Dokumentation

1. Benutzerhandbuch und Systemvoraussetzung

Die der HTC Vive beiliegenden Controller haben folgende Tastenbelegung:

Rechte Hand:
- Trigger: Sprühfunktion (Wenn Dose in der Hand), Selektion (Im Farbmenü)
- Pinch/Grab: Dose greifen
- Touchpad: Teleportfunktion

Linke Hand:
- Pinch/Grab: Farbmenü öffnen/schließen
- Touchpad: Teleportfunktion

Nach dem Starten des Spiels befindet sich der Anwender in einer virtuellen Szene einer Häusergasse und kann sich darin sowohl normal als auch mittels der Teleportfunktion für größere Entfernungen bewegen. Auf der Boombox steht eine Spraydose, die aufgenommen werden kann. Die beleuchtete Hauswand kann dann besprüht werden (Die anderen Objekte in der Szene dienen ausschließlich zur Dekoration und können nicht bemalt werden).

Im Farbmenü kann sowohl eine neue Sprühfarbe erstellt und ausgewählt werden, als auch die Sprüheigenschaft verändert werden. Zum Ändern der Farbe können mit verschiebbaren Reglern der Farbwert (*Hue*) und die Farbsättigung (*Saturation*) eingestellt werden. In der Vorschau sieht man die neue Farbe, die von dort aus auch einfach „entnommen“ werden kann. Erst dann ändert sich auch die Sprühfarbe der Dose. Über die Buttons an der Seite kann die Sprühweite ausgewählt werden (schmal (*Skinny*), normal (*Medium*) und breit (*Fat*)). Außerdem kann hier ein Löschspray (*Eraser*) ausgewählt werden oder ganze Malereien rückgängig gemacht werden (*Undo*).

Des weiteren befinden sich in der Nähe der Boombox Buttons, über die das Level gewechselt oder die Simulation verlassen werden kann. Die zweite Szene beinhaltet ein bemalbares Zugabteil.

SteamVR stellt folgende Voraussetzungen an das System:

- CPU: Intel Core i5-4590/AMD FX 8350 äquivalent oder besser.
- Betriebssystem: Windows 7 SP1, Windows 8.1 oder besser, Windows 10
- Grafikkarte: NVIDIA GeForce GTX 970, AMD Radeon R9 290 äquivalent oder besser
- RAM: 4 GB

Die HTC Vive hat folgende empfohlene Systemvoraussetzungen:

- CPU: Intel Core i5-4590 äquivalent oder besser
- Betriebssystem: Windows 7 64bit oder besser
- Grafikkarte: NVIDIA GeForce GTX 970, AMD Radeon R9 290 äquivalent oder besser
- RAM: 8GB
- HDMI 1.3 und USB 3.0

Damit die HTC Vive auch optimal genutzt werden kann, sollte vor dem Start die Kalibrierfunktion genutzt werden (Raumabmessung). Durch die im Spiel enthaltene Teleportfunktion reicht auch ein Bereich von ca. 2x2m² aus.

2. Anweisungen zur Kompilierung des Spiels

Implementiert wurde das Projekt hauptsächlich in Unity 2018.4.x und Visual Studio, wobei die Skripte auch mit anderen Programmen geöffnet werden können. Unsere eigens erstellten Assets wurden in Autodesk Maya 2017 erstellt.

In Unity werden folgende Assets verwendet:

- SteamVR Plugin Version 2.3.2
- OpenVR Version 1.0.5
- TextMesh Pro Version 1.4.1
- Brookly Home Version 1.0
- Railway Tank 15-1427 (Soviet Locomotives) Version 1.2
- Simple Color Picker Version 1.2

Exportiert kann das Spiel dann über die Build-Funktion von Unity, als Szenen werden Scene_Wall und Scene_Train benötigt (im Assets → Scenes Ordner) und als Platform PC, Mac und Linux Standalone ausgewählt. Die ausführbare Datei kann dann bei angeschlossener VR-Brille und Controllern und der laufenden Anwendung SteamVR von Steam gestartet werden.

3. Übersicht über die Struktur des Programms

Unsere Simulation besteht aus 2 Szenen (*Scene_Wall* und *Scene_Train*). Elementar in beiden Szenen ist neben den dekorativen Objekten das Player-Objekt, welches aus dem SteamVR Package entstammt, eine Teleport-Area und die Dose/Zugabteil. Wichtig für unsere Simulation und auch Implementierung der Skripte sind die beiden Hand Objekte.

Von uns implementiert und an die Handobjekte gebunden sind:

- ColorPicker und Activator (Funktionalität des Farbmenüs)
- Cylinder (ein Pointer für eine bessere Selektion)

Für das Farbmenü gibt es zum einen einen *ColorManager*, welches die aktuelle Farbe der Sprühdose verwaltet, und ein *ColorIndicator*, welcher für die Farberstellung (zusammen mit *ColorHuePicker* und *ColorSaturationPicker*) und Preview verantwortlich ist. Der *ColorPickActivator* aktiviert/deaktiviert das Farbmenü.

Die Teleport Fläche und Funktionalität wurden aus dem SteamVR Package übernommen.

Die Dose wurde in Maya modelliert und kann über ein Spawn Objekt erstellt und somit in der Simulation verwendet werden. Die Spawn Mechanik wird auch hier über ein Skript von SteamVR gesteuert. Selbst erstelle Texturen wurden in Photoshop erstellt und einige Objekte wurden mit Bump-Maps erweitert.

Das *RayButtons* Skript kann an Buttons gebunden werden, die über die Controller selektiert werden können und führt die entsprechenden Aktionen aus.

Ein *SprayDrip* Skript kümmert sich um die Funktionalität der Farbtropfen, wenn lange an einer Stelle gesprüht wird.

4. Physik der Simulation

Das Sprühen der Farbe wurde hauptsächlich mit zwei Klassen implementiert. Die Klasse *Sprayer*, welche von der aufhebbaren Sprühdose genutzt wird um die Farbe zu versprühen und die Klasse *SprayTarget*, die für Objekte benutzt wird, die angesprüht werden können. Das Auftragen der Farbe funktioniert indem die Textur des Zielobjektes geändert wird. Hierfür wird zum Start des Spiels die Textur einmal kopiert, damit wir nur auf der Kopie arbeiten. Wenn mit dem *Sprayer* Skript gesprüht wird, wird mit einem Raycast mit dem Dosenkopf als Ursprung auf Kollisionen in der Szene geprüft. Wenn eine Kollision vorhanden ist, wird geprüft, ob es eine Kollision mit einem SprayTarget war. Wenn dies der Fall ist, wird bei diesem Ziel eine Funktion aufgerufen, die auf der Textur Farbe aufträgt. Dabei wird die Texturkoordinate der Kollision als Mittelpunkt für die Farbe übergeben, welche bei dem Raycast automatisch zurückgegeben wird. Es werden außerdem die aktuelle Farbe des Sprayers, die Distanz der Kollision, die Sprühstärke und Distanz- und Radiusmultiplikatoren des aktuellen Sprühkopfes übergeben.

Das *DrawSpray* Skript in *SprayTarget* prüft dann ob die Kollision nahe genug war und berechnet anhand der Distanz wie stark die Farbe aufgetragen werden soll. Um die Textur zu ändern wird erst der Radius der Farbe berechnet und mithilfe von GetPixels ein Array der Farben aller Pixel in diesem Radius aus der Textur erhoben. Die Farben des Arrays werden dann so geändert, dass sie eine Mischung der gesprühten Farbe und der vorherigen Farbe sind, wobei der Anteil der gesprühten Farbe aus der Distanz der Raycastkollision und dem Abstand des Pixels zum Punkt der Kollision berechnet wird.

Falls das Löschspray verwendet wird (Spray mit transparenter Farbe), wird statt der Farbe des Sprays die Farbe der Pixel aufgetragen, die sie in der Originaltextur haben.

Da die Texturengröße von Objekten nicht von der Größe des Objektes abhängen, hat jedes *SprayTarget* eigene Distanz- und Radiusmultiplikatoren, die bei der Malfunktion benutzt werden. So können alle *SprayTargets* angepasst werden, sodass die aufgetragene Farbe mit dem Partikeleffekt der Sprühdose übereinstimmt.

Wurde viel Farbe auf eine Stelle gesprüht, kann es zu Drips kommen. Dies wird beim Sprühen geprüft indem die Farbe am Kollisionspunkt mit der Sprühfarbe verglichen wird und wenn sie ähnlich genug sind, können zufällig Drips erstellt werden. Die Drips verwenden das Skript *SprayDrip*, welches zu beginn ein *SprayTarget*, eine Texturkoordinate und eine Farbe übergeben bekommt. Die Drips nutzen dann in ihrer FixedUpdate Funktion das *DrawSpray* Skript, um auf der Textur ihres Ziels zu malen, wobei sie danach ihre gespeicherte Texturkoordinate ändern, sodass die Farbe aussieht, als würde sie nach unten fließen. Nach ungefähr einer Sekunde wird das Drip Objekt gelöscht.

5. Arbeitsaufteilung

Ole Mauri (cgt101592): Implementierung der Sprayfunktion
Nicolas Löwes (cgt101991): Implementierung des ColorPicker
Denis Chrusciel (cgt101701): Modellierung der Dose, des Zugabteils und der Szenen
Christoph Brandt (cgt101660): Setup, Dokumentation, Shader/Texturierung