

Bachelorarbeit

René Perlbach

Konzept, Entwurf und prototypische Implementierung zur
Portierung einer Maus/Tastatur-Steuerung auf eine Gamepad-Steuerung

Konzept, Entwurf und prototypische Implementierung zur
Portierung einer Maus/Tastatur-Steuerung auf eine Gamepad-Steuerung

Bachelorarbeit von René Perlbach

54 Seiten, 20 Abbildungen, 3 Tabellen

Hochschule Mittweida, University of Applied Science

Fakultät Angewandte Computer- und Biowissenschaften

Studiengang Medieninformatik und Interaktives Entertainment

Erstprüfer: Prof. Dr.-Ing. Wilfried Schubert

Zweitprüfer: Bodo Pfeifer, Dipl.-inf (FH)

Abgegeben am 22.07.2016

Thema der Bachelorarbeit

Portierung einer Maus/Tastatur-Steuerung
auf eine Gamepad-Steuerung

Stichworte

Gamepad, Benutzerschnittstelle, Spieleentwicklung, Prototypentwicklung, Unity, C#, C++

Zusammenfassung

Ziel dieser Arbeit ist eine existierende Benutzerführung, die auf einer Kombination von Maus und Tastatur basiert, auf ein Gamepad-Eingabegerät zu übertragen. Als Benutzerführung gilt hierbei sowohl das grafische User Interface als auch die haptische Bedienung. Als Gamepad gilt eine Variante mit 2 Thumbsticks, einem Directional-Pad und mindestens 8 zusätzlichen Buttons. Die dazu entstehenden Aufgaben teilen sich in Analyse der existierenden Bedienung, Konzeptionierung und Implementierung der neuen Steuerung auf. Im analytischen Teil wird untersucht, in welchen Situationen bestimmte Eingabearten getätigt werden.

Basierend auf der Analyse wird ein Konzept ausgearbeitet, um Funktionen der Software mit dem Gamepad abzubilden. Eine wichtige Limitierung dabei ist ein möglichst minimaler Eingriff in die existierende Nutzerführung und Oberflächengestaltung. Branchenübliche Bedienstandards für Gamepad User Interfaces werden dabei entsprechend berücksichtigt und dargelegt. In der Implementierungsphase wird das entworfene Konzept in die Software eingearbeitet. Im Vorfeld wird die programmiertechnische Struktur der existierenden Maus- und Tastatursteuerung untersucht. Basierend auf den Erkenntnissen wird ein Implementationskonzept entworfen und schlussendlich umgesetzt.

Title of the paper

Porting a mouse/key board control to a game pad control

Keywords

Game pad, User interface, Game development, Prototyping, Unity, C#, C++

Abstract

The focus of this paper is on porting an existing control scheme, based on using a mouse and a keyboard, to a gamepad control scheme. This paper is divided into analysis of the existing control scheme, conception and implementation of the new one.

The analytic focused part is about testing specific input situations of the software as well as analyzing the program structure of the existing mouse and keyboard implementation.

Based on the results an implementation concept for a gamepad control scheme was created. An important point of this work is the minimal influence on the existing user interface design. Standards of interface design and usability was analyzed and explained.

Finally, based on the concept, the gamepad control scheme was implemented.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Motivation	1
1.2 Über FAKT	2
1.3 Die Crazy Machines-Reihe	2
1.4 Crazy Machines 3	2
1.5 Gamepads	3
1.5.1 Definition	3
1.5.2 Geschichte	3
2 X-Box 360-Controller	5
2.1 Über das Eingabegerät	5
2.2 Aufbau	6
3 Analyse von Crazy Machines 3	10
3.1 Menü	11
3.2 Solve-Mode	12
3.3 Machine-Editor	13
3.4 Part-Editor	14
3.5 Ergebnis der Analyse	14
4 Prototyp-Entwicklung mit Unity	15
4.1 Über Unity	15
4.2 Kosten und Nutzungsbedingungen	16
4.3 Gamepad-Prototyp für Crazy Machines 3	17
4.3.1 Anforderungen des Prototyps	17
4.3.2 Funktionen des Prototyps	17
4.3.3 Prototyp-Auswertung	23
5 Umsetzung mit C++ in der FAKTUM Engine	24
5.1 Über C++	24
5.2 Über die FAKTUM Engine	24

6 Gamepad-Funktionen	24
6.1 Schnittstelle	24
6.2 Kamerasteuerung	27
6.3 Manipulation von Objekte im 3D-Raum	28
6.4 User Interface	30
6.4.1 Auswertung und Steuerung	30
6.4.2 Listen	34
6.4.3 Ringmenü	36
6.4.4 Editoren	42
6.5 Interaktive Eingabemöglichkeiten	46
7 Gesamtauswertung	47
Fremdwortverzeichnis	48
Abkürzungsverzeichnis	50
Quellenangabe	51
Verwendete Tools	52
Ehrenwörtliche Erklärung	53
Anhang	54

1 Einleitung

Die moderne Softwareentwicklung hat viele Facetten. Einige Beispiele wären: Anwendungssoftware, für die Industrie, Ausbildungssysteme für die Medizin oder Computerspiele zur Unterhaltung. Besonders der Bereich der Computerspiele konnte in den letzten Jahren einen deutlichen Wachstum aufweisen. Der schnelle Vertrieb über Plattformen wie Steam und die Entwicklungsorientierung für mobile Geräte und verschiedene Konsolen-Systeme unterstützte dies stark. Deswegen ist es mittlerweile Standard, dass viele Computerspiele schon Möglichkeiten anbieten, um nicht nur mit Maus und Tastatur bedient zu werden. So kann die Unterhaltungssoftware später schneller auf eine Konsolen-Version abgeändert werden und ein breiteres Spektrum an Nutzern wird erreicht. Zu dem ist es nicht nur für Konsolen-Spieler üblich, Gamepads zu verwenden. Der Trend vom Büro-artigem Computersystem geht immer mehr zum Entertainment-System vor der Couch oder der direkte Sprung in die virtuelle Welt wird über eine Virtual Reality (VR)-Brille ermöglicht. Bei diesen modernen Spielweisen bietet sich ein Gamepad eher an, als die klassische Maus und Tastatur-Steuerung, da die Orientierung auf dem Eingabegerät für die meisten Nutzer einfacher und intuitiver gestaltet ist. Gamepads sind für die Hand optimiert und die wenigen Tasten lassen sich bedienen, ohne den Blick auf das Eingabegerät richten zu müssen.

1.1 Motivation

Seit 2015 entwickelt das deutsche Entwicklerstudio FAKT Software GmbH das Spiel „Crazy Machines 3“. Eine Gamepad-Steuerung darf neben zeitgemäßer Grafik, moderner Spieler-Standards, wie Erstellen und Teilen von eigenem Content und VR-Support nicht fehlen. Ursprünglich, wie die meisten seiner Vorgängerteile, wurde das Spiel zunächst für eine Maus und Tastatur-Steuerung entwickelt. Der Wunsch nach einer Gamepad-Steuerung wurde schon früh vom Publisher Daedalic Entertainment GmbH und den ersten Beta-Testern noch während der Grundlagen-Entwicklung geäußert.

1.2 Über FAKT

Das deutsche Entwicklerstudio FAKT Software GmbH wurde 1999 gegründet und ist inhabergeführt. Die Firma bietet individuelle Softwarelösungen für bekannte Unternehmen, wie z.B.: Zoo Leipzig GmbH, NVIDIA Corporation, Intel GmbH und Microsoft GmbH an und setzt auch eigene Projektideen um.

1.3 Die Crazy Machines-Reihe

Eines der bekanntesten Projekte von der FAKT Software GmbH ist die Denkspiel-Reihe „Crazy Machines“. Im Jahre 2004 wurde „Crazy Machines – Die Erfinderverkstatt“ als erster Teil veröffentlicht. Seitdem erschienen mehrere Erweiterungen und Nachfolger wie z.B.: Crazy Machines 2, Crazy Machines Elements und Crazy Machines Golden Gears.

1.4 Crazy Machines 3

Seit 2015 entwickelt die Firma einen neuen Teil dieser Spiele-Reihe, welcher unter dem Namen „Crazy Machines 3“ bearbeitet wird. Mit zeitgemäßer 3D-Grafik, moderner UI, NVIDIA PHYSX-Technologie, leicht zugänglicher Bedienung und vielen Möglichkeiten für Nutzer, ihrer Fantasie freien Lauf zu lassen. Das Spiel soll voraussichtlich im Jahre 2016 veröffentlicht werden. Hauptaufgabe des Spieles ist das spielerische Lösen von physikalischen „Rube-Goldberg-Rätseln“ durch Einsetzen von Bauteilen.

Die besondere Neuheit in Crazy Machines 3 ist die Möglichkeit zur Erstellung von eigenen Bauteilen und das Anbinden von Logik-Systemen an diese. Dadurch können die Nutzer nicht nur eigene Rätsel aus vorgefertigten Elementen erstellen, sondern auch selbst bestimmen, wie diese Elemente aussehen und funktionieren sollen. Die Erstellten Rätsel (Maschinen genannt) und Bauteile können im Steam Workshop hochgeladen werden, damit sie netzwerkbasierend von anderen Nutzern verwendet und/oder weiterentwickelt werden können.

1.5 Gamepads

1.5.1 Definition

Ein Gamepad, Joypad oder auch Gamecontroller genannt, ist ein Eingabegerät, welches traditionell für die Steuerung von Computer -oder Konsolenspielen genutzt wird. Form und Anzahl der Tasten, Steuerkreuze und Analog-Sticks ist nicht festgelegt und variiert deshalb oftmals zwischen den Gamepads verschiedener Hersteller. Zu dem werden meist mit jeder neuen Generation von Gamepads neue Features hinzugefügt und für den Nutzer haptisch angenehm verwendbare Formen entwickelt.

1.5.2 Geschichte

Die ersten virtuellen Spiele, wurden noch ohne Gamepad gesteuert. Zur Bedienung dienten dabei, direkt an die Systeme installierte Steuermechaniken, wie Joysticks oder Steuereinheiten mit Nummerntasten, welche beispielsweise Arcade-Automaten verwendet haben. Diese Steuereinheiten erinnerten eher an Fernbedienungen. So hatten die Steuerpulte der Magnavox Odyssey (1972), welche als die erste, in Serie produzierte Heimkonsole behandelt wird, kaum Knöpfe und kein Steuerkreuz oder einen Analog-Stick. Stattdessen befanden sich an den Seiten des Controllers Drehknöpfe, die zur Steuerung von, heute als „Klassiker“ bezeichneten Spielen, wie beispielsweise „Pong“, genutzt wurden.



Abbildung 1: Magnavox Odyssey inklusive Controller (1972).

Erst ab 1983 führte Nintendo die Eingabemöglichkeit über Gamepad mit der Famicom-Konsole (später Nintendo Entertainment System, kurz: NES) ein. Das Gamepad dieser Konsole verfügte über ein Steuerkreuz, welches in den meisten Spielen zur Bewegung von beispielsweise Charakteren genutzt wurde und einigen anderen Buttons auf der gleichen Seite für weitere Eingabemöglichkeiten.



Abbildung 2: Nintendo Entertainment System-Gamepad (1983).

Mit dem Super Nintendo Entertainment System (SNES) wurden noch Schultertasten eingeführt. Später liesen verschiedene Entwickler von Gamepads, noch Analog-Sticks und Druck-sensible Eingabemöglichkeiten zur feineren Steuerung folgen. Das „RumblePak“ wurde ebenfalls schon als Erweiterung für Nintendo64-Controller angeboten, um Gamepads nicht nur zu einem reinen Eingabegerät zu machen, sondern auch ein gezieltes Vibrations-Feedback in Spielsituationen zu bieten. Es beinhaltet kleine Motoren, um Vibrationen zu erzeugen. Diese Vibrations-Systeme wurden von verschiedenen Entwicklern in späteren Gamepads standardmäßig integriert.



Abbildung 3: Nintendo64-Controller und N64-RumblePak (1996).

Moderne Gamepads wie das PlayStation 4-Gamepad von Sony oder der Steam-Controller von Valve unterstützen zu dem noch Berührungs-Steuerung und Systeme mit Bewegungssensoren wie die Nintendo Wii-Steuerung oder HTC Vive-Eingabegeräte werden immer mehr Standard, besonders in Verbindung mit VR-Brillen.



Abbildung 4: Steam-Controller (2015).

2 X-Box 360-Controller

2.1 Über das Eingabegerät

Im Jahre 2006 führte Microsoft mit der Spiele-Konsole X-Box 360 auch den zugehörigen X-Box-360-Controller ein. Die Kompatibilität mit einem PC war von Anfang an gegeben. Es existieren 2 Varianten von diesem Gamepad. Eine kabellose und eine mit Kabel. Die kabellose Variante wird mit Akkumulatoren oder herkömmlichen Batterien betrieben und benötigt bei der Verwendung an einem PC einen Adapter, der an dem PC per USB-Anschluss verbunden werden muss. Die Variante mit Kabel benötigt ebenfalls einen USB-Anschluss. Der Controller kann nach der Installation der zugehörigen Treiber am PC als alternative Eingabemöglichkeit von Spielen und anderer Software genutzt werden. Auch 10 Jahre nach seiner Veröffentlichung ist der X-Box360-Controller ein häufig genutztes Eingabegerät für die Steuerung von PC-Spielen.

2.2 Aufbau

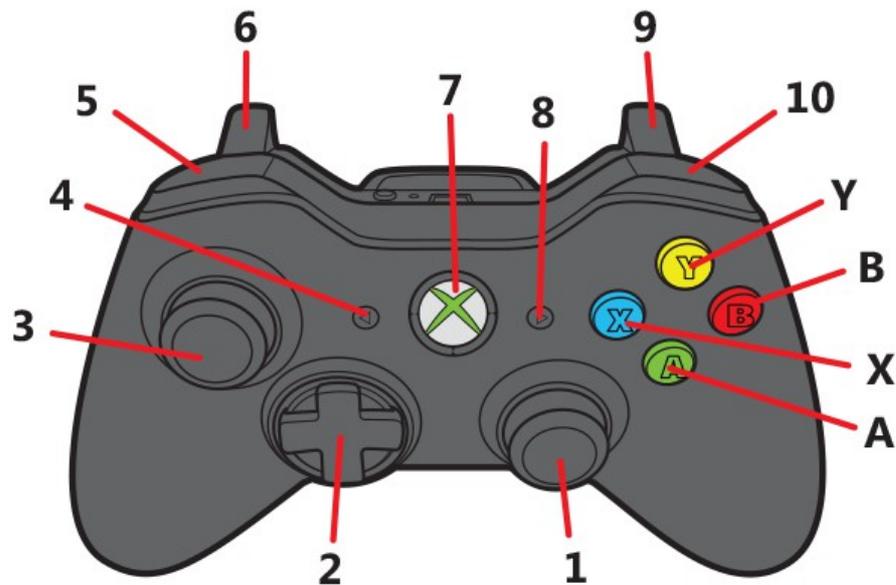


Abbildung 5: X-Box 360-Gamepad.

1 Rechter Thumbstick

Der rechte Thumbstick wird in der Regel mit dem Daumen (Engl.: Thumb) der rechten Hand bewegt. Er kann in seine X und Y-Richtung bewegt oder auch wie eine Taste per Druck aktiviert werden. Im Gegensatz zu einem herkömmlichen Steuerkreuz ermöglichen Thumbsticks eine Ausgabe von Werten, die nicht nur die Zustände Betätigt und nicht Betätigt wiedergeben. Die Achsen fragen permanent ihren aktuellen Stand ab und können je nach Stärke der Neigung des Sticks Werte ausgeben, die eine feine Steuerung ermöglichen.

2 D-Pad/Steuerkreuz

Ein klassisches Steuerkreuz ist in 4 Richtungen unterteilt. Die 4 Richtungstasten werden in etwa wie normale Eingabetasten behandelt. Es kann für jede der Richtungen der Zustand Betätigt oder nicht Betätigt gelten. Eine Besonderheit an dem Steuerkreuz ist, dass gegenüberliegende Richtungen nicht zeitgleich betätigt werden können. So ist es zwar möglich, gleichzeitig die beispielsweise obere und linke Richtung des Steuerkreuzes zu aktivieren, jedoch wäre es nicht möglich Rechts und Links oder Oben und Unten gleichzeitig zu aktivieren.

3 Linker Thumbstick

Der linke Thumbstick funktioniert wie der rechte und wird in der Regel mit dem Daumen der linken Hand bewegt.

4 Back-Taste

Die Back-Taste ist eine Eingabetaste, welche die Zustände Betätigt und nicht Betätigt erreichen kann. In den meisten Spielen wird diese Eingabetaste eher für die Funktionen verwendet, welche mit dem Menü oder dem Aktivieren von besonderen Zuständen zu tun haben. Auch die schwer zu erreichende Lage und die Beschriftung sorgen dafür, dass diese Eingabetaste sich eher schlecht für essenzielle Funktionen eignet.

5 Linke Schultertaste

Schultertasten funktionieren wie einfache Eingabetasten, können also nur den Zustand Betätigt oder nicht Betätigt einnehmen. Durch die besondere Positionierung an der oberen Seite des Gamepads, ist es dem Nutzer möglich mit dem Zeigefinger diese Tasten zu bedienen. So können neben der hauptsächlich mit den Daumen erfolgende Steuerung der Thumbsticks, Face-Buttons, Steuerkreuze usw., auch nebenbei ohne großen Aufwand, zusätzliche Aktionen getätigt werden. Dabei wird meistens mit dem linken Zeigefinger auch die linke Schultertaste bedient.

6 Linker Trigger

Trigger sind analoge Tasten, welche im Gegensatz zu gewöhnlichen binären Eingabetasten, mehr Betriebszustände, als Betätigt oder nicht Betätigt einnehmen können. In Abhängigkeit von der Druckstärke, mit der die Trigger betätigt werden, können niedrige (leichter Druck) bis hohe (starker Druck) Analog-Werte ausgegeben werden. Die Bedienung des linken Triggers erfolgt meist mit dem linken Zeigefinger oder Mittelfinger.

7 Guide-Taste

Diese Taste funktioniert wie eine einfache Eingabetaste. Die Besonderheit dieser Taste liegt weniger in der Bedienung, sondern mehr in der Anbindung an das jeweilige System. Bei Benutzung der X-Box werden mit Hilfe dieser Taste Funktionalitäten der Konsole aufgerufen. Wiederum an einem PC angeschlossen, kann mit dieser Taste während eines Spieles, unter anderem das Steam-Overlay aufgerufen werden. So wird diese Taste weniger im Spiel genutzt, hat jedoch für die im Hintergrund laufenden Systeme einen entscheidenden Nutzen.

8 Start-Taste

Die Start-Taste befindet sich etwas nach rechts versetzt in der Mitte des Gamepads. Hierbei handelt es sich um eine einfache Eingabetaste. Meistens wird diese Taste genutzt, um Spiele zu pausieren oder Menüs aufzurufen.

9 Rechter Trigger

Der Rechte Trigger funktioniert wie der Linke. Er gibt Analog-Werte aus, in Abhängigkeit von der Druckstärke, mit der er bedient wird. Meistens wird der rechte Trigger mit dem rechten Zeigefinger oder Mittelfinger betätigt.

10 Rechte Schultertaste

Die rechte Schultertaste ist wie die linke Schultertaste eine am oberen Gamepad-Rand präsente Eingabetaste. Im Normalfall wird die rechte Schultertaste mit dem rechten Zeigefinger bedient.

Face Buttons

Als Face-Buttons werden die Tasten auf einem Gamepad bezeichnet, die bei einer typischen Art, das Eingabegerät zu halten, für den Nutzer direkt sichtbar sind.

Bei dem X-Box360-Controller sind die Face-Buttons mit den Buchstaben A, B, X und Y in den Farben Grün, Rot, Blau, Gelb dargestellt und auf der rechten, oberen Seite positioniert.

A Unterer Face-Button/A-Taste

Die A-Taste kann 2 Zustände erreichen: Betätigt oder nicht Betätigt. Sie ist eine der am häufigsten genutzten Eingabetasten. In Spielen werden mit ihr meistens oft zu nutzende Aktionen aufgerufen. Zu dem wird diese Taste oft in Menüs für Funktionen zum Bestätigen einer Auswahl genutzt. Auch in Dialogen wird diese Taste meist zum Bestätigen verwendet. Zudem ist sie mit dem Buchstaben A gekennzeichnet, welcher im Alphabet an erster Stelle steht und sie ist grün dargestellt. Dadurch hat die A-Taste eine positives oder bestätigendes Erscheinen und kommt in der Rangordnung der Face-Buttons zu erst. Sie wird in der Regel mit dem rechten Daumen bedient.

B Rechter Face-Button/B-Taste

In den meisten Spielen und auch Anwendungssystemen, wie Steam im Big Picture Mode, wird bei einer Gemapad-Steuerung die B-Taste oft für eine Abbruch-Aktion genutzt. Eine rote Färbung unterstützt diesen Abbruch-Charakter ebenfalls. Die Funktionalität des Buttons ist wie bei allen anderen Face-Buttons.

X Linker Face-Button/X-Taste

Ein weiterer Face-Button ist auf dem X-Box-Controller die X-Taste. Diese Taste funktioniert, wie die anderen Face-Buttons und wird in Spielen oft für häufige Aktionen oder Zusatzfunktionen genutzt, wenn die A-Taste bereits belegt ist.

Y Oberer Face-Button/Y-Taste

Der obere Face-Button ist auf dem X-Box-Controller mit Y gekennzeichnet. Er funktioniert wie die anderen 3 Face-Buttons und ist für den Nutzer gut erreichbar.

3 Analyse von Crazy Machines 3

Bevor mit der Prototyp-Entwicklung und letztendlich auch der finalen Umsetzung zur Portierung der Steuerung begonnen werden konnte, musste analysiert werden, welche Funktionen das Spiel überhaupt besitzt. Grundsätzlich konnten die Funktionen des Spiels in 4 Bereiche eingeordnet werden.

3.1 Menü

Das Menü dient hauptsächlich zur Verwaltung von Inhalten und Einstellungen des Spieles. Hier können Level (Maschinen genannt) geladen oder gelöscht werden. Die Maschinen unterteilen sich in die 2 Typen, der nur spielbaren oder der spielbaren und editierbaren Maschinen. Die nur spielbaren Maschinen werden im Solve-Mode geöffnet, wobei die editierbaren Maschinen im Machine-Editor geöffnet werden. Grundlegende Spiele-Informationen, wie die Steuerung des Spieles, die Namen der Entwickler des Spieles, sowie die grundlegenden Spiele-Optionen, sind über das Menü ebenfalls erreichbar.

Aufbau und Bedienung des Menüs erinnern dabei an eine moderne Verwaltungsanwendung oder einen Online-Shop, wirken dadurch etwas weniger spielerisch, sind aber dadurch für die meisten Nutzer intuitiv bedienbar. Die verschiedenen Möglichkeiten im Menü sind in Tabs eingeteilt, die einfach jeder Zeit umgeschaltet werden können. Ähnlich aufgebaute Menü-Systeme sind in aktuellen Spielen, wie beispielsweise Dota 2 oder Anwendungs-Verwaltungssystemem, wie beispielsweise Steam, ebenfalls zu finden. Gerade, weil „Crazy Machines 3“ über Steam vertrieben wird, ist das Navigieren mit Hilfe von Tabs für den Anwender sofort und leicht verständlich.

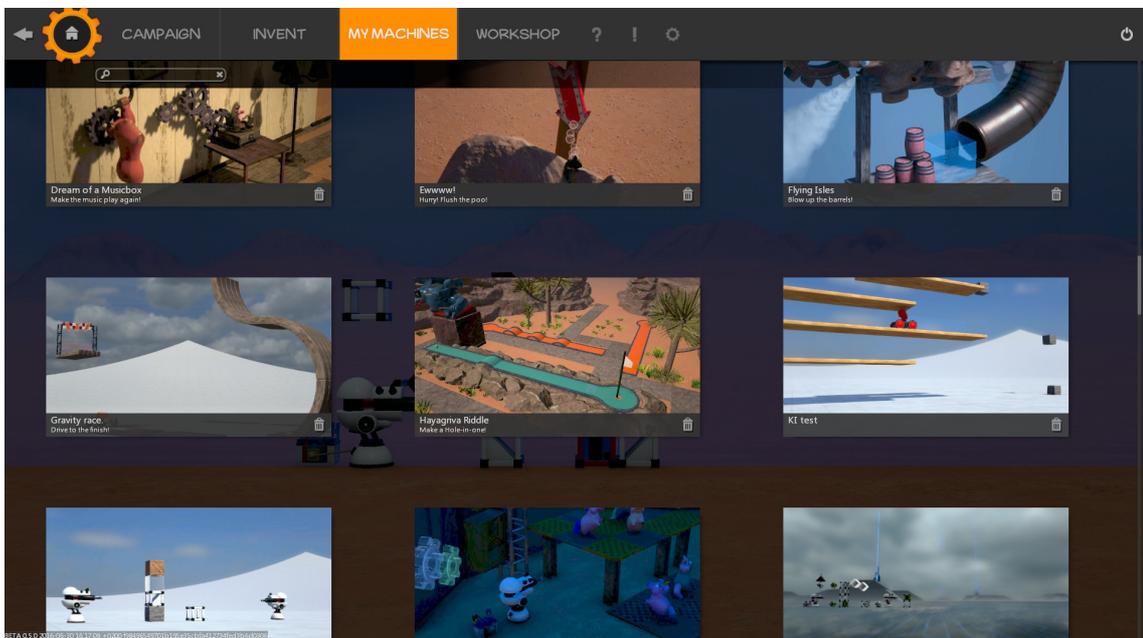


Abbildung 6: Die „My Machines“-Sektion im Menü.

3.2 Solve-Mode

Im Solve-Mode wird das eigentliche Spielprinzip repräsentiert. Schon wie in allen vorherigen „Crazy Machines“ Teilen, muss hier mit Hilfe von verschiedenen Objekten ein Physik-Rätsel gelöst werden. Der Nutzer kann Objekte aus seinem Inventar, welches am linken Bildschirmrand dargestellt ist, auswählen und diese in der Spielwelt platzieren. Per Mausklick auf den Play-Button, wird die Simulation gestartet und alle Systeme in der Maschine fangen an zu Arbeiten. So wird beispielsweise die Gravitation aktiviert und Objekte fallen herunter, Strom fließt oder Motoren fangen an zu Laufen. In jeder Maschine muss eine bestimmte Aufgabe erfüllt werden. Dies kann das einfache Betätigen eines Schalters sein oder das lösen eines komplexen, physikalischen Rätsels. Wenn die Bedingungen zur Lösung erfüllt sind, wird die Aufgabe in der Maschine als gelöst angesehen.

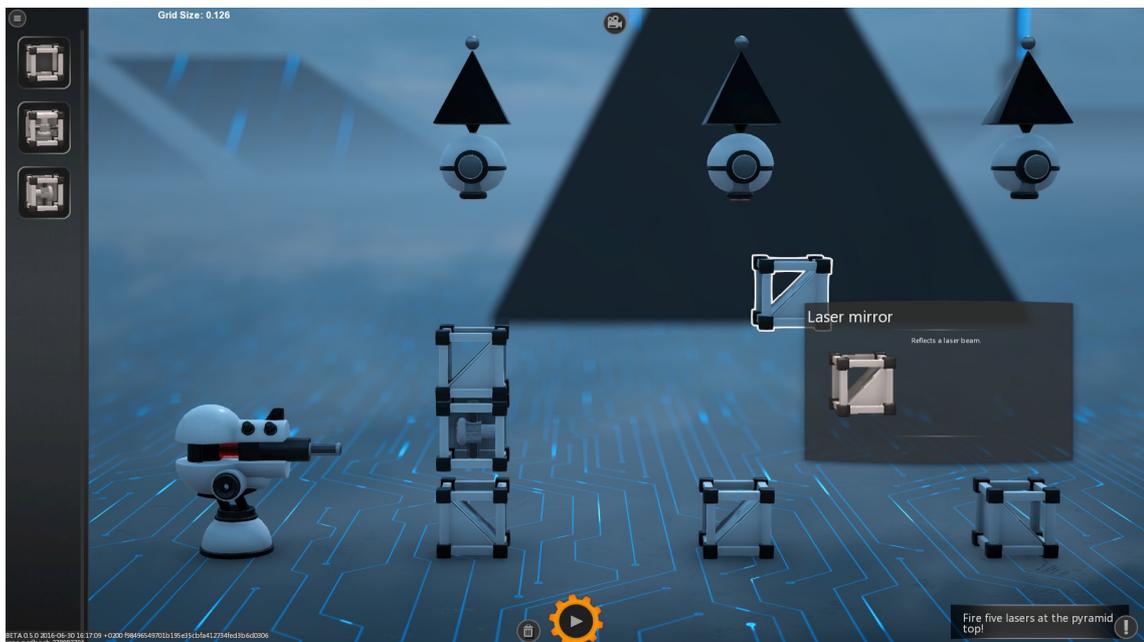


Abbildung 7: Solve-Mode.

3.3 Machine-Editor

Im Machine-Editor können Nutzer ihre eigenen Maschinen erstellen und diese auch im Steam-Workshop mit anderen Nutzern teilen. Hier stehen dem Nutzer alle Bauteile, die im Basis-Spiel vorhanden sind, aus dem Workshop heruntergeladen oder selber erstellt wurden, in einem Katalog zur Verfügung. Diese Bauteile können frei in der Spielwelt platziert und deren Größe oder Rotation verändert werden. So können unter anderem Kugelbahnen, Stromkreisläufe oder dekorative Szenerien erstellt werden. Der Untergrund, die Himmelskugel und der Farbfilter der Kamera kann ebenfalls verändert werden. Zu dem stehen dem Nutzer noch Möglichkeiten für weitere Einstellungen zur Verfügung. Bauteile können gezielt als Nutzbar markiert werden, um später im Solve-Mode zur Verfügung zu stehen. Zu dem können die Bauteile auch als fest in der Welt verankert markiert werden. Dieser Zustand wird als „Static“ oder „Statisch“ bezeichnet. Statische Objekte werden nicht von Gravitation oder anderen Kräfteinwirkungen, wie Stößen oder Rotationskräften, beeinflusst. Sowie die Objekte als „Static“ markiert werden können, besteht auch die Möglichkeit sie mit der „Freeze“-Funktion zu markieren. Dabei wird die Bewegung von Objekten auf eine Ebene begrenzt. Diese Funktion verhindert, dass beispielsweise Kugeln oder Fahrzeuge sich außerhalb der Spielebene bewegen können.

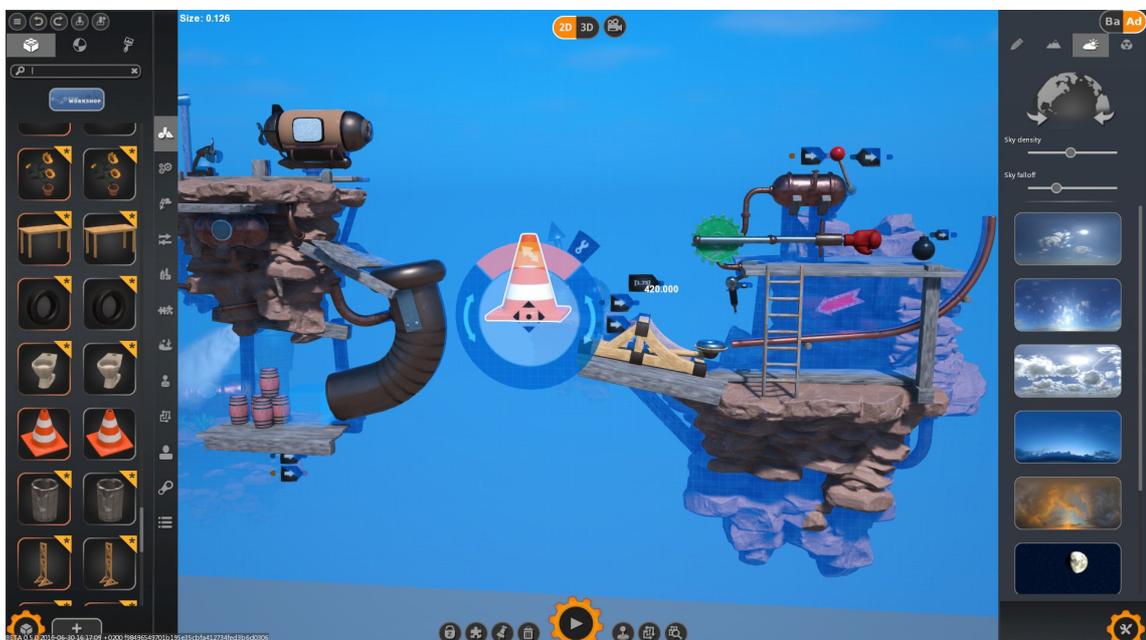


Abbildung 8: Machine-Editor.

3.4 Part-Editor

Der Part-Editor taucht noch tiefer in die funktionalen Ebenen des Spieles ein. Er gibt die Möglichkeit, aus 3D-Objekten, Effekten, Sounds und Logik-Teilen, eigene Bauteile zu erstellen. Auch hier können die Grund-Elemente skaliert oder rotiert werden.

Die verschiedenen Elemente werden miteinander über so genannte Snap-Punkte, entweder frei oder an speziellen Stellen, miteinander verbunden.

Zu dem kann entschieden werden, wie die Elemente miteinander verbunden sind.

Verbindungen können beispielsweise Gelenke, Achsen oder feste Verbindungen sein.

Farbe und Material der Elemente sind im Part-Editor ebenfalls anpassbar und es

besteht auch hier die Möglichkeit, eigene Bauteile im Steam-Workshop hochzuladen.

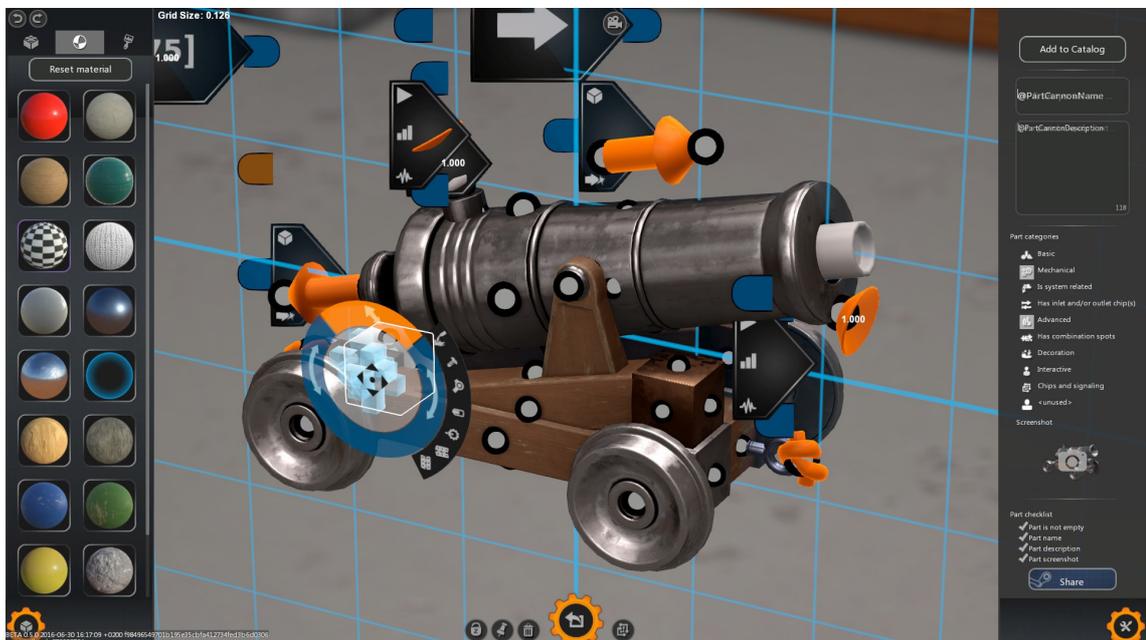


Abbildung 9: Part-Editor.

3.5 Ergebnis der Analyse

Der Umfang an Funktionalitäten von „Crazy Machines 3“ ist sehr groß. Bei der Umsetzung einer Gamepad-Steuerung für das Spiel, muss einerseits jede Funktion abgedeckt werden und andererseits auch die Nutzbarkeit einfach, schnell und intuitiv erfolgen. Die Unterteilung des Spieles in verschiedene Editoren ist dabei sehr hilfreich. Bei der Maus-Tastatur-Steuerung waren in den jeweiligen Editoren nahezu alle Funktionen immer erreichbar. Es wurde entschieden, dass bei der Gamepad-

Steuerung alles noch weiter in Ebenen unterteilt werden könnte, um die Bedienung zu staffeln. So sollte es beispielsweise möglich werden, direkt im Katalog oder im 3D-Raum zu navigieren, aber nicht zeitgleich in beiden Eingaberegionen. Bei der Analyse wurde erkannt, dass die UI für die Gamepadsteuerung entsprechend angepasst werden muss. Dabei sind alle grafischen Tasten, die üblicherweise mit der Maus bedient werden, durch Gamepad-Eingabetasten zu ersetzen. Die Funktionen der grafischen Tasten müssen auf Gamepad-Tasten gelegt werden. Eine Informationsanzeige zur Steuerung wird dabei am unteren Bildschirm-Rand erklären, wann und wie welche Funktion erreichbar sind. Somit wurde die gesamte Toolbar, die sich bei der Maus-Tastatur-Steuerung am unteren Bildschirm-Rand befindet überflüssig und könnte entfernt werden. Zu dem sollte gerade bei der UI-Änderung berücksichtigt werden, dass Icons und Schriften größer dargestellt werden, als in der ursprünglichen Spiel-Steuerung. Dies wäre für die spätere Konsolen-Varianten von „Crazy Machines 3“ sehr hilfreich, da gerade weil beim Spielen über Konsole an einem TV-Gerät der Nutzer meistens deutlich weiter vom Bildschirm entfernt sitzt, als bei der Nutzung eines PCs. Trotz des Aufwandes und dem Umfang an Funktionen, wurde sich für eine Umsetzung der Gamepad-Steuerung entschieden, da in der Vergangenheit bereits Crazy-Machines-Teile für Sony Playstation3 und Nintendo Wii umgesetzt wurden.

Ein Vorbild für die Umsetzung war auch Little Big Planet von Sony. Dieses Spiel hat einen Level-Editor, der dem Machine-Editor sehr ähnlich ist und als eines der bekanntesten Spiele seiner Art, konnte es im Bereich Usability als Vorbild dienen.

4 Prototyp-Entwicklung mit Unity

4.1 Über Unity

Unity ist eine Spiel-Engine für 3D -und 2D-Anwendungen und wird primär im Bereich der Unterhaltungssoftware genutzt. Sie findet aber auch in Bereichen Architektur, Inneneinrichtung und zur Darstellung von Simulationen Verwendung. Der Editor der Unity-Engine bringt Darstellungsmöglichkeiten für 2- sowie 3-dimensionale Objekte und Texte mit. Mit ihm können 3D-Modelle, ganze Szenen, Grafiken, Texturen, Normal-Maps, Programmiercode und Ordnerstrukturen angezeigt und bedient werden. Über verschiedene Kontextmenüs ist es möglich, Objekte und deren Eigenschaften anzulegen. Per Drag-and-Drop Bedienung ist es möglich, neue Ressourcen zu

importieren. Zum Editieren von Programmiercode liefert Unity einen Texteditor namens MonoDevelop mit. Es werden aber auch andere Tools, wie zum Beispiel Visual Studio, unterstützt. Die Engine enthält bereits viele Grundfunktionen wie Rendering von 3D-Objekten, Licht-und-Schattendarstellung, physikalische Grundfunktionen und 3-dimensionale Grundformen. Mit diesen, in der Engine bereits enthaltenen Grundfunktionen, wird die Arbeit des Softwareentwicklers erheblich erleichtert, da nicht jedes Feature selbst neu entwickelt werden muss. Bei der professionellen Nutzung von Unity ist es ausreichend, den für die Anwendung spezifischen Programmiercode zu erstellen und die Objekte in der 3D/2D-Szenerie zu positionieren.

Die Unity-Engine ist ein weit verbreitetes Spiele- und Software-Entwicklungswerkzeug, womit bereits namhafte Spieletitel entwickelt wurden. Sie eignet sich nicht nur für die Entwicklung kompletter Spiele, sondern wird sehr oft auch nur für Prototypen von Spielkonzepten oder einzelnen Funktionen genutzt. Die Verwendung der Engine für derartige Zwecke lässt keinerlei Anschaffungskosten anfallen. Sie ermöglicht dank guter Grundfunktionen im System und optimaler C#-Anbindung, eine schnelle Prototyp-Entwicklung. Genau diese Engine wurde benutzt, um für „Crazy Machines 3“ einen Gamepad-Prototypen zu entwickeln.

4.2 Kosten und Nutzungsbedingungen

Unity existiert in 2 Formen. Die kostenlose Personal Version und die kostenpflichtige Pro Version. Grundlegende Funktionen der Engine sind bei beiden Versionen gleich, jedoch die Pro Version bringt noch zusätzliche Features und Berechtigungen, wie Multi-User-Verwaltung und Exportfunktionen, mit sich.

Die Unity-Engine kann frei für nichtkommerzielle Zwecke genutzt werden, beispielsweise: Studentenprojekte oder Prototypen für Software.

Es ist dabei zu beachten, dass Entwickler, die durch eine mit Unity entwickelte Software Gewinne erwirtschaften, ab einem bestimmten Geldbetrag Abgaben an Unity Technologies leisten müssen.

4.3 Gamepad-Prototyp für Crazy Machines 3

4.3.1 Anforderungen des Prototyps

In Vorbereitung auf die Weiterentwicklung des Spieles, hinzu modernen Eingabegeräten, war es erforderlich, vorab ein Gamepad-Prototyp zu entwickeln. Dieser Prototyp sollte ein gewisses Anfühlen an das Spiel „Crazy Machines 3“ für elementare Grundfunktionen, wie Kamerabewegung, Bauteil-Auswahl, Platzierung und Bewegung im Maschineneditor und Navigation im Katalog ermöglichen. Diese Grundfunktionen wurden dabei in Unity nachgebaut und geprüft.

Im speziellen sollte Augenmerk, auf die 2-dimensionalen Bewegungen für die Objekt- und Kameranavigation, sowie das 3-dimensionale Rotieren der Kamera gelegt werden. Zudem sollten schon einige Funktionen, wie Rotieren, Skalieren, Verschieben der Objekte und Navigieren im Katalog prototypisch nachgebaut werden, um im Vorfeld das Zusammenspiel aller Funktionen zu untersuchen.

4.3.2 Funktionen des Prototyps

Direkt nach dem Start der entwickelten Prototyp-Software, werden zur Hilfestellung einige Informationen zur Steuerung eingeblendet. In runden Klammern wird die Steuerung per Gamepad erklärt, in eckigen Klammern sind Eingabemöglichkeiten per Tastatur erkennbar. Zudem werden noch Variablen für aktuell selektierte Objekte und Geschwindigkeit für Kamerarotation und Bewegung dargestellt.



Abbildung 10: Startbildschirm und Informationen zur Steuerung des Prototyps.

Kamerabewegung

Die Kamerabewegung erfolgt mit dem linken Thumbstick. Wenn dieser auf seiner X-Achse bewegt wird, kann die Kamera nach Links oder Rechts eine Bewegung ausführen. Eine Bewegung des Thumbsticks auf seiner Y-Achse lässt die Kamera nach Oben und Unten fahren.

Kamerarotation

Mit dem rechten Thumbstick lässt sich die Kamera rotieren. Dabei fokussiert sie ständig einen Punkt, um den sie Rotiert.

Kamerazoom

Die Trigger-Tasten ermöglichen das Heranzoomen, wenn kein Objekt selektiert ist. Dabei wird die Kamera auf den fokussierten Punkt, um den auch die Rotation stattfindet, zubewegt, bis ein Mindestabstand erreicht wird.

Panels öffnen

Im Prototyp wird das Öffnen von Panels am linken und rechten Bildschirmrand mit den Schultertasten ermöglicht. So kann mit der linken Schultertaste der Katalog erreicht werden. Mit der rechten Schultertaste lässt sich ein leeres Panel, was nur exemplarisch zur Darstellung des Eigenschaftfensters dient, öffnen.

Katalog navigieren

Die Navigation im Katalog erfolgt mit dem linken Thumbstick. So kann aus einer Anzahl an Objekten ausgewählt werden. In einem 3 x 6 Felder großen Raster werden Slots mit Text, welcher die Namen der Objekte beinhaltet, angezeigt. Die aktuelle Position des Cursors, während der Auswahl eines Objektes im Katalog, lässt sich durch eine grau hervorgehobene Fläche erkennen.

Objekte aus Katalog nutzen

Durch das Betätigen des A-Buttons kann ein Objekt aus dem Katalog ausgewählt und in der 3D-Umgebung erzeugt werden. Dabei wird das Katalogpanel sofort geschlossen und das erzeugte Objekt kann in der Umgebung platziert werden. Das Akzeptieren der Position des Objektes erfolgt ebenfalls mit der Betätigung des A-Buttons.

Objekte auswählen

Befindet sich ein Objekt aus der Szene in der Mitte des Bildschirms, so wird es für diesen Moment mit einer farbigen Markierung hervorgehoben. Dadurch kann genau erkannt werden, welches Objekt gerade selektiert werden kann.

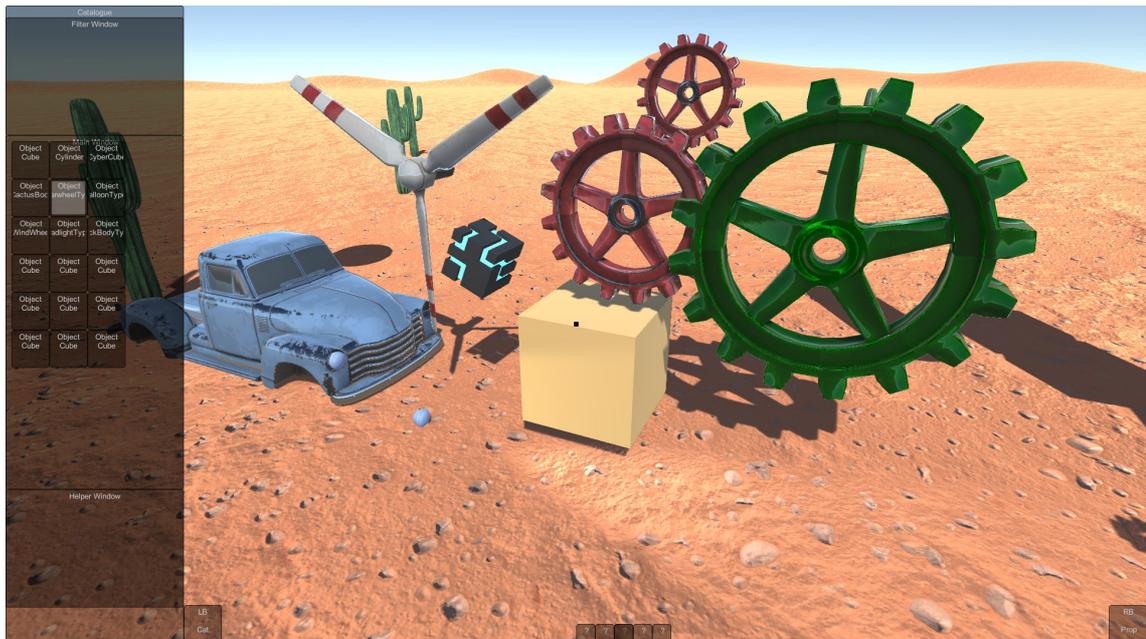


Abbildung 11: Katalogpanel (Links) und ein hervorgehobenes Objekt (Mitte) im Prototyp.

Objekte aufnehmen

Wenn ein Objekt hervorgehoben wurde, ist die Möglichkeit gegeben, es mit der Betätigung des A-Buttons aufzunehmen. Ist dies geschehen, so kann es anschließend verschoben werden. Des Weiteren besteht die Möglichkeit ein Ringmenü aufzurufen, welches zusätzliche Manipulationsmöglichkeiten wie Rotieren und Skalieren des Objektes ermöglicht.

Ringmenü

Ist der Fall gegeben, dass ein Objekt gerade ausgewählt ist, kann mit der Betätigung des Y-Buttons das Ringmenü aufgerufen werden. Durch die Bewegung des linken Thumbsticks ist es möglich, eine Auswahl zu treffen. So kann das Verschieben, Rotieren und Skalieren eines aktiven Objektes eingestellt werden. Ein Ringmenü hat viele Vorteile. So können viele Optionen schnell ausgewählt werden und die kreisförmige Anordnung der Auswahlmöglichkeiten lässt sich intuitiv bedienen, weil die Bewegung des Sticks in die jeweilige Richtung auch grafisch zusammenpassend angezeigt wird.

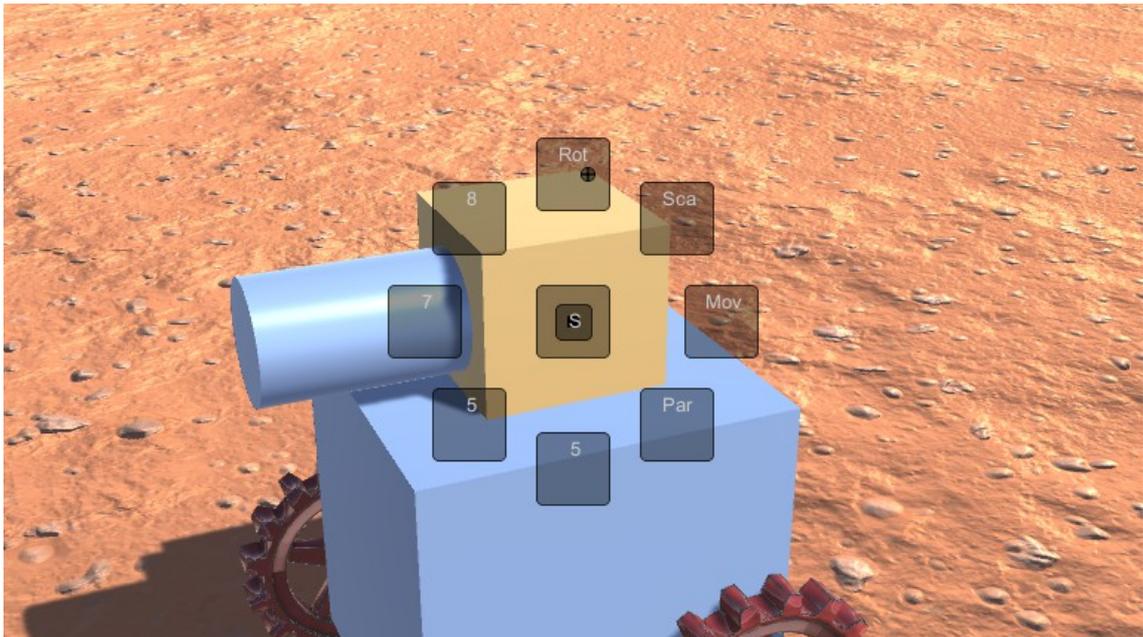


Abbildung 12: Ringmenü im Prototyp.

Objekte verschieben

Wenn ein Objekt durch Betätigung des A-Buttons aufgenommen wurde, kann es mit Hilfe des linken Thumbsticks bewegt und entsprechend platziert werden.

Wenn im Ringmenü die Auswahl „Mov“ selektiert wurde, kann das Objekt durch Nutzung der Trigger-Tasten ebenfalls in der Tiefe verschoben werden. Dank dieser Möglichkeit kann man Objekte auch im Hintergrund platzieren.

Objekte rotieren

Im Ringmenü kann die Möglichkeit zur Rotation von Objekten aktiviert werden, indem die Auswahl auf „Rot“ gestellt wird. Ist dies geschehen, kann das aktive Objekt mit den Trigger-Tasten nach Links oder Rechts gedreht werden.

Objekte skalieren

Das Skalieren von Objekten funktioniert genau wie das Rotieren mit den Trigger-Tasten. Dabei können Minimal- und Maximallimits erreicht werden. Das Maximum bei der Skalierung beträgt 300% der Ausgangsgröße, das Minimum, welches erreicht werden kann ist 10%.

Objekte einfärben

Im Prototyp wurden schon einige Features, wie das Einfärben von Objekten implementiert. Am unteren Bildschirmrand befinden sich mit Fragezeichen gekennzeichnete Felder, welche die Toolbar repräsentieren. Mit dem D-Pad kann man zwischen den Feldern navigieren. Die meisten Toolbar-Felder sind für den Prototypentest nur exemplarisch dargestellt, doch die Funktion zum Umfärben wurde hier schon erreichbar gemacht.

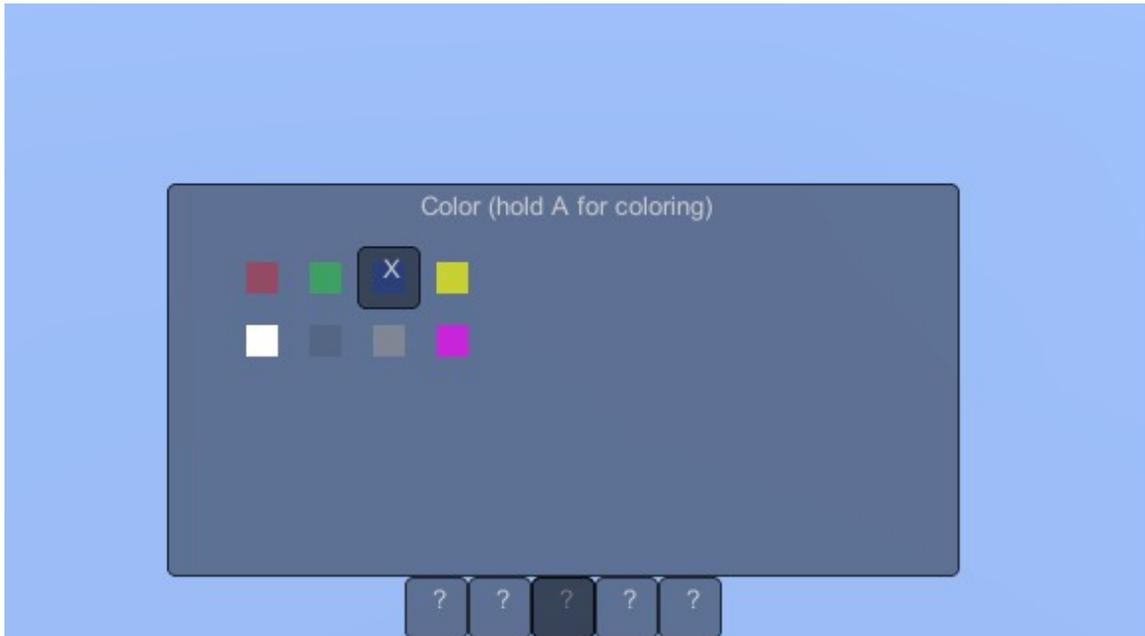


Abbildung 13: Nah-Ansicht der Farbauswahl.

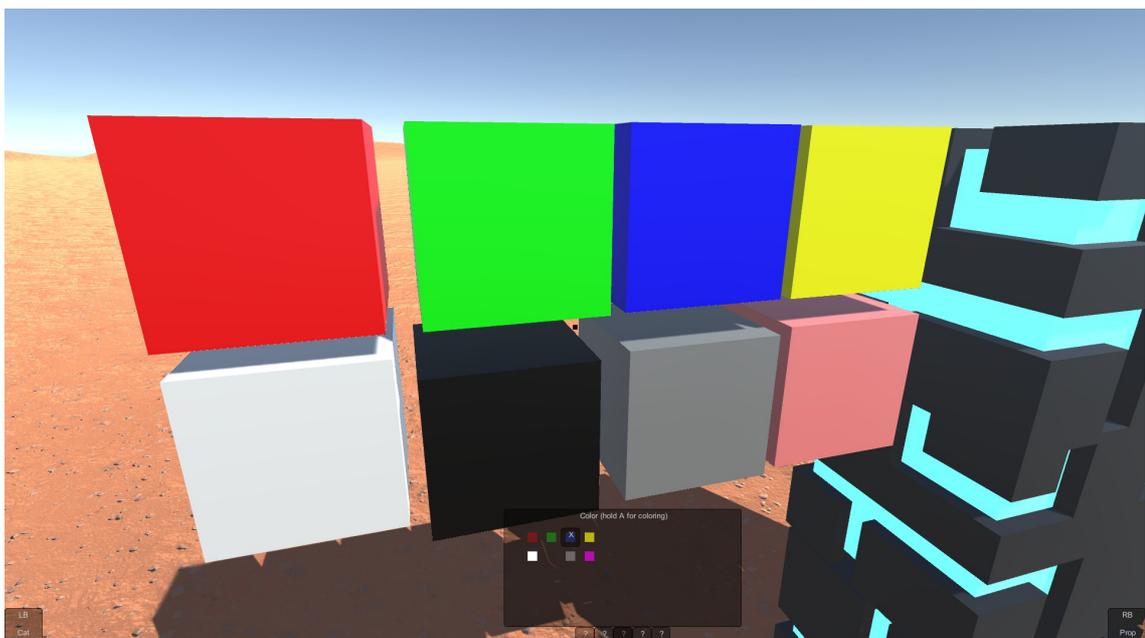


Abbildung 14: Auswahl der Farben (Unten) und einige umgefärbte Objekte.

4.3.3 Prototyp-Auswertung

Die Prototypen-Analyse hat nachgewiesen, dass eine Gamepad-Steuerung von „Crazy Machines 3“ auf jeden Fall umsetzbar ist. Gerade im Bereich der Kameranavigation und Objektbewegung im 3-dimensionalen Raum gab es keine Unklarheiten.

Das Navigieren und Umsetzen der vielen Möglichkeiten der UI und die gesamte Tool-Auswahl wurde im Prototyp nur teilweise zum Anfühlen der Steuerung umgesetzt.

Das Ringmenü hat sich bereits in vielen anderen Konsolen-Spielen bewährt. Deshalb konnte eingeschätzt werden, dass es auch sehr gut für „Crazy Machines 3“ geeignet ist und soll es weiterhin genutzt werden. Es wurde festgestellt, dass die Toolbar bei dem Einsatz der Gamepad-Steuerung unkomfortabel bedienbar ist und sollte deswegen ganz ausgeblendet und ebenfalls durch ein ein Ringmenü ersetzt werden. Weitere Anpassungen für die Kamerasteuerung wurden für den Prototyp als annehmbar eingestuft, sollten aber für die finale Version so überarbeitet werden, dass der Nutzer eine weichere Steuerung vorfinden kann. Das Auswerten der Thumbsticks im Prototyp erfolgte linear, d.h.: der aktuelle Wert wird direkt verrechnet und in eine Kamerabewegung umgewandelt. Eine quadratische Funktion für die Werteberechnung würde eine weiche Steuerung unterstützen. So könnte sich die Kamera bei einer feinen Bedienung der Thumbsticks nur langsam bewegen und ein Anschlag bis zum Maximum eine deutlich schnellere Bewegung hervorrufen.

5 Umsetzung mit C++

5.1 Über C++

C++ ist eine Programmiersprache, die sich aus der Programmiersprache C heraus entwickelt hat. Sowohl in der Systementwicklung, als auch in der Anwendungsentwicklung vertreten, ist C++ eine der am häufigsten genutzten Programmiersprachen.

5.2 Über die FAKTUM-Engine

Die FAKTUM-Engine ist eine von FAKT in C++ entwickelte Programmierstruktur, welche der Firma bei der schnellen und fehlerfreien Umsetzung von Software behilflich ist. Sie existiert schon seit 10 Jahren und wird stetig weiterentwickelt.

Speziell für die Entwicklung von interaktiven Anwendungen erstellt, beinhaltet die Engine nützliche Funktionen wie beispielsweise Rendern von Grafiken, Berechnen von Physik, Lesen und Schreiben von Daten und das Auswerten von XML aber auch XAML-Code. Zudem verschlüsselt und komprimiert die Engine auch Dateien, um diese später in einer Anwendung effizient laden zu können, sowie auch das unerlaubte Bearbeiten zu verhindern.

6 Gamepad-Funktionen

6.1 Schnittstelle

Gamepad-Tasten

Die jeweiligen Keycodes von angeschlossenen Gamepads können entsprechend ausgewertet werden. Dabei übergeben sie für jedes Eingabeelement einen entsprechenden Wert. Dieser Wert kann einer binären True/False-Ausgabe oder einem analogen Fließkommazahlen-Ausgabe von -1 bis 1 bzw. 0 bis 1 entsprechen. Die ausgelesenen Werte werden zur besseren Übersicht in folgende Variablen übertragen:

X-Box-Controller	Variablen	Mögliche Werte bei Abfrage
Y	KEY_GAMEPAD_FACE_BUTTON_TOP	0 oder 1
A	KEY_GAMEPAD_FACE_BUTTON_BOTTOM	0 oder 1
X	KEY_GAMEPAD_FACE_BUTTON_LEFT	0 oder 1
B	KEY_GAMEPAD_FACE_BUTTON_RIGHT	0 oder 1
LB	KEY_GAMEPAD_SHOULDER_LEFT	0 oder 1
RB	KEY_GAMEPAD_SHOULDER_RIGHT	0 oder 1
Linker Thumbstick	KEY_GAMEPAD_THUMBSTICK_LEFT	0 oder 1
Rechter Thumbstick	KEY_GAMEPAD_THUMBSTICK_RIGHT	0 oder 1
D-Pad Oben	KEY_GAMEPAD_DPAD_UP	0 oder 1
D-Pad Unten	KEY_GAMEPAD_DPAD_DOWN	0 oder 1
D-Pad Links	KEY_GAMEPAD_DPAD_LEFT	0 oder 1
D-Pad Rechts	KEY_GAMEPAD_DPAD_RIGHT	0 oder 1
Back	KEY_GAMEPAD_SPECIAL_LEFT	0 oder 1
Start	KEY_GAMEPAD_SPECIAL_RIGHT	0 oder 1
Linker Trigger	AXIS_GAMEPAD_TRIGGER_LEFT	0 bis +1f
Rechter Trigger	AXIS_GAMEPAD_TRIGGER_RIGHT	0 bis +1f
X-Achse linker Thumbstick	AXIS_GAMEPAD_THUMBSTICK_LEFT_X	-1f bis +1f
Y-Achse linker Thumbstick	AXIS_GAMEPAD_THUMBSTICK_LEFT_Y	-1f bis +1f
X-Achse rechter Thumbstick	AXIS_GAMEPAD_THUMBSTICK_RIGHT_X	-1f bis +1f
Y-Achse rechter Thumbstick	AXIS_GAMEPAD_THUMBSTICK_RIGHT_Y	-1f bis +1f

Tabelle 1: Bezeichnung, Variablen und Werte von Gamepad-Eingabemöglichkeiten.

Die Funktion **GamepadKey(KEY eKey, BOOL bPressed)** bekommt den jeweiligen, betätigten Button (KEY eKey) und den Status, ob ein Button betätigt wurde (BOOL bPressed) übergeben.

Gamepad-Achsen

Die Funktion **GetAxis(Axis eAxis)** fragt den Wert von Achseneingaben ab. Die X- und Y-Verschiebung des rechten und des linken Thumbsticks liefern dabei einen Fließkommazahlen-Wert von -1f bis 1f, die Trigger-Achsen geben einen Wert von 0f bis 1f aus.

Deadzone für Achseneingaben

Die Achsen von Gamepads liefern permanent einen Wert, auch wenn die Achsen nicht betätigt werden. So kommen extrem kleine Werte bei einer permanenten Abfrage zu Stande. Die Achsen können nicht perfekt in ihrer mechanischen X- und Y-Richtung auf 0/0 gesetzt sein.

Ein Totzonen-Bereich, auch als Deadzone bezeichnet, muss daher bestimmen, ab wann der Wert einer Achse für weitere Berechnungen übergeben werden soll. Dabei wird zunächst ein Minimalwert definiert, der überschritten werden muss, um den ausgelesenen Wert weiterzuverwenden. Da der Wert der Achse positiv oder negativ sein kann, muss der Betrag ermittelt werden. Überschreitet der Wert des Betrages der Achseneingabe den Wert der Deadzone, können weitere Anweisungen erteilt werden.

Cursor bewegen

Der Mauscursor soll ebenfalls per Bewegung des linken Thumbsticks in bestimmten Situationen gesteuert werden. Dabei wird der aktuelle Wert des Mauscursors abgefragt, neu berechnet und wieder auf eine andere Position gesetzt.

Die neue Cursorposition berechnet sich aus der alten Position, dem Wert eines Thumbsticks und einem Faktor für die Geschwindigkeit, mit der sich der Mauscursor bewegen soll. Der Wert des Thumbsticks wird bei der Berechnung mit seinem eigenen Betrag multipliziert. So sorgen feine Bewegungen des Thumbsticks dafür, dass der Mauscursor eine niedrigere Bewegungsgeschwindigkeit hat und dem Nutzer somit eine feinere Steuerung ermöglicht wird. Die Funktion **GetCursorPos(&pCursorPosition)** wurde speziell zum Auslesen der Mauscursorposition erstellt. Diese übergibt die X- und Y-Werte der aktuellen Cursorposition kontinuierlich an die Variable, mit dem Typ Punkt **CursorPosition**. Nach der Berechnung eines neuen Wertes wird mit

SetCursorPos(x,y) die neue Position für den Mauscursor gesetzt.

6.2 Kamera-Steuerung

Kamera bewegen

Die Kamerabewegung auf der X- und Y-Achse wird ermöglicht, indem der Kamera mit dem vorgegeben Befehl **Move(FVector3(fThumbstickLX * fVelocity, fThumbstickLY * fVelocity, 0.f))** die X- und Y-Werte des linken Thumbsticks innerhalb eines Vektors übergeben werden. Der Z-Wert des Vektors ist 0. So wird die Kamera nur auf X- und Y-Achse bewegt und bleibt auf der Z-Ebene stetig auf der gleichen Position.

Kamera rotieren

Die Rotation der Kamera wird in eine horizontale und vertikale Rotation aufgetrennt. Der Befehl **RotateHorizontally(...)** ermöglicht die horizontale Drehung, der Befehl **RotateVertically(...)** ermöglicht die vertikale Drehung. Auch diese beiden Befehle sind schon vorgegeben und müssen nur in der passenden Situation angesteuert werden. Den Rotationsbefehlen wird ein Wert, der sich aus einem vorher definierten Geschwindigkeitsfaktor und dem Wert der Thumbstick-Achsen berechnet, übergeben. Bei der horizontalen Rotation wird der X-Wert des rechten Thumbsticks übergeben, bei der vertikalen Rotation, der Y-Wert. Zudem wird noch die aktuelle Position zur Höhe und Weite der Kamera eingerechnet.

Kamera-Zoom

Mit dem Befehl **Zoom(...)** kann der Kamera ein Wert übergeben werden, mit dem ein Heran- oder Herauszoomen möglich gemacht wird. Dabei wird ein Wert übergeben, der sich aus einem vordefinierten Geschwindigkeitsfaktor und den Trigger-Achsenwerten des Gamepads berechnet.

6.3 Manipulation von Objekten im 3D-Raum

Parts/Elemente selektieren

Objekte im 3-dimensionalen Raum des Spieles müssen ebenfalls selektiert werden können. Wenn ein Objekt selektiert ist, werden so genannte Manipulatoren aufgerufen. Manipulatoren beeinflussen die Eigenschaften von Objekten. So können Position, Rotation, Größe, sowie andere Werte beeinflusst werden. Bei der Selektion eines Parts oder eines Elementes, werden zunächst bestimmte Bedingungen abgefragt, die bestimmen, ob ein Manipulator überhaupt aufgerufen werden darf. Sind alle Bedingungen erfüllt, kann der Manipulator aufgerufen werden und damit ist das Objekt als ausgewählt gekennzeichnet. Die Auswahl von Objekten erfolgt mit **GamepadObjectFunctions()**.

Parts/Elemente deselektieren

Das Deselektieren von Objekten wird benötigt, um Elemente oder Parts abzusetzen und deren über die Manipulatoren veränderten Eigenschaften zu akzeptieren. Hierfür werden die zuvor aufgerufenen Manipulatoren beendet. Die Funktion **EscapetemStates()** ermöglicht das Beenden von Manipulatoren.

Parts/Elemente rotieren

Das Rotieren von Objekten erfolgt mit dem Neigen des rechten Thumbsticks auf seiner X-Achse. Zunächst müssen bestimmte Bedingungen erfüllt werden. Ein Manipulator muss aktiv sein, keine UI-Region sollte ausgewählt sein und das Spiel darf sich nicht im Spielmodus (Solve State) befinden. Sind diese Bedingungen erfüllt, wird der X-Wert der Thumbstick-Achse ausgelesen. Anschließend wird abgefragt, ob die Deadzone des Thumbsticks überwunden wurde und der aktive Manipulator überhaupt die Möglichkeit zur Rotation bietet. Ist die Möglichkeit zur Rotation gegeben, ruft der Manipulator den Befehl **StepRotate(...)** auf. Dabei wird ein Wert übergeben, der sich aus dem aktuellen Wert der Thumbstick-Achse und einem Faktor für die Skalierung des Rotationsschrittes berechnet.

Parts/Elemente skalieren

Objekte können auch in ihrer Größe verändert werden. Dies funktioniert ähnlich, wie das Rotieren. Durch die Bewegung des rechten Thumbsticks auf seiner Y-Achse wird die Größe von einem ausgewähltem Objekt verändert. Dabei können eine Mindestgröße oder eine Maximalgröße erreicht werden, die das weitere Skalieren in die jeweilige Größenrichtung verhindert. Durch den Befehl **StepScale(...)** wird dies im Code ermöglicht.

Value-Manipulator

Vom Value-Manipulator wird immer dann gesprochen, wenn ein Part eine skalierbare Variable beinhaltet, die der Nutzer einstellen kann. Dabei kann es sich bei den einstellbaren Werten um Zahlen für mathematische Operationen, Kanälen von Radio-Systemen und interaktiven Chips, sowie die Anzahl, wie oft Klänge an Sound-Chips abgespielt werden sollen, handeln.

Element-Offset-Verschiebung

Im Part-Editor besteht die Möglichkeit, Elemente auf bestimmte Arten zu verschieben. Dabei wird das Objekt auf der Normalen, am Snap-Punkt verschoben. Hierfür wird ein spezieller Manipulator verwendet. Dieser wird mit dem D-Pad aktiviert. Anschließend kann mit dem rechten Thumbstick die Verschiebung eingestellt werden. Dabei wird die Y-Bewegung des Sticks ausgewertet.

Element-Offset-Rotation

Wenn der Offset-Manipulator aktiv ist, kann auch um die Normale des Snap-Punktes rotiert werden. Dies geschieht ebenfalls mit der Betätigung des rechten Thumbsticks. Allerdings wird hierbei die X-Bewegung genutzt, um die Rotation zu ermöglichen.

Brushes

Brushes (Engl.: Pinsel) sind in „Crazy Machines 3“ Zustände, die aktiviert werden können, welche dem Nutzer das Anwenden von Aktionen ermöglicht.

Ähnlich, wie in einem Zeichenprogramm können so per Klick Veränderungen an Objekten vorgenommen werden. So können beispielsweise Bauteile gelöscht, umgefärbt oder als nutzbar markiert werden.

Arten von Brushes

Name	Beschreibung
Remove	ermöglicht das Löschen von Bauteilen oder Elementen
Static	fixiert Bauteile oder Elemente am Ort
Freeze	gibt einem Bauteil nur die Möglichkeit zur Bewegung in der 2D-Ebene
Usable	markiert ein Bauteil als benutzbar im Solve-Mode
Material	verändert das Material eines Elementes
Color	verändert die Farbe des Materials eines Elementes

Tabelle 2: Brushes.

Brushes Anwenden

Wenn ein Brush aktiv ist, kann er durch Betätigen des unteren Face-Buttons auf das Objekt, auf dem sich der Cursor befindet, angewendet werden.

6.4 User Interface

6.4.1 Auswertung und Steuerung

Konzept der UI-Zustände

Da es bei einer Gamepadeingabe nicht viel Sinn macht, wie zum Beispiel bei einer Maussteuerung, das Klicken auf UI-Objekte abzufragen, musste ein anderes System zur Verwaltung der aktuellen UI-Objekte und deren Funktionen erstellt werden. Hierfür wurden mehrere Variablen angelegt, die sich bestimmte Positionen oder aktive Elemente merken.

Durch gezieltes Setzen und Abfragen der Variablen kann ein spezieller Zustand im Spiel für die UI aktiviert oder ausgelesen werden. So weiß das Programm, ob es sich beispielsweise im Menü befindet und in welcher Liste, mit der aktuellen Listenposition, gerade eine Navigation möglich ist. Dem Nutzer werden durch das Auswerten der nachfolgenden Variablen in ihren jeweiligen Konstellationen, die verschiedenen Funktionen bei der Betätigung der Gamepad-Tasten und Achsen, sowie die sichtbaren UI-Zustände, inklusive Beschreibung für die aktuellen Eingabemöglichkeiten, zur Verfügung gestellt.

Variable	Beschreibung
nActualUIRegion	sagt aus, welche UI-Region gerade aktiv ist
nActualList	sagt aus, welche Liste gerade aktiv ist
nActualListPosition	der Wert, der aktuellen Position in einer Liste, wird beispielsweise als Index eines Objektes, das aus einer Liste heraus erzeugt werden soll, übergeben
nActualCategoryTab	der Wert eines bestimmten Tabcontrols in den Katalogen
nCircleSelectionPosition	aktive Auswahl einer Funktionalität in einem Ringmenü

Tabelle 3: Variablen für UI-Zustände.

Anhand dieser Variablen kann ermittelt werden, wo sich der Nutzer gerade in der UI befindet und welche Möglichkeiten ihm dort angeboten werden.

Beispiel:

nActualUIRegion = *Menu_Levels*
nActualList = *Menu_LevelList*
nActualListPosition = *0*

Der Nutzer befindet sich im Menü, Bereich Level-Auswahl. Dabei navigiert er in der Level-Liste und hat gerade das 1. Level in der Liste ausgewählt. In genau diesem Zustand besteht nun die Möglichkeiten, das Level zu laden, im Menü in ein anderes Gebiet (Beispielsweise Startbildschirm oder Optionen) zu wechseln, in der Level-Liste zu navigieren oder das Menü zu schließen. Andere Funktionen, wie ein Bauteil im Spiel verschieben oder in anderen Listen navigieren, können in diesem UI-Zustand nicht angesteuert werden.

UI-Regionen auswerten

Die Variablen der UI sind jederzeit abrufbar. So können durch If-Anweisungen oder Switch-Cases Bedingungen gestellt werden, deren Erfüllung weitere Möglichkeiten zur Ausführung von Programmteilen zur Verfügung stellen. Um den Code schlank zu halten, wurden Funktionen entwickelt, die in sich schon Bedingungsabfragen stellen können. Dadurch wird vermieden, dass immer wiederkehrende Bedingungsabfragen erneut programmiert werden müssen. Die Funktion **GamepadListSelectionByTab()** fragt ab, in welcher UI-Region sich der Nutzer gerade befindet und setzt dann die zugehörigen Listen aktiv. Diese Funktion wird aufgerufen, wenn der Nutzer sich zum Beispiel im Menü oder im Katalog befindet und die Tabs mit dem Gamepad durchschaltet. Die Funktion **CategoryTabNavigation(INT nStep)** wird im Katalog aufgerufen, wenn der Nutzer den rechten oder linken Trigger betätigt. Dabei wird die Variable **nActualCategoryTab**, in den mit **nStep** übergebenen Wert, verändert und eine Funktion zum Filtern der Kataloginhalte aufgerufen.

UI-Elemente ansteuern

Die bereits vorgegebene Funktion **SimulateClick(...)** behandelt UI-Objekte so, als würden diese mit Hilfe einer Maussteuerung angeklickt werden. So können beispielsweise Toggle-Buttons angeklickt werden. Ansonsten müsste jedes Mal der Zustand des Toggle-Button abgefragt, gesetzt und zu dem noch die dazugehörige Funktion, die an den Button gebunden ist, ermittelt und extra aufgerufen werden.

Storyboards ansteuern

Die Funktion **PlayStoryboard(...)** spielt bestimmte Storyboards in der UI ab. Storyboards sind im XAML definierte Animationen oder Zustandsänderungen von UI-Objekten. So können beispielsweise Panels animiert, oder Objekte, wie Buttons und Images ein- beziehungsweise ausgeblendet werden.

Anzeige der möglichen Aktionen

Die Eingabemöglichkeiten per Gamepad sollten immer möglichst intuitiv gestaltet sein. In den meisten Spielen sind Bewegungssteuerungen für Charaktere oder Navigation in einem Menü oder Inventar auf dem linken Thumbstick und/oder D-Pad erreichbar. Der untere Face-Button (A-Button auf einem X-Box360-Controller) ist oftmals für häufig genutzte Aktionen und der Bestätigung einer Auswahl vorgesehen. Für das Abbrechen von Aktionen wird dann auch meistens der rechte Face-Button (B-Button auf einem X-Box360-Controller) genutzt.

Natürlich haben alle Spiele und Systeme unterschiedliche Funktionen. Damit der Nutzer weiß, welche Eingabemöglichkeiten in bestimmten Situationen zur Verfügung stehen, wird oftmals das Symbol des jeweiligen Buttons oder Thumbsticks, mit einem kurzen Text zur Erklärung eingeblendet. Dies könnte man mit dem Einblenden von Tooltips bei Maus-Tastatur-Steuerungen vergleichen, wenn sich der Mauscursor über einem UI-Element befindet.

Darstellung von Informationen

Bei der Darstellung, der dem Nutzer zur Verfügung stehenden Aktionen, wurde sich für das Einblenden von Texten am unteren Bildschirmrand entschieden. Diese Methode wird oft genutzt, um die Informationen an einer nicht störenden und leicht erkennbaren Position für den Nutzer erkennbar zu machen. Ein Spiel, welches die Eingabemöglichkeiten auf ähnliche Art und Weise darstellt, ist „Little Big Planet“ von Sony Computer Entertainment.

Umsetzung

Um die, für den Nutzer erreichbaren Möglichkeiten überhaupt darstellen zu können, müssen UI-Objekte erstellt und angesteuert werden. Zudem müssen die jeweiligen Fälle, Wann, Was und Wo angezeigt wird, unterschieden werden.

Bei dem Ansteuern der UI-Objekte werden bestimmte Elemente ein- oder ausgeblendet und Text zur Beschreibung der Aktionen dargestellt.

Die Fallunterscheidung, wann welche Anzeige dargestellt werden soll, wird durch eine Funktion ermöglicht, die abfragt, welche Zustände gerade im Spiel aktiv sind.

In dieser Funktion ist jeder Zustand definiert, in dem eine Anzeige erfolgen soll.

6.4.2 Listen

Listen in Abhängigkeit der UI-Regionen auswerten

Mit Hilfe der **Funktion GamepadListSelectionByTab()** wird die aktuelle, als aktiv festgelegte UI-Region abgefragt und in Abhängigkeit davon die jeweils dazugehörige Liste ermittelt. Nach der Auswertung der Variable für die aktuelle UI-Region, wird eine andere Variable, welche die aktuell zu bedienende Liste repräsentiert, gesetzt.

Listen-Navigation

Die Funktion **GamepadNavigateList(INT nStep)** ermöglicht das Navigieren in Listen. Dabei wird zunächst ermittelt, welche UI-Region gerade aktiv ist und welche Liste dementsprechend angesteuert ist. Zu dem wird ein Maximalwert ermittelt, den die Variable für die aktuelle Listenposition erreichen kann. Dazu werden in Abhängigkeit von der aktuellen Liste, Befehle aufgerufen, die beispielsweise die Anzahl der verfügbaren Environments, Farben oder Elemente, ermitteln. Der ermittelte Wert wird zusätzlich noch um 1 subtrahiert und in eine neue Variable übergeben. Diese zusätzlich Operation ist notwendig, da der niedrigste Index für die aktuelle Listenposition 0 ist. Der Funktion zum Navigieren in Listen wird der Wert **nStep** übergeben. Dieser bestimmt die Schrittgröße, in der die Listenposition weiter geschaltet wird. Dieser Wert wird mit der aktuellen Listenposition verrechnet. Übersteigt der Wert dabei das mögliche Maximum, wird der Wert der aktuellen Listenposition auf das Maximum gesetzt. Das Ende der Liste ist also erreicht. Fällt der Wert unter 0, wird er auf das Minimum von 0 gesetzt.

Die Navigation in den jeweiligen Listen erfolgt über das Betätigen des D-Pads. In Abhängigkeit von der aktuellen UI-Region wird entschieden, welcher Wert der Funktion zum Navigieren übergeben wird. So wird beispielsweise bei einer einreihigen Liste, wie dem Properties-Panel, bei dem Betätigen des unteren D-Pad-Buttons, der Wert 1 übergeben. Bei einer zweireihigen Liste, wie der Element-Liste, wird der Wert 2 übergeben, da das rechts liegende Listenelement bei einer Navigation nach Unten einfach übersprungen wird.

Objekte in Listen benutzen

Wenn eine Liste gerade aktiv ist, können Objekte, wie Bauteile oder Elemente, aber auch Maschinen und Brushes, in der Liste ausgewählt und durch Betätigen des unteren Face-Buttons benutzt werden. Dabei wird der Funktion

GamepadUseObjectFromList(...) die aktuelle Listenposition übergeben. Diese Funktion entscheidet dann, je nachdem welche Liste gerade aktiv ist, ob ein Objekt erzeugt, ein Brush aktiviert oder eine Maschine geladen werden soll.

Objekte aus Listen löschen

Ähnlich, wie das Erzeugen von Objekten, funktioniert auch die Löschfunktion. Die Funktion **GamepadDeleteObjectFromList(...)** verarbeitet einen Zahlenwert, der übergeben wird und entscheidet dann, welche Löschfunktion aufgerufen wird. Dabei können jedoch nur bestimmte Bauteile und Maschinen gelöscht, jedoch standardmäßig im Spiel enthaltene Bauteile und Maschinen können so nicht entfernt werden. Nur, Objekte, die der Nutzer selber erstellt oder aus dem Steam-Workshop heruntergeladen hat, können gelöscht werden. Auch diese Fallunterscheidung übernimmt die Funktion, indem sie abfragt, ob das Objekt als „vom Nutzer erstellt“ markiert wurde.

Filtermöglichkeiten in Part/Element-Listen

Im Machine- und Part-Editor sind eine Vielzahl von Objekten gegeben. Diese können nach bestimmten Kriterien gefiltert werden, sodass der Nutzer bestimmen kann, ob beispielsweise nur Logik-Bauteile, Effekte oder simple Objekte angezeigt werden sollen. Durch das Benutzen der Schultertasten während das Katalog-Panel geöffnet ist, wird dies dem Nutzer ermöglicht. Am oberen Rand des Panels wird die aktuelle Filterung angezeigt.

6.4.3 Ringmenü

Definition

Das Ringmenü oder Circleselection, ist eine Auswahlmöglichkeit. Dies ist eine Alternative zu klassischen, kachelförmig dargestellten Auswahlmöglichkeiten und wird häufig für die Steuerung von Inventar, Dialog- oder Schnellzugriffssystemen in Spielen verwendet.

Ein Beispiel für die Verwendung eines solchen Systems ist das Schnellzugriffssystem aus „The Elder Scrolls Online“, in dem der Nutzer sich eine Favoritenauswahl an Verbrauchsgütern, wie Tränken oder Nahrung anlegen kann. Das MMORPG von Zenimax Media Inc. ist sowohl für eine klassische Maus- und Tastatursteuerung für PC, als auch für eine Gamepad-Steuerung auf Konsolen ausgelegt. Das Ringmenü ermöglicht in diesem Beispiel einen schnellen Zugriff auf Objekte im Inventar und kann aufgerufen werden, indem eine bestimmte Taste für eine gewisse Zeit gehalten wird. Die Auswahl erfolgt dann mit dem Bewegen des Gamepad-Thumbsticks oder per Mausbewegung.



Abbildung 15: Ringmenü aus „The Elder Scrolls Online“.

In der ursprünglichen Maus- und Tastaturvariante von „Crazy Machines 3“ werden viele Funktionen im Spiel per Klick auf Buttons ausgeführt, zum Beispiel das Öffnen von Panels für den Katalog, für die Materialien, für die Farben oder für die Einstellungen. Brushes mit Funktionen die Objekte beispielsweise befestigen oder löschen können, werden ebenfalls per Button-Klick aktiviert. Da die Anzahl der Eingabetasten auf einem Gamepad stark begrenzt ist, kann nicht jede Funktion direkt auf eine eigene Eingabetaste gelegt werden, also wurden alle ähnlichen Funktionen zusammengeführt und für die Gamepad-Variante von „Crazy Machines 3“ in einem Ringmenü zugänglich gemacht. So muss der Nutzer nur eine Eingabetaste bedienen und eine einfache Bewegung mit einem Thumbstick zur Auswahl tätigen. Dies ist für eine überschaubare Anzahl von Auswahlmöglichkeiten eine effizientere Methode als lineares Durchschalten oder die Bedienung einer kachelförmig angeordneten Auswahl. In „Crazy Machines 3“ wird das Ringmenü genutzt, um Panels zu öffnen oder bestimmte Werkzeuge zu aktivieren. So müssen ähnliche Funktionen, die einen schnellen Zugriff benötigen keine eigenen Tasten zugewiesen werden.

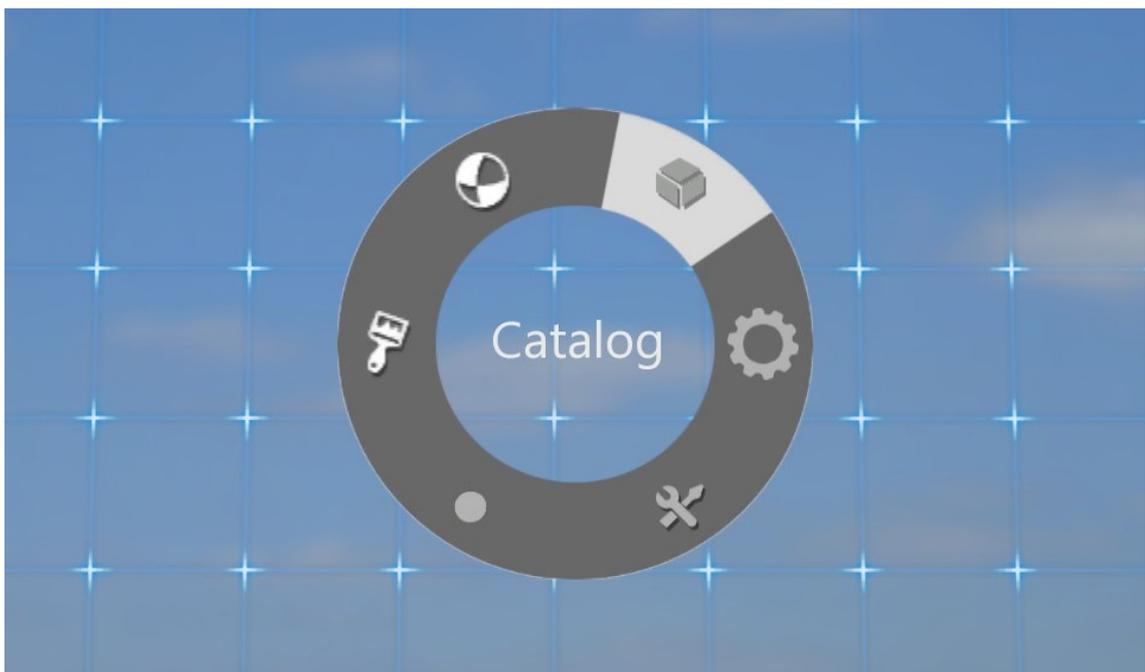


Abbildung 16: Ringmenü in „Crazy Machines 3“.

Funktionen

Das Ringmenü ermöglicht im Maschinen-Editor 6 grundlegende Funktionen:

Bauteilkatalog öffnen

Der Bauteilkatalog gibt dem Nutzer die Möglichkeit, aus einem Sortiment von im Spiel bereits enthaltenen, aus dem Steam Workshop heruntergeladenen und selbst erstellten Bauteilen, eine Auswahl zu treffen.

Bei der Selektion dieses Punktes im Ringmenü, wird links im Bildschirm ein Panel geöffnet. Darin befindet sich die Liste der Bauteile, in der der Nutzer navigieren und seine Auswahl treffen kann. Die Bauteile können nach verschiedenen Kategorien sortiert werden. So kann der Nutzer gezielt Objekte für seinen aktuellen Bedarf finden. Außerdem können während der Navigation in der Liste auch selbsterstellte Bauteile bearbeitet und gelöscht werden. Aus dem Steam Workshop heruntergeladene Teile, können zwar nicht bearbeitet, aber de-abonniert werden und es besteht die Möglichkeit direkt im Steam Overlay auf die Workshop-Seite des Objektes zu gelangen. Standardmäßig im Spiel enthaltene Teile, bieten keine weiteren Optionen an.



Abbildung 17: Bauteilkatalog und einige Bauteile.

Materialienauswahl öffnen

Dem Nutzer ist die Möglichkeit zur Änderung von unterschiedlichen Materialien für seine Bauteile gegeben. So kann er beispielsweise eine Holz-Kiste in eine Metall-Kiste verwandeln oder ein gewöhnliches Automobil aus Metall, mit Gummireifen zu einem Fahrzeug aus Stein, mit Styroporrädern werden lassen.

Die Materialienauswahl öffnet sich ähnlich wie der Bauteilkatalog in einem extra Panel, links im Bildschirm. Die Anordnung der Objekte und Navigation innerhalb der Liste funktioniert ebenfalls wie im Katalog. Hier kann der Nutzer aus verschiedenen Materialien eine Auswahl treffen.

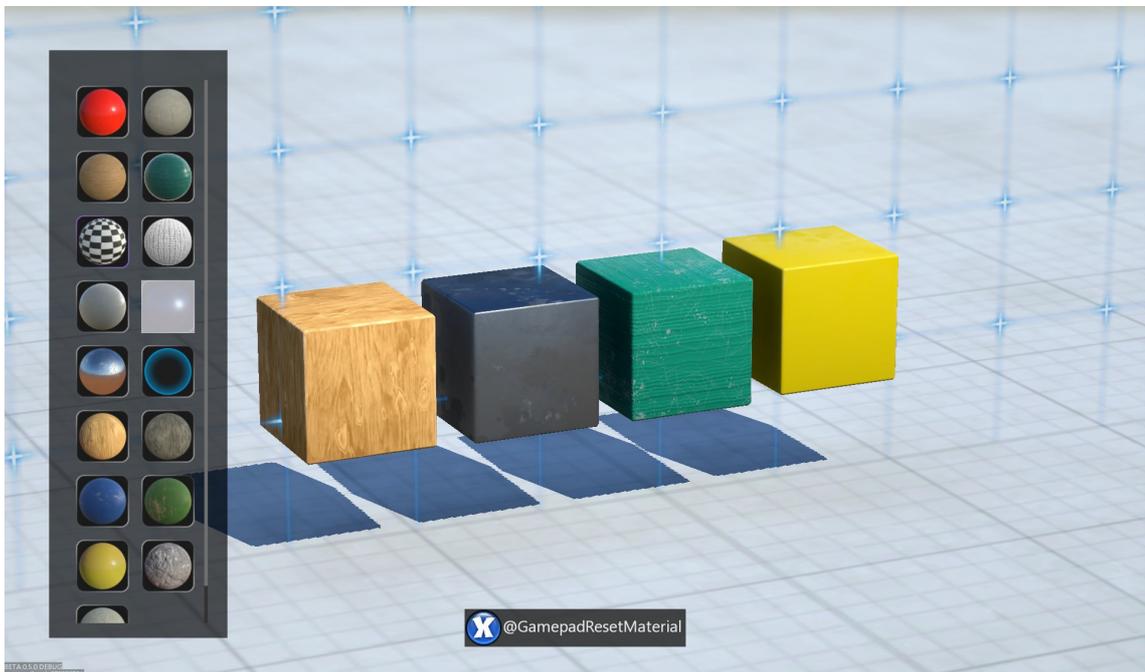


Abbildung 18: Materialienauswahl und Würfel mit unterschiedlichen Materialien.

Farbauswahl öffnen

Ebenfalls die Auswahl von verschiedenen Farben, die auf Objekte angewendet werden können, ist gegeben. Dank dieser Möglichkeit hat der Nutzer noch mehr Freiheiten in der Gestaltung von Bauteilen. So kann beispielsweise ein grüner Kaktus, Pink eingefärbt werden oder die Styroporräder des steinernen Fahrzeuges erstrahlen in unterschiedlichen Farbtönen. Die Navigation und Auswahl in diesem Panel funktioniert exakt, wie in der Materialienauswahl.

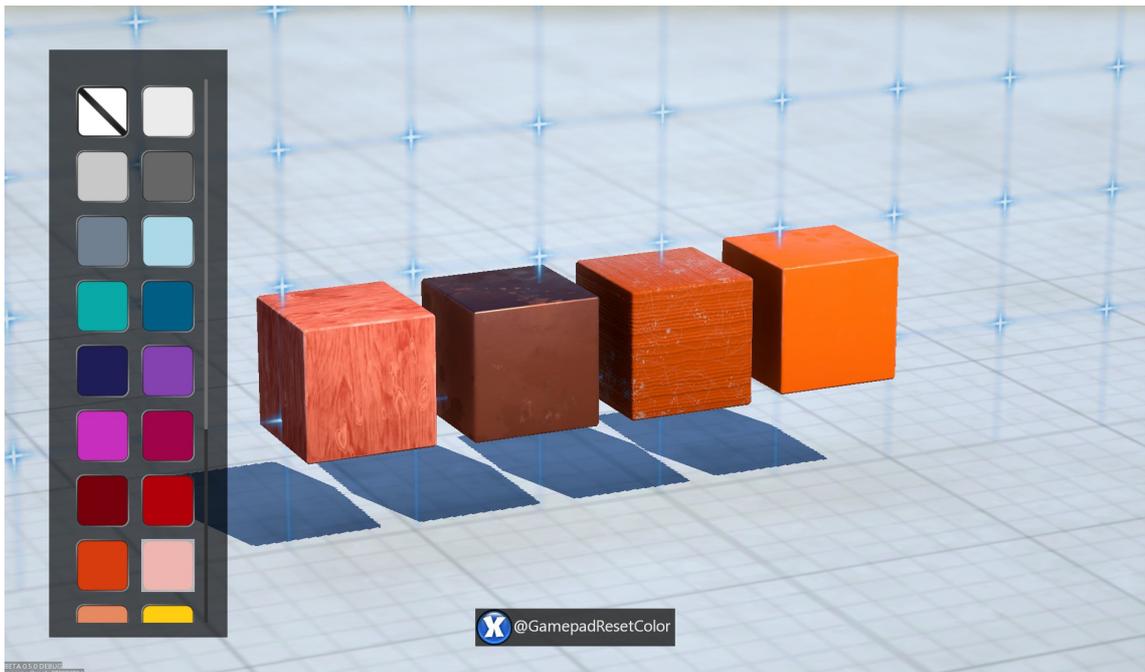


Abbildung 19: Farbauswahl und eingefärbte Würfel.

Erstellung eines neuen Bauteils

Die Erstellung eines eigenen Bauteiles, ermöglicht dem Nutzer seiner Kreativität vollkommen freien Lauf zu lassen. So können nicht nur Maschinen erstellt werden, sondern auch deren Bestandteile. Das Spiel bietet bereits schon viele Bauteile, wie Motoren, Ventilatoren und Laserkanonen. Selbst verrückte Objekte, wie mechanische Boxhandschuhe oder Pflanzen die bei Lichteinwirkung wachsen können, sind vorhanden. Doch der Nutzer kann auch eigene Bauteile erstellen. Dabei können verschiedene Objekte aneinandergesetzt und mit einer eigenen Logik versehen werden. So könnte beispielsweise ein ferngesteuertes Pulverfass auf Rädern das nach einer gewissen Zeit explodiert, erstellt werden. Im Part-Editor, der bis auf gewisse Punkte ähnlich, wie der Maschinen-Editor funktioniert, ist diese Möglichkeit gegeben. Er kann über das Ringmenü direkt erreicht werden.

Werkzeug-Auswahl öffnen

Mit Hilfe der Werkzeugauswahl kann der Nutzer Objekte als statisch, nutzbar, oder auf einer Ebene verankert markieren. Zu dem können so auch Objekte gelöscht oder Verbindungen zwischen Logikteilen geschaffen werden. Diese Werkzeuge lassen sich ähnlich bedienen und sind deshalb in dieser Sektion zusammengefasst. Per Betätigung

6.4.4 Editoren

Panels öffnen/schließen

Im Maschinen- oder Bauteileditor müssen die rechten und linken Panels für Objektkataloge und Eigenschaften ein und ausgeblendet werden. Hierfür werden bestimmte Storyboards aufgerufen und die jeweiligen Variablen für die UI-Zustände gesetzt. Per Ringmenü und Bewegen des linken Thumbsticks in eine bestimmte Richtung, werden die Panels geöffnet. Bestimmte Aktionen wie das Anwenden von Objekten, Materialien und Farben schließen die Panels wieder und die weitere Bedienung des Editors wird wieder ermöglicht. Zudem können die Panels auch direkt per Tastendruck geschlossen werden, wenn der Nutzer die Aktion abbrechen möchte.

Erweiterte Panel-Navigation

Spezielle Bereiche der Panels können besondere Funktionen ermöglichen, welche über das einfache Bedienen von Listen hinaus geht. So müssen beispielsweise Textboxen oder spezielle Buttons, in den jeweils speziellen Situationen, angesteuert werden. In den Properties-Panels, muss eine Tag-Vergabe für Maschinen und Bauteile, die Möglichkeit in den Screenshot-Mode zu gelangen und das Teilen von Objekten im Steam-Workshop gegeben werden. Das Katalog-Panel hat ebenfalls spezielle Funktionen, wie neue Bauteile erstellen oder den Steam-Workshop aufrufen. Auch hierfür wird zunächst die aktuelle UI-Region abgefragt. So kann entschieden werden, welche Funktionen erreichbar sind. In den Properties-Panels wird zudem die aktuelle Listenposition abgefragt. In Abhängigkeit der aktuellen Listenposition wird unterschieden, ob eine Namens- oder Beschreibungsvergabe, eine Tag-Vergabe, das Teilen von Maschinen oder Bauteilen oder das Katalogisieren von Bauteilen erfolgen soll. So werden beispielsweise im Maschineneditor, wenn das Properties-Panel geöffnet und der Tab für die Informationen aktiv ist, bei einer aktuellen Listenposition zwischen 2 und 11, das Vergeben von Tags für die Maschine ermöglicht. Eine aktuelle Listenposition von 12 hingegen würde den Screenshot-Mode aufrufen.

Sky-Rotation

Die Rotation der Himmelskugel erfolgt per Betätigung der Trigger-Tasten. Dabei wird zunächst überprüft, ob eine Himmelsrotation überhaupt getätigt werden kann. Diese Aktion soll nur nutzbar sein, wenn der Nutzer auch Zugriff auf die Himmelsauswahl hat. Die Rotation des Himmels kann in die positive oder negative Richtung erfolgen, je nachdem, welcher Trigger am Gamepad aktiviert wird. Zudem ist die Rotationsgeschwindigkeit ebenfalls abhängig von der Stärke der Trigger. Um die Rotation zu ermöglichen, wird mit der Funktion **GetAxis(...)** der Wert der Trigger ausgelesen. Obwohl die Trigger-Tasten nicht so empfindlich sind wie die Thumbsticks, muss zur Sicherheit auch hier eine Deadzone überwunden werden. Kann diese Deadzone nicht überwunden werden, wird die Rotationsgeschwindigkeit des Himmels (**m_fSkyAngleChangeVelocity**) auf 0 gesetzt, andernfalls wird eine Differenz aus den Stärken der Trigger berechnet und übergeben.

Umschalten von Tabs

Das Umschalten von Tabs wird per Nutzung der linken und rechten Schultertaste ermöglicht, wenn Panels oder das Menü geöffnet sind.

Die Funktion **TabNavigation(INT nStep)** fragt die aktuelle UI-Region ab und verarbeitet diese Information, um die Tabs durchzuschalten. Dazu wird zunächst ermittelt, ob das aktive UI-Gebiet der Tabs im Menü oder in den Panels ist. So wird beispielsweise, wenn die Variable **nActualUIRegion** zwischen den Werten 1 und 7 liegt, das Umschalten von den Menü-Tabs ermöglicht. Der Wert 1 entspricht dem Home-Tab im Menü, der Wert 7 entspricht dem Options-Tab. Zwischen diesen äußeren Tabs liegen noch weitere Tabs, wie beispielsweise die Tabs für Level-Auswahl oder Workshop-Levels.

Der Funktion zum Tab umschalten wird eine Variable **nStep** übergeben. Diese Variable entscheidet, wie weit und in welche Richtung Tabs umgeschaltet werden. Der Wert -1 schaltet eine UI-Region zurück, der Wert +1 schaltet eine UI-Region vor. Da zunächst der Bereich bestimmt wurde, in dem sich die Tabs zum umschalten befinden (Menü oder Panels), wird beim Unterschreiten eines Minimalwertes oder beim Überschreiten eines Maximalwertes, die aktuelle UI-Region auf das jeweils andere Ende des eingegrenzten Bereiches gesetzt.

Menü öffnen/schließen

Das Menü wird ebenfalls per Storyboard ein- und ausgeblendet und setzt die Variablen zum Navigieren zwischen den Tabs und den verschiedenen Listen in den Menübereichen. Das Menü wird wie in den meisten Spielen (beispielsweise Little Big Planet) per Betätigung des Startknopfes geöffnet oder geschlossen. Besondere Fälle, in denen man das Menü nicht öffnen oder schließen kann, sind ebenfalls gegeben.

Simulation Starten/Stoppen

Durch Betätigung des oberen Face-Buttons wird die Simulation gestartet und wieder gestoppt. So kann der Nutzer im Solve-Mode sehen, ob er das Rätsel gelöst hat, wenn die Kettenreaktion einen Erfolg ausgibt. Im Machine-Editor hingegen ist das Abspielen der Simulation nützlich, um die eigene Maschine zu Testen.

Aktive Kamera umschalten

Während die Simulation läuft, ist es möglich zwischen verschiedenen Kameras hin- und herzuschalten. So muss die Standardkamera, eine automatische Kamera und bis zu 4, durch den Inhalt von Bauteilen definierte Kameras angesteuert werden. Das Umschalten der Kameras erfolgt durch die Betätigung der Schultertasten. Dabei wird eine Variable um den Wert 1 erhöht oder verringert. In Abhängigkeit des Wertes dieser Variable wird dann entschieden, welche Kamera aktiv zu schalten ist. Bei einem Wert von 0, wird die Standardkamera aktiviert, der Wert von 5 aktiviert eine automatische Kamerafahrt, die der Nutzer mit dem Ansteuern von bestimmten Bauteilen festlegen kann. Die Werte dazwischen steuern bis zu 4 vom Nutzer festlegbaren Kamerakanälen an. Die Kamerakanäle werden in speziellen Elementen per Value-Manipulator festgelegt. Falls bei dem Ansteuern der Kamerazustände einer dieser Kanäle nicht belegt ist, wird der jeweils Nächste angesteuert. So ist, wenn kein Kamerakanal definiert ist, nur das Hin- und Herschalten zwischen automatischer und Standardkamera möglich. Die Funktion **GamepadChangeCamera(INT nStep)** ermöglicht dies. Die Variable **nStep** gibt die Schrittgröße der Umschaltungen zwischen den Kameras vor. Diese ist mit -1 und +1 definiert und wird über die linke oder die rechte Schultertaste des Gamepads gesteuert.

Simulationsgeschwindigkeit ändern

Die Geschwindigkeit der Simulation kann insgesamt 3 Zustände erreichen. Die normale Geschwindigkeit, eine Zeitlupengeschwindigkeit und den kompletten Stillstand. Die Zustände der Simulationsgeschwindigkeit werden per Betätigung des linken Face-Buttons (Bei einem X-Box-Controller die X-Taste) durchgeschaltet.

Part Editor betreten

Der Part-Editor kann auf 3 Arten betreten werden.

1. Ein neues Bauteil erstellen

Über das Ringmenü kann ein neues Bauteil angelegt werden. Dabei wird einfach der Part-Editor ohne Vorgaben geladen und ist so zu sagen leer.

2. Ein Bauteil aus dem Katalog heraus bearbeiten

Vom Nutzer erstellte Bauteile können direkt im Katalog abgeändert und überschrieben werden, indem der linke Face-Button betätigt wird, während der Katalog geöffnet ist. Dabei wird der Part-Editor aufgerufen und das zu bearbeitende Bauteil geladen. Nach der Bearbeitung wird das Bauteil im Katalog überschrieben.

3. Ein in einer Maschine eingesetztes Bauteil bearbeiten

Wenn ein Bauteil in einer Maschine ausgewählt ist, kann es durch Betätigung des linken Face-Buttons, ähnlich wie bei der Bearbeitung aus dem Katalog, editiert werden. Jedoch wird dieses Bauteil nach dem Verlassen des Part-Editors nur in der Maschine verändert, nicht im Katalog.

Part Editor verlassen

Durch das Betätigen des oberen Face-Buttons kann der Part-Editor verlassen werden und der Nutzer gelangt in den Maschineneditor zurück. Falls kein neues Bauteil angelegt und ein bestehendes Bauteil verändert wurde, erfolgt beim Verlassen noch die Sicherheitsabfrage, ob die Änderungen übernommen werden sollen.

Parts katalogisieren

Bauteile können katalogisiert werden. Das bedeutet, dass im Katalog eine Kopie des gerade im Editor befindlichen Bauteiles abgespeichert wird. Die Möglichkeit zum Katalogisieren ist über das Eigenschaften-Panel erreichbar und wird durch das Betätigen des unteren Face-Buttons ausgeführt.

6.5 Interaktive Eingabemöglichkeiten während der Simulation

In „CrazyMachines 3“ ist die Möglichkeit gegeben, Bauteile auch für den Nutzer, während die Simulation läuft, bedienbar zu machen. Der untere und der rechte Face-Button, sowie das D-Pad können zur Bedienung genutzt werden. Gerade diese beiden Eingabetasten und das Steuerkreuz eignen sich gut für beispielsweise Fahrzeuge oder bewegliche Figuren. Diese Tastenbelegung ist intuitiv bedienbar und wird von vielen anderen Spielen bereits erfolgreich verwendet.

7 Gesamtauswertung

Die Portierung der Maus-Tastatur-Steuerung von „Crazy Machines 3“ auf eine Gamepad-Steuerung verlief erfolgreich.

Die Aufwandsanalyse und der Prototyp wurden von FAKT Software geprüft und bestätigt, so dass die Umsetzung mit C++ erfolgen konnte.

Während der Arbeit an der Gamepad-Steuerung konnten viele Erkenntnisse im Bereich C++, Design und Entwicklung von Gamepad-Steuerungen, sowie Entwicklung von Unterhaltungssoftware gewonnen werden.

Zudem konnte FAKT während der Umsetzung der Steuerung, auch Experimente im Bereich Virtual Reality starten, da sich nur eine Gamepad-Steuerung für die Bedienung des Spieles mit einer VR-Brille als geeignet erwies. Die Gamepad-Variante bietet sich dabei gut an, da man so das Spiel steuern kann, ohne auf das Eingabegerät zu achten. Der Publisher Daedalic Entertainment lies die Steuerung testen.

Das Ergebnis der Testreihe fiel sehr positiv aus, da alle Funktionen als intuitiv bedienbar und leicht erlernbar eingestuft wurden. Da sich das Spiel derzeitig noch in Entwicklung befindet, können sich bis zur Veröffentlichung noch Funktionen, Grafiken oder Tastenbelegungen ändern.

Fremdwortverzeichnis

Bauteil/Part	Teil einer Maschine in "Crazy Machines 3"
Content	Inhalte eines Spieles
Drag-and-Drop Bedienung	Bedienungsmethode, bei der Objekte einfach nur per Verschiebung platziert oder benutzt werden
Druck-sensible Eingabemöglichkeiten	eine berührungsempfindliche Eingabemöglichkeit zur Bedienung
Element	Bestandteil eines Bauteiles in „Crazy Machines 3“
Environments	optische Umgebung einer Maschine in „Crazy machines 3“
HTC Vive	Virtual Reality-System, bestehend aus Brille, Sensoren und Stab-förmigen Eingabegeräten
If-Anweisungen	Wenn-Dann-Abfrage in einem Programm
Keycodes	Tasten zugewiesene Werte zur jeweiligen Identifikation
Maschine/Machine	Ein spielbares Level in „Crazy Machines 3“
Nintendo Wii-Steuerung	Eingabesystem, welches Bewegungen in ein Spiel oder andere Software übertragen kann
Normale	senkrecht zu einer Fläche verlaufende Gerade
Panels	grafisch dargestellte Fläche/Gebiet/Tafel, ähnlich wie ein Fenster
Properties-Panel	Gebiet, in dem Eigenschaften einer Maschine oder eines Bauteiles dargestellt werden
Publisher	Vertriebspartner
Radio-System	Ein System in „Crazy Machines 3“, welches wie Funksignalübertragung funktioniert
Rendering	Darstellung von Grafiken und 3D-Modellen
Rube-Goldberg-Rätsel	physikalisches Kettenreaktions-Rätsel

Snap-Punkt	Verbindungspunkt zwischen 2 Objekten
Steam Big Picture Mode	Gamepad-orientierte Darstellung von Steam
Steam Overlay	Feature des Steam-Clients
Steam Workshop	Plattform zum Erstellen und Teilen von Spiele-Inhalten
Switch-Cases	Fallunterscheidung in einem Programm
Systeme	in „Crazy Machines 3“ sind Systeme besondere Funktionen, wie Laser/Blitz/Wind/Radio-Simulationen
Tabcontrols	Steuerelement, welches Tabs enthält
Tabs	Reiter-Elemente in einer grafischen Benutzerschnittstelle
Toggle-Buttons	Knopf, der die Zustände Aktiv und Deaktiv erreichen kann.
Toolbar	Werkzeugleiste in einer Anwendung
Tooltips	Informationen, die für Bedienungselemente eingeblendet werden
Usability	Bedienbarkeit
Virtual Reality	virtuelle Realität, die in Computersystemen simuliert wird
Virtual Reality-Brille	eine Brille, die zur Darstellung einer virtuellen Realität dient
Virtual Reality-Support	technische Unterstützung von Virtual Reality-Systemen

Abkürzungsverzeichnis

2D	2-dimensional
3D	3-dimensional
d.h.:	das heißt
Engl.:	Englisch
GmbH	Gesellschaft mit beschränkter Haftung
Inc.	incorporated
MMORPG	Massively Multiplayer Online Role-Playing Game
N64	Nintendo 64
NES	Nintendo Entertainment System
PC	Personal Computer
SNES	Super Nintendo Entertainment System
UI	User Interface
USB	Universal Serial Bus
usw.	und so weiter
VR	Virtual Reality
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
z.B.:	zum Beispiel

Quellenangabe

Willms André: Go To C++ Programmierung, Addison Wesley Longman, 1999

Forster Winnie, Freundorfer Stephan: Joysticks: Eine illustrierte Geschichte der Game-Controller 1972 – 2004, Utting, 2004

https://gameplan.de/wp-content/uploads/2015/07/gameplan_02_S008-S009.pdf

(Stand: 05.07.2016)

<http://www.shortlist.com/tech/gaming/history-of-the-video-game-controller#>

(Stand: 05.07.2016)

<http://www.nintendojo.com/features/columns/hot-air/hot-air-rumbled/attachment/n64-rumblepak>

(Stand: 05.07.2016)

<http://techreport.com/review/29615/valve-steam-controller-reviewed>

(Stand: 05.07.2016)

<https://unity3d.com/>

(Stand: 04.07.2016)

<http://support.xbox.com/en-US/xbox-360/accessories/controllers>

(Stand: 18.06.2016)

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

(Stand: 10.06.2016)

Verwendete Tools

Unity Version: 5.1.1f1

Visual Studio 2013

Microsoft Expression Blend 2013

OpenOffice.org Writer 3.4.1

PhotoFiltre 6.2.0

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als angegeben verwendet habe. Insbesondere versichere ich, dass ich alle Werke, aus denen ich wörtlich oder sinngemäß Texte entnommen habe, als Quelle angegeben habe.

Ort: Leipzig

Datum: 22.07.2015

René Perlbach

Anhang

Auf dem, dieser Arbeit beigefügtem Datenträger befindet sich:

- diese Arbeit als PDF-Format
- der mit Unity umgesetzte Prototyp
- der in C++ geschriebene Quellcode für die Gamepad-Steuerung von „Crazy Machines 3“