgU8IGM.......... 5

Abbildung 2: Nintendo 64,
[https://www.google.ch/search?q=nintendo+64&tbm=isch&ved=2ahUKEwjJqNeq0lrwAhUXgaQKHY4FCtgQ2-cCegQIABAA&oq=Ninten&gs_lcp=CgNpbWcQARgAMgQIlxAnMgclABCxAxBDMgclABCxAxBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDUMljCViA6glg9vIJaABw.....](https://www.google.ch/search?q=nintendo+64&tbm=isch&ved=2ahUKEwjJqNeq0lrwAhUXgaQKHY4FCtgQ2-cCegQIABAA&oq=Ninten&gs_lcp=CgNpbWcQARgAMgQIlxAnMgclABCxAxBDMgclABCxAxBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDUMljCViA6glg9vIJaABw.....)..... 5

Abbildung 3: Beispiel für Pixel Art Bild: Pixel Art Mario,
https://www.google.ch/search?q=pixel+Art+mario&tbm=isch&ved=2ahUKEwjhwqDX0YrwAhW00YUKHXEVBEoQ2-cCegQIABAA&oq=pixel+Art+mario&gs_lcp=CgNpbWcQAzIECCMQJzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADoECAAAQQ1D-LVjXNmCWomgAcAB4AYABywGIAfs.......... 8

Abbildung 4: 2D Unity Screenshot..... 9

Abbildung 5: Skript Screenshot 10

Internetquellen

https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

<https://www.game.de/spielgeschichte/>

https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

<https://www.game.de/spielgeschichte/>

https://de.wikipedia.org/wiki/Geschichte_der_Videospiele

<https://www.game.de/spielgeschichte/>

<https://spielkultur.ea.de/themen/gesellschaft-und-kultur/ein-computerspiel-entsteht-von-der-idee-bis-zum-spielvergnugen/>

<https://de.wikipedia.org/wiki/Programmierung>

<https://de.wikipedia.org/wiki/Programmierung>

<https://de.wikipedia.org/wiki/Pixel-Art>

<https://unity3d.com/de/2d/solution-guide#:~:text=Unity%20ist%20eine%20vollst%C3%A4ndige%20Plattform%20f%C3%BCr%202D.,und%20kommerzielle%20Erfolge%20zu%20erzielen.>

https://de.wikipedia.org/wiki/Visual_Studio