zz



LernApp VR

[Untertitel des Dokuments]



[Datum]

[Firmenname]

[Firmenadresse]

Inhalt

Es wurden keine Einträge für das Inhaltsverzeichnis gefunden.

# Lernspiel für Programmieren / Entwicklungsumgebung in VR

## Welche Zielgruppe soll die Anwendung ansprechen

### Welche Programmiersprache soll oder sollen verwendet werden

* Wichtig für die Bestimmung der Beispiele
* Ist eine Konvertierung nötig oder ist sie überhaupt möglich -> Compiler benötigt

### Mögliche Zielgruppen

* Einsteiger
* Fortgeschrittene
* Experten

### Mögliche Altersgruppen für die Anwendung

* Kinder
* Jugendliche
* Egal welches Alter

## Durch welche Funktionen soll sie sich über einer normalen Lernanwendung abheben

* Interaktives Lernprogramm / Tutorials
* Bessere Visuelle Darstellung von Inhalten
* Höheres Sichtfeld, wodurch mehr Fenster möglich sind
* Leichteres Kopieren und Einfügen von Elementen
  + Erkennung durch Handbewegung
* Integrierte Sprachfunktionen
  + Google Suche
* Ersatz des kompletten Desktops aber effektiver
* Funktionen und Klassen könnten visuell präsentiert werden und auch ihr jeweiligen Funktionen
* Googlen von Grundfunktionen ist nicht mehr nötig

## Welche Methoden zum Lernen von Programmiersprachen gibt es und worin könnte VR den Unterschied dazu machen

## Arbeitet man potentiell schneller mit der Anwendung als mit einer normalen Entwicklungsumgebung

* Tests durchführen

## Probleme der Anwendung

* Der Tragekomfort von VR Brillen ist immer noch nach langen Sitzungen zu schlecht
* Virtuelle Tastatur

# Recherche was andere Entwicklungsumgebungen gut macht

## Auf welche Funktionen achte ich bei der Wahl der Entwicklungsumgebung

* Übersichtliches Design
* Viele automatisierte Prozesse
* Einfacher Test von Code und Bearbeitung der Variablen während der Laufzeit
* Intelli Sense
  + Vorschlag für bestimmte Aktionen wie
    - Imports, Namensgebung der Variablen

## Welcher Teil beim lernen einer neuen Programmiersprache fiel mir schwer

* Merken der Syntax und der Schlüsselbefehle
* Welche Datenstruktur ist zum speichern von meinen Daten sinnvoll und wie ist die Syntax für diese, welche Befehle stehen mir zur Verfügung
* Auswahl von Schleifen und deren Bezeichnern
* Anwenden von Codestyle

# Wie löst meine Anwendung diese Probleme und kann daher effektiver im Lernen und Programmieren sein

## Visuelle Unterstützung

* Anzeigen von Kontrollstrukturen und deren Funktionsweise
  + Kleine Vorstellung und Anwendungsbeispiel für die jeweilige Kontrollstruktur
  + Angabe von Vor- und Nachteilen und Laufzeit
* Anzeigen von Datenstrukturen und deren Funktionsweise
  + Kleine Vorstellung und Anwendungsbeispiel für die jeweilige Kontrollstruktur
  + Angabe von Vor- und Nachteilen und Laufzeit
  + Angabe welcher Datentyp zum Speichern am besten geeignet ist
* Anzeigen von Datentypen
  + Kleine Vorstellung und Anwendungsbeispiel für die jeweilige Kontrollstruktur
  + Angabe wie groß der Datentyp ist
* Anzeigen von Beispielen für:
  + Einfache Ausgabe über die Kommandozeile
  + Loggen von Informationen

# Spaltung der Anwendung in verschiedene mögliche Richtungen

## Lern Anwendung für Anfänger in der Programmierung (Lernanwendung)

* Hilft Einsteigern beim Lernen der Programmiersprache
* Gibt visuelle Beispiele für kleinere Programme
* Hier sind mehrere Programmiersprachen mögliche

## Unterstützung beim Programmieren (Richtige Entwicklungsumgebung) zu Aufwendig?

* Hilft beim Programmieren durch Vorschläge
* Nur eine Programmiersprache möglich
* Größerer Aufwand und schwerer zu Programmieren

# Scratch als Vergleich

* Farben haben immer die gleiche Bedeutung und weisen auf Eigenschaften hin
* Leichter Einstieg für Kinder
* Nicht geeignet um „richtige“ Programmiersprachen zu lernen

# Zwei Richtungen in welche die Anwendung gehen könnte

## Einsteigerhilfe für Neueinsteiger in die Programmierung allgemein

* Ähnlich wie Scratch
* Einfache Programmierung durch Drag und Drop
* Keine richtige Programmiersprache notwendig
* Lernen was programmieren ist

## Einsteigerhilfe in neue Programmiersprachen

* Lernen der allgemeinen Syntax für die Programmiersprache
* Wikipedia Funktion wenn man über bestimmte Elemente Informationen haben will
* Anzeigen von Beispiel für bestimmte Aufgaben
* Visuelle Darstellung von typischen Datenstrukturen

# Vergleichbare Produkte

## Produkte mit VR

* [https://www.roadtovr.cofm/watch-logix-impressive-multi-user-visual-programming-interface-vr/](https://www.roadtovr.com/watch-logix-impressive-multi-user-visual-programming-interface-vr/)
* Riftsketch
* Big Screen VR -> mehr Bildschirme für die Programmierung

## Produkte zum Lehren von Programmiersprachen

### Produkte für Kinder

* Scratch <https://scratch.mit.edu/>
* Brettspiele wie Robot Turtles / Code Master
* Code.org -> <https://code.org/>
* Code Combat -> <https://codecombat.com/>
* Kara -> <https://www.swisseduc.ch/informatik/karatojava/>
* Lego Mindstorm -> <https://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>
* Kids @ digital

### Produkt für jugendliche bis ältere Altersgruppen

* Code Academy <https://www.codecademy.com/>
* Code Avengers <https://www.codeavengers.com/>
* Usw. Udemy Codeschool
* CodinGame <https://www.codingame.com/start>

# Zielbeschreibung nach SMART

* S = Spezifisch das bedeutet: konkret und unmissverständlich zu benennen, worum es geht (im Sinne von „klar formuliert“)
* M = Messbar das bedeutet, das Ziel so zu formulieren, dass später objektiv zu erkennen ist, ob das Ziel erreicht wurde oder nicht
* A = Attraktiv/ Akzeptabel/ Aktiv das bedeutet, den Endzustand positiv zu beschreiben das bedeutet, dass das Ziel von allen Beteiligten akzeptiert wird das bedeutet, das Ziel aktiv zu formulieren, d.h. dass aktive Verben verwendet werden müssen
* R = Realistisch das bedeutet, Ziele zu formulieren, die durch das eigene Verhalten aktiv beeinflusst werden können Ziele sollten gleichzeitig machbar und herausfordernd sein
* T = Terminiert das bedeutet, die Zielkontrolle zu unterstützen, indem bei der Formulierung festgelegt wird, zu welchem (konkreten) Zeitpunkt das Ziel erfüllt sein soll

# Welche Stakholder gibt es

* Entwickler (Ich)
* Auftraggeber (Herr Kruse)
* Nutzer

# Dokumentation

* Zu jeden Bereich Dokumentation führen

# Pflichtenheft

## Funktionale Anforderungen

* Architekturkonzept
  + Benennung der Komponenten und der Kernfunktionen
* Use-Case
  + Wie interagieren die die Nutzer mit dem System
  + Welche Architekturkomponenten erfüllen welche Funktionen

## Nicht-Funktionale Anforderungen

* Verfügbarkeitskonzept
  + Wann soll der Dienst zur Verfügung stehen
* Skalierung
  + Wie viele Nutzer können es benutzen
  + Welche Daten fallen an
* Konzept Wartung und Konfiguration
  + Umsetzung von Wartungen mit neuen Versionen
* Überwachung und Statistik
  + Sicherheitsmaßnahmen die Überwacht werden müssen
  + Führen von Statistiken
* Performance
  + Wie lange benötigt das Programm bis zur Fertigstellung seiner Funktion
* Komptabilität
  + Mit welchen System soll die Software kompatibel sein

# Leistungsumfang

* Architekturleistung
  + Design
* Entwicklungsarbeiten
  + Installation von Software
  + Einrichtung von Software
* Systemleistungen

# Lieferumfang

* Format
* Executables
* Konfigurationen
* Dokumente

# Einteilung des Projektes

* 30% Design
* 30% Implementierung
* 30% Test
* 10% Reserve

# Auswahl Geräte zur Darstellung und Software dazu

## Welches VR Headset

* HTC Vive

## Logitech Tastatur ?

<https://www.youtube.com/watch?v=QbZgSlHAh1E>

## Welche Software

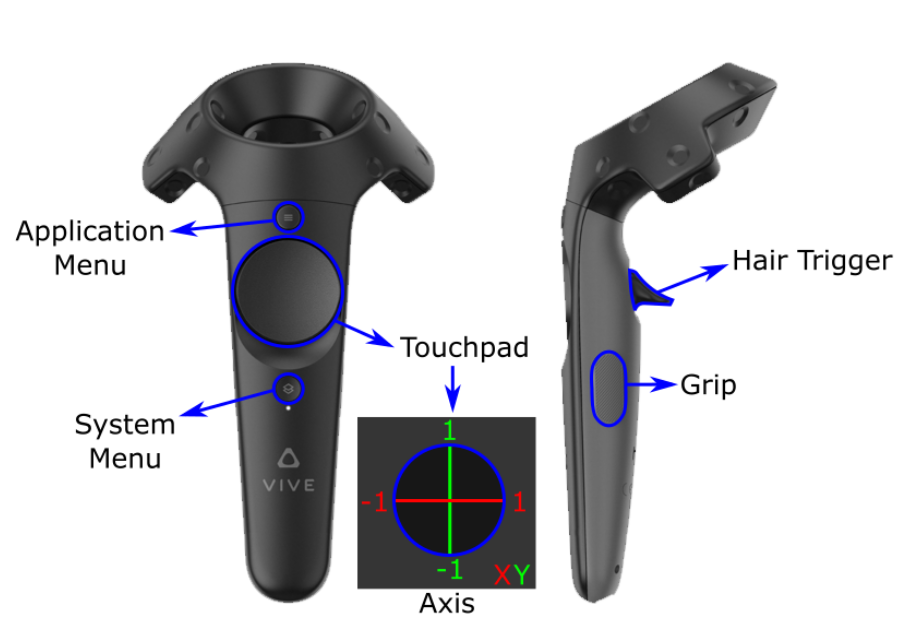
* OpenVR
* VRTK

# HTC VIVE / Pro

## Headset

## Controller

### Welcher Input ist möglich



#### Welche Einstellungen sind möglich

* Einstellung von Geschwindigkeit des Controllers und der Geschwindigkeit der Drehung

#### Welche Tasten können benutzt werden

* Jede Taste kann mit Einfach Doppelt Lang oder Gehalten verwendet werden

# Welche Themen sollen gelernt werden

## Sortieralgorithmen

### Binary Tree Sort

### Bubblesort

## Schleifen

## Datentypen

## Objektorientierte Programmierung

# Steuerungskonzept

## Welche Steuerungen werden benötigt

* Laserpointer zum Zeigen auf Dinge
* Touchpad für scrollen und Zoom
* Trigger Button für das greifen und auswählen
* Menübutton für das Menü

## Verbleibende Buttons

* Grid Button
* System Button

## Wechseln zwischen Rechts und Links Steuerung

* Einer der Controller führt den Laserpointer für die Auswahl daher Auswahl zwischen Links- und Rechtshänder Modus möglich

## Welche Steuerungskonzepte werden sonst noch verwendet

# Designkonzept

## Begrifflichkeiten

* Kurs
  + Zusammenfassung aus einzelnen Lektionen
* Lektion
  + Behandelt ein Thema einzeln

## Hauptmenü mit der Auswahl von Kursen oder einzelnen Lektionen

* Lucidchart

## Darstellung von Datentypen und anderen Informatikelementen

### Welche Darstellungen werden benötigt

* Daten (Objekte)
* Bits
* Datentypen (Boolean, INT, Float)
* Container / Klassen
* Symbole

### Welche Farben stellen was dar (Allgemein Farben verwenden)

## Was haben die Designs der verschiedenen Aufgaben alle gemeinsam und welche können direkt wiederverwendet werden

## Wie soll Feedback für den User dargestellt werden

### Hilfe

### Belohnung

### Versagen

## Welche Lerntypen gibt es und wie kann dies verwendet werden

### Auditiver Lerntyp

* Lernt durch hören und sprechen
* Sound für die Lektionen miteinfügen

### Visueller Lerntyp

* Lernt durchsehen

### Motorischer Lerntyp

* Lernt durch Anpacken und tun
* Interaktion mit dem Controller kann dafür genutzt werden
* Bestimmte Handbewegungen können an das Wissen erinnern

### Kommunikativer Lerntyp

* Lernt durch aktiven Anteil am Unterricht und sucht die Interaktion
* KI / Guide kann dabei helfen und durch die jeweiligen Lektionen zu führen
* Spracherkennung wäre eine Idee um mehr Interaktion zu fordern

### Personenorientierter Lerntyp

* Lernt direkt mit der Lehrkraft
* Ob KI die richtige Lehrkraft ersetzen kann ist fraglich

### Medienorientierter Lerntyp

* Lernt gut mit technischen Mitteln
* VR Brille klärt das Thema

# Tutorial / Einführung

# Beispiel Design für Bubblesort

## Funktion Bubblesort

* Liste wird von rechts nach links sortiert und es wird immer getauscht, wenn das jeweilige Element kleiner ist
* Wenn kein Element wieder getauscht werden muss ist die Liste sortiert

## Was soll mit der Lektion vermittelt werden

* Die allgemeine Funktion des Algorithmus mit einer visuellen Darstellung
* Ein kleines Bespiel wo der Nutzer selbst sortieren kann
* Analyse der Laufzeit und typische Vorteile und Nachteile

# Namens Ideen

# Notizen

* <https://www.hci.uni-wuerzburg.de/projects/getit/>
* <https://www.raywenderlich.com/9189-htc-vive-tutorial-for-unity>
* CutieKeyboard testen