



# Haunted

## ▼ 19.08 - Gruppenfindung

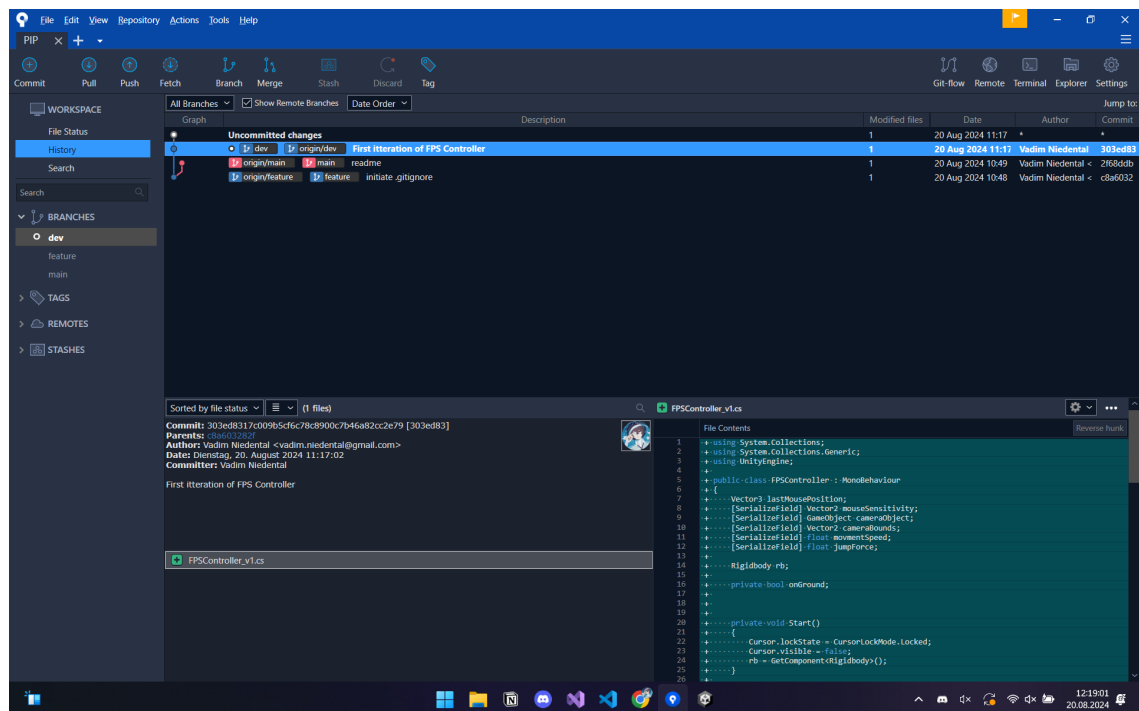
W

## ▼ 20.08 - Sourcetree - Movement

### ▼ Sourcetree

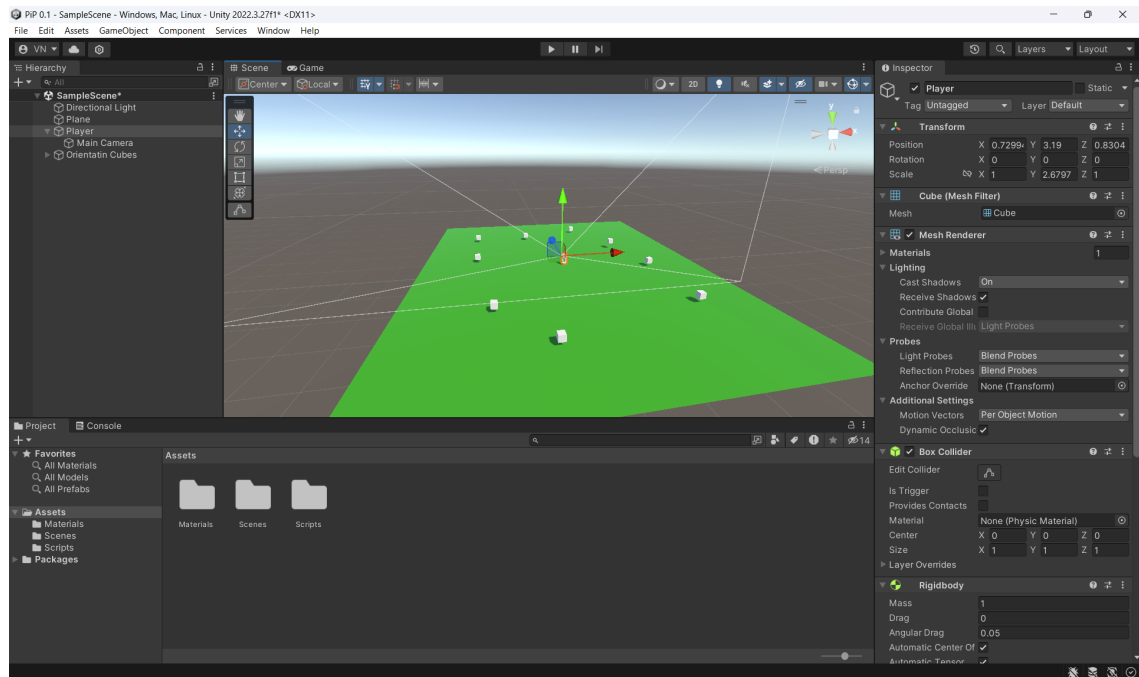
Das Programm Sourcetree wurde mit den Programmieren besprochen, installiert und mit GitHub verknüpft für einen sicheren und leichten Transfer der Codes.

Dient zur besseren Übersicht des Fortschrittes des Codes.



## ▼ Unity

Erstellung eines neuen Projekts (PiP 0.1) zur Aufstellung eines Prototypen und fürs Experimentieren mit der Code-Basis.



## ▼ Scripts

### ▼ Movement

FPSController\_v1.cs

Eine gegebene Codebasis aus dem Unterricht wurde genommen für den First Person Controller.

Anpassung an den Camera Boundaries.

```

1 reference
41 private void RotateCharacter()
42 {
43     var mouseDelta = Vector2.zero;
44     mouseDelta.x += Input.GetAxis("Mouse X");
45     mouseDelta.y += -Input.GetAxis("Mouse Y");
46
47     transform.Rotate(Vector3.up * mouseDelta.x * mouseSensitivity.x);
48
49     cameraObject.transform.Rotate(Vector3.right * mouseDelta.y * mouseSensitivity.y);
50
51     // set max and min rotation
52     float maxRotationX = -90f;
53     float minRotationX = 90f;
54
55     // allow to rotate the camera in x axis
56     var cameraRotationX = cameraObject.transform.localEulerAngles.x;
57
58     if (cameraRotationX > 180f)
59     {
60         cameraRotationX -= 360f;
61     }
62
63     // clamp the rotation
64     cameraRotationX = Mathf.Clamp(cameraRotationX, maxRotationX, minRotationX);
65
66     // apply the clamped rotation
67     cameraObject.transform.localEulerAngles = new Vector3(cameraRotationX, 0, 0);
68 }

```

Ein maximaler und minimaler Wert wurden hinzugefügt, damit sich die Kamera nicht weiter als diese Werte hinaus bewegen kann.

Änderung der Movement Inputs  
if-statements → Switch.

```

12 references
public enum Movement
{
    Forward,
    Backward,
    Left,
    Right,
    None
}

1 reference
private Movement GetMoveInput()
{
    if (Input.GetKey(KeyCode.W)) return Movement.Forward;
    if (Input.GetKey(KeyCode.S)) return Movement.Backward;
    if (Input.GetKey(KeyCode.A)) return Movement.Left;
    if (Input.GetKey(KeyCode.D)) return Movement.Right;
    return Movement.None;
}

```

```

1 reference
private void MoveRB()
{
    Movement movement = GetMoveInput();
    Vector3 moveVector = Vector3.zero;

    switch (movement)
    {
        case Movement.Forward:
            moveVector = Vector3.forward;
            break;
        case Movement.Backward:
            moveVector = Vector3.back;
            break;
        case Movement.Left:
            moveVector = Vector3.left;
            break;
        case Movement.Right:
            moveVector = Vector3.right;
            break;
        case Movement.None:
            break;
    }

    moveVector = moveVector.normalized;
    rb.angularVelocity = Vector3.zero;
}

```

## ▼ 21.08 - Movement - Pick & Throw

### ▼ Script

#### ▼ Movement

Der Jump wurde verfeinert und angepasst, damit er zuverlässig läuft. Dafür wird ein Raycast initiiert, der checkt, ob der Spieler den Boden berührt.

```

134 private bool onGround;
135     1 reference
136 private void CheckForSlope()
137 {
138     RaycastHit hit;
139     Vector3 rayOrigin = transform.position;
140     Vector3 rayDirection = -Vector3.up;
141     float rayDistance = 1.3f;
142
143     /*Debug.Log(onGround);*/
144
145     // Raycast to check if the player is on the ground
146     Debug.DrawRay(rayOrigin, rayDirection * rayDistance, Color.red);
147
148     if (Physics.Raycast(rayOrigin, rayDirection, out hit, rayDistance) && rb.velocity.y <= 0)
149     {
150         onGround = true;
151     }
152     else
153     {
154         onGround = false;
155     }
156 }

```

Die Kamera wurde angepasst, sodass die WASD Tasten relativ zur Kamera-Position sind, und nicht global in eine Richtung.

```

84 private void MoveRB()
85 {
86     Movement movement = GetMoveInput();
87     Vector3 moveVector = Vector3.zero;
88
89     // Create directions for camera
90     Vector3 cameraForward = cameraObject.transform.forward;
91     Vector3 cameraRight = cameraObject.transform.right;
92
93     // Ignore upward movement
94     cameraForward.y = 0;
95     cameraRight.y = 0;
96
97     // Normalize vectors
98     cameraForward.Normalize();
99     cameraRight.Normalize();
100
101
102     switch (movement)
103     {
104     case Movement.Forward:
105         moveVector = cameraForward;
106         break;
107     case Movement.Backward:
108         moveVector = -cameraForward;
109         break;
110     case Movement.Left:
111         moveVector = -cameraRight;
112         break;
113     case Movement.Right:
114         moveVector = cameraRight;
115         break;
116     case Movement.None:
117         break;
118     }
119 }

```

Serialize Fields wurden Werte gegeben

```

8  Unity Script | 0 references
9  ▼ public class Movement : MonoBehaviour
10 {
11     Vector3 lastMousePosition;
12     [SerializeField] Vector2 mouseSensitivity = new Vector2(5f, 5f);
13     [SerializeField] GameObject cameraObject;
14     [SerializeField] float movementSpeed = 0.1f;
15     [SerializeField] float jumpForce = 10f;
16
17     Rigidbody rb;

```

#### ▼ Pick & Throw

Grundkonzept für das a Aufheben und Werfen von ausgewählten Objekten

Code Basis aus dem Internet gefunden .

>> Anwendung diesen Codes und Erstellung einer eigenen Variante.

Der ausgewählte Code beinhaltet Fehler in der Unity Umgebung.

>> Objekt deformiert sich bei einer vertikalen Bewegung der Kamera.

### ▼ 22.08 - Pick & Throw - Game Design

#### ▼ Scripts

##### ▼ Pick & Throw

Das System für das Aufheben, Werfen und Hinlegen von Items wurde auf Version 1.0 gepusht. (Funktionen sind weitestgehend fertig und brauchen keine wirklichen Anpassungen mehr.

Bug der Deformation wurde behoben.

#### ▼ Game Design

#### Items & Potions

##### ▼ Ideen

Unicorn Blood (Silver)

This sacred blood of a unicorn holds fast unimaginable powers.

— Keine Ideen welche Nutzung das haben kann —

>> Regenbogen Raum mit vielen verschiedenen Aspekten, die Abstrakt wirken können

#### Phoenix Feather (Red)

A feather of the most sacred beings of all Heidelbergers Schloss.  
Allows on use to burn specific items, like the ground for example or a wooden plank to reveal something.

>> Roter Raum, der eventuell an die Hölle basiert ist.

#### Moonlight Water (Light Blue)

Water that directly stems from the Moon's atmosphere.  
It has the power to transform the player into a howling moon mouse.

>> Raum der auf Mond und Astronomie basiert. (Von außen klein, innen gigantisch, sofern umsetzbar per Code usw.)

#### Elderflower Essence (Goldish-Yellow)

The essence of the Elden Tree, which once was the rejuvenator of life for itself.

Can grow platforms, which either can be used as weight for a pressure plate or to jump across gaps.

>> Ein anfangs grüner Raum voller Pflanzen, der sich dann zu gelb und an schließlich grau wandelt, mit der Blume als einzelnes gelbes Objekt im grauen Raum.

#### Emerald's Gem (Green)

A valuable gem of its own kind. Its structural integrity and green color are the key for the perfect potion.

With this gem you can see things, which are not visible to the naked eye.

>> Eventuell eine Höhle, mit vielen, funkelnden Kristallen, oder was anderes. Nicht spezifisches geplant.

Anschließend kann man mit all diesen Items den Trank brauen, um das Ziel zu erreichen

Spielmechaniken wurden bestimmt und essenzielle Items wurden ausgedacht.

Items werden notwendig sein um die Rätsel lösen zu können und werden eventuell ein Hauch von Metroidvania im Spiel haben.

#### ▼ 23.08 - Pick & Throw Updates

Grundsätzliche Initialisierung des Aufhebens und Werfens aus Update in eine Separate HandleActions() Methode konvertiert um Probleme zu vermeiden. Weiter wurde ein Raycast für debugging issues aufgestellt.

#### ▼ 19.08 - Inventar - Code Verbesserungen

Brainstorming für Inventar Implementation und Nutzung.

~~Inventar auf Knopfdruck~~  
~~Ein Slot Inventar~~

Minecraft Slot System für die jeweiligen Items. Eventuell eine Drop Methode auf Q später um die Kombination mit dem Item Merger im Nachhinein zu vereinfachen / verbessern.

Grundlegende Verbesserung und Anpassung der Codes

#### ▼ 28.08 - Inventar

##### ▼ Scripts

##### ▼ Inventar

Aufstellung des Slot Inventars für Items

<https://youtu.be/Hj7AZkyojdo?si=Vq7cefAGQ8As3hGL&t=197>



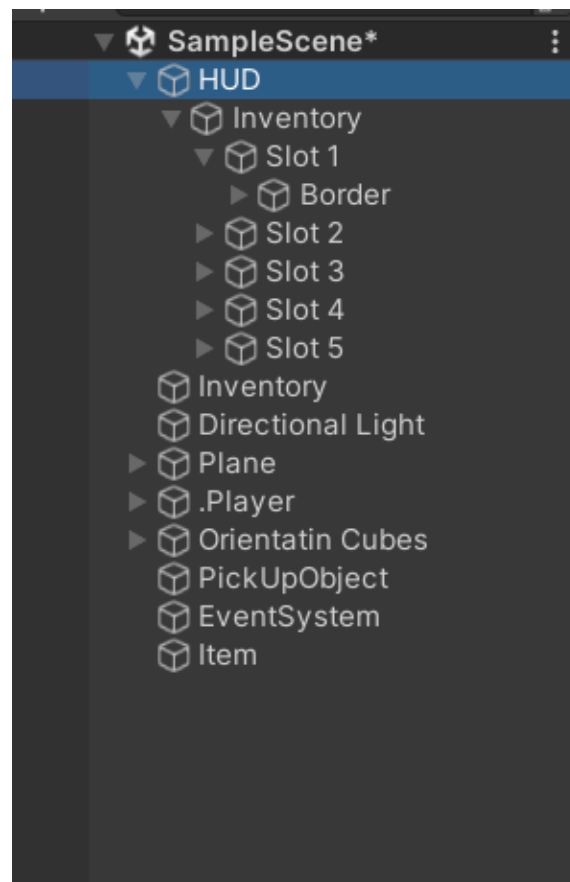
### ▼ 03.09 - Inventar - HUD - Item Inventar

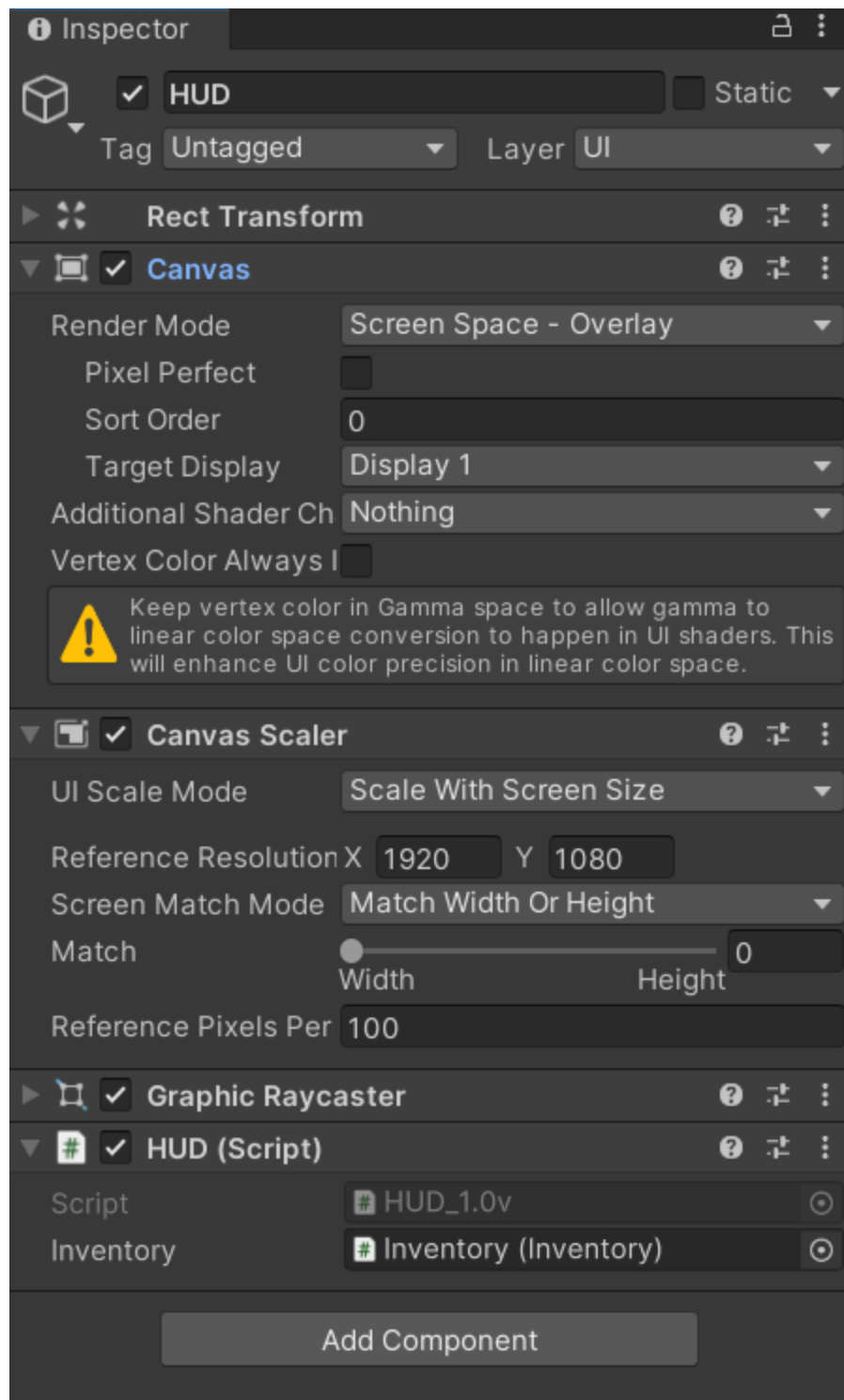
#### ▼ Scripts

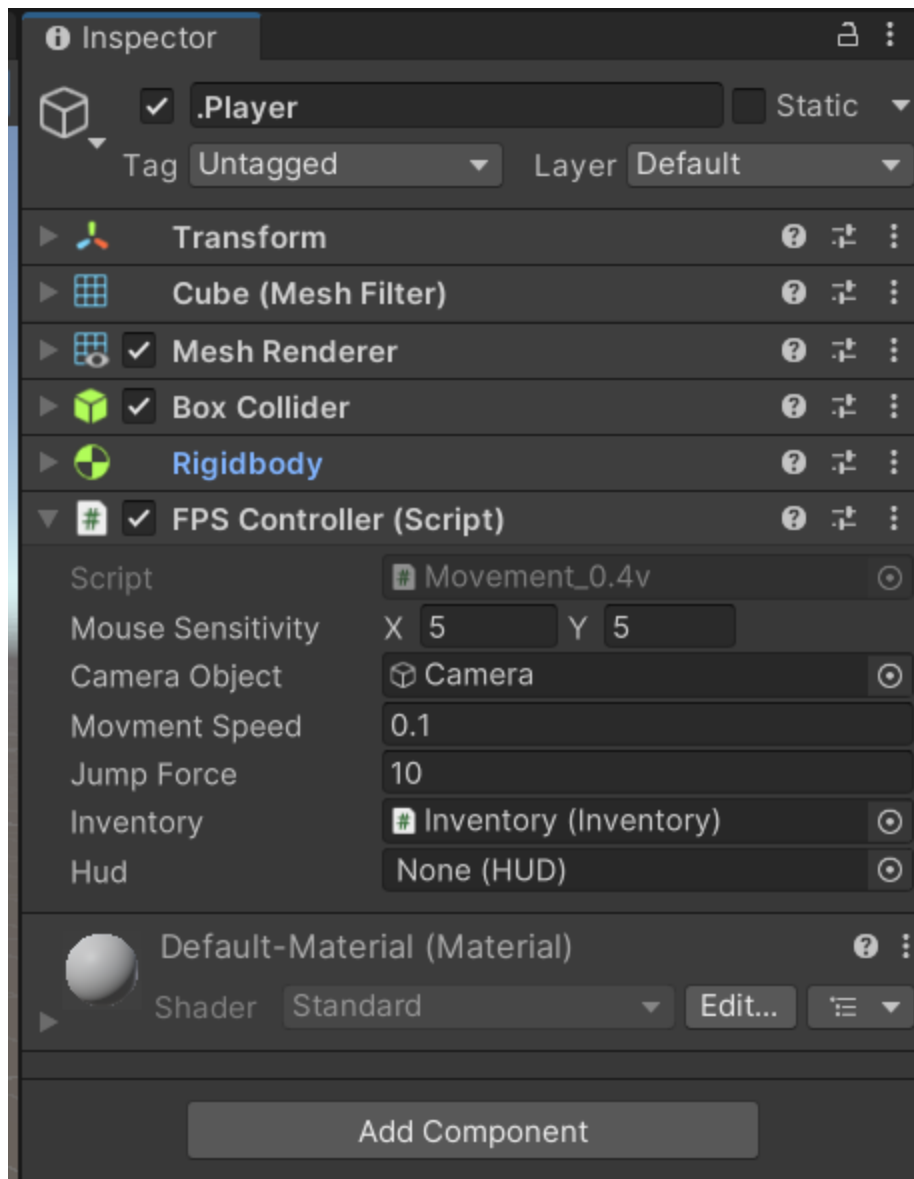
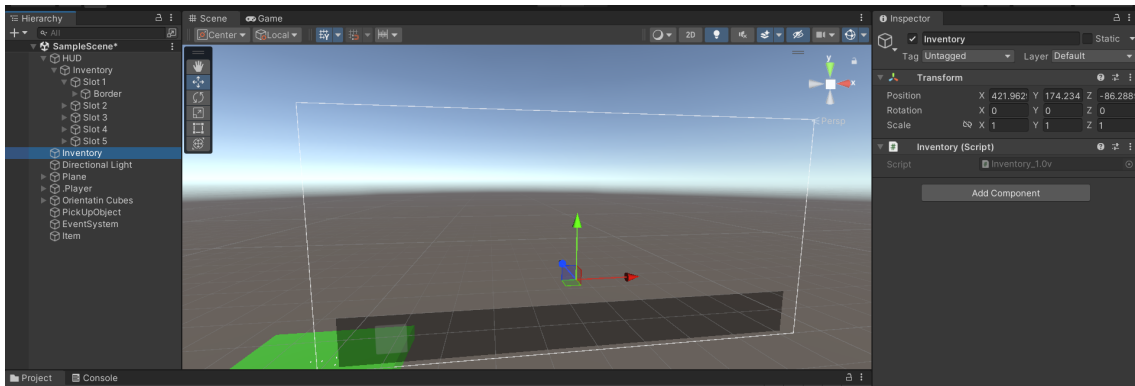
Viele Scripts wurden verändert und das Inventar wurde nun vollständig implementiert mit allen Funktionen.

#### ▼ Unity

In Unity wurden viele Objekte angepasst und verändert um sich mit den neuen Scripts des HUDS anzupassen.







Item ausgewählt / nicht ausgewählt:



#### ▼ 19.08 - Item Inventar > Pick & Throw

Export einiger Methoden und Variablen aus dem Pick & Throw Script in das Inventory Items Script.

Ziel dieser Änderung ist das Aufheben von Objekten zu verhindern, solange ein Item im Slot ausgewählt ist.

Anpassungen am Item Inventar - Automatisches Abwählen von Items, wenn ein anderen Item gewählt wird.

#### ▼ 05.09 - Zusammenführung der gesamten Codebasis

Unity Projekt wurde auf ein Projekt begrenzt. Die Scripts von Philip wurden implementiert.

#### ▼ 06.09 - Informierung zum Fortschritt der Untergruppen

Allgemeine Planung zum weiteren Aufbau des Spiels und Ziele für die Woche.  
Geschaut inwieweit die Modelle fertig sind.  
Kurzes Guide zu FBX angesehen.

#### ▼ 10.09 - Zusammenführung der gesamten Codebasis - Sicherstellung - 3D Modelle

Unity Projekt wurde auf ein Projekt begrenzt. Die Scripts von Kazim wurden implementiert.

Die neue Codebasis steht und das Unity Projekt ist auf Version 0.2 mit allen Funktionen der Programmierer

Projekt Version 0.2 wurde auf dem USB gesichert und den 3D'lern gegeben zum austesten der Methoden und Ideensammlung des Level Designs.

Experimente mit 3D Modellen in Unity und Mesh Objekten

#### ▼ 11.09 - PlayerMotor - Gem Ability - Shader - Export

##### ▼ PlayerMotor

Anpassung des Movements und Ergänzung von normalSpeed und sprintSpeed Variablen

##### ▼ Gem Ability

Collision und RB aktiv lassen, auch wenn nicht sichtbar.

Ermöglichung neuer Rätsel

Plan für später >> PlayerMotor Movement deaktivieren, if Gem active

##### ▼ Shader

Transparenter Shader für Gem hinzugefügt als Overlay in Unity

##### ▼ Export

Bereitstellung des Exports und Nutzung der Unity Datei für die Teammitglieder.

#### ▼ 12.09 - Player Motor - PlayerSwap - PickThrow - Scene

#### ▼ PlayerMotor

Spieler ist am Start des Spiels nicht mehr geduckt

#### ▼ PlayerSwap

Beide Modelle des Spielers laufen gleichzeitig, jedoch ist die jeweils andere deaktiviert bis man nicht gewechselt hat.

#### ▼ PickThrow

PickThrow Script deaktiviert wenn das Modell gewechselt ist.

#### ▼ Scene

Erstellung und Unterteilung der Szenen PlayGround und Main. Der PlayGround dient für alle mechanischen Elementen des Spieles, zum Probieren und Exportieren in neue Szenen.

Die Main Scene dient für den eigentlichen Aufbau und Test des Spiels mit 3D Modellen.



#### ▼ 13.09 - Austausch - Grobkonzept Level Design

##### ▼ Austausch

Austausch mit anderen Abteilen der Gruppen, größtenteils mit 3D

##### ▼ Level Design

Erstes Grobes Konzept für den Aufbau des Levels aufgestellt.

#### ▼ 17.09 - Item Abilities - Bug Fixes

##### ▼ Item Abilities

Jedem Item ein Individuelles Script gegeben zum hinzufügen ins Inventar.

##### ▼ Bug Fixes

Kleinartige Bug Fixes im Interaction Script

#### ▼ 18.09 - Movement - Flower Ability

## ▼ Movement

Ducken entfernt und eine bool gegen das Sprinten in der Luft hinzugefügt.

## ▼ Flower Ability

Erstellung eines Scriptes, welches ausgewählte Objekte vor dem Spieler generiert. Funktion mit Raycast und Collision Detection.

## ▼ 19.09 - Flower Ability

### ▼ Flower Ability

Generelle Verbesserungen im Script

## ▼ 20.09 - Feather Ability - Time Changer - Bug Fixes

### ▼ Feather Ability

Hinzufügung eines Scriptes, welches bei der Verwendung des Items einen Radius zur Detection erschafft, um Objekte mit dem Tag "burnable" zu zerstören.

### ▼ Time Changer

Erstellung eines Scriptes, bei Collision mit bestimmten Objekt ändert sich der Zyklus von Tag zu Nacht.

### ▼ Bug Fixes

Kleine Bug Fixes und Umstrukturierung einiger Scripts.

## ▼ 23.09 - Krank



## ▼ 24.09 - Krank



## ▼ 25.09 - Horn Ability - Gem Ability / Reveal Ability

### ▼ Horn Ability

Beifügung des HornAbility Scripts und ItemHorn Scripts für Inventory.

Dient als eine Art Fackel für dunkle Passagen im Spiel.

Kleinfügige Probleme mit der Lichtquelle

>> Das Item selbst hat keine Lichtquelle, stattdessen wird eine Externe Lichtquelle außerhalb des Spiels für die Quelle genutzt, und bei Benutzung auf die Position des Spielers gesetzt.

### ▼ Gem Ability

Anpassungen in Unity wurden vorgenommen damit alles funktioniert, des weiteren wurde das Script überarbeitet, damit es richtig die Funktion erfüllt. Umsetzung in das Inventar und Item.

## ▼ 26.09 - Dialogue - Key - Barrel Puzzle - Gem Ability - Textures

### ▼ Dialogue

Ein neues und überarbeitetes Dialogue Script wurde aufgestellt. Funktioniert per Collision Detection, wenn der Spieler drin ist, kann er mit dem ausgewählten Objekt interagieren und auf M0 weiterklicken.

//> Eventuelles Problem in der Logik - Index wird nicht richtig gezählt;

Index 0 erlaubt nur zwei Lines und skippt den Rest,

Index 1 nur eine

Index -100 lässt alle Lines erscheinen ohne überflüssigen Fehlern

### ▼ Key

Ein Schlüssel wurde beigefügt in die Spielumgebung und mit dem PickThrow Script aufgebaut. Sofern der Schlüssel mit der Tür collided öffnet sich die Tür und der Schlüssel wird zerstört.

### ▼ Barrel Puzzle

Ähnlich zum Key Script nur wird das Objekt nicht zerstört bei collision.

### ▼ Gem Ability



Verbesserung des Gems und Bug Fixing.  
Erneute Inkorporation mit Inventory.

▼ Textures

Versucht Textures einzufügen zu Modellen, vergebens.

▼ 27.09 - Drop Items - Item Merger - Horn Ability - Player Change - Textures

▼ Drop Items

Items im Inventar können auf Q nun gedroppt werden, um später mit dem Item Merger zu interagieren.

▼ Item Merger

Anpassung des Item Merger Scripts und der Item Scripts. Der Merger hat ein ähnliches Detection System wie der Dialogue. Dem Merger wurde ein RB beigefügt damit die Items mit Trigger interagieren können.

>> Die Items dürfen kein RB haben.

▼ Horn Ability

Beifügung, dass wenn das Licht aktiv ist und man M0 auf eine Fackel drückt, zündet sich die Fackel an.

▼ Player Change

Bool Variable beigefügt mit einer Raycast Collision Detection um zu verhindern dass sich der Spieler zurück verwandeln kann, wenn er mit einer Decke kollidiert.

▼ Textures

Informierung über Blender, FBX und Unity bezüglich der Texturen.

Großes Problem durch den Fehlerhaften Export der Modelle als FBX von 3DE gefunden.

>> Die Texturen und Materials konnten nicht extracted werden von der FBX.

>>> Musste die .blend Datei öffnen, einen eigenen Export mit den richtigen Einstellungen durchführen damit die Texturen und Materials extrahierbar sind.

>>>> Nicht alle Texturen wurden extrahiert, weswegen manuelles arbeiten mit Export der PNGs aus Blender und Einfügen in die Unity Ordner nötig war. (es ist SEHR mühsam und Zeitaufwendig)

## ▼ 28.09 - 3D Modelle - Unity Aufbau

### ▼ 3D Modelle

Einführung der finalen 3D Modelle und Aufbau des gesamten Levels in Unity

## ▼ 29.09 - Großartige Script Überarbeitungen

### ▼ Item Merger

Funktionalität geändert, dass es nicht mehr mit den Inventory Items funktioniert, sondern mit ausgewählten Items für das Rätsel im Alchemie Turm

### ▼ PlayerLook

Sensitivität angepasst

>> Time.deltaTime entfernt damit eine consistency bei dem Player herrscht

### ▼ PlayerInteractObject

Ein Interactable Script für Objekte wurde beigelegt, ähnlich wie der Interact Script

>> Damit kann man z.B. Hebel umheben, damit eine Tür aufgeht.

### ▼ Saw

Script für die Säge erstellt, damit sich die Säge rotiert und bewegt.

### ▼ Respawn

PlayerDeadly, SetSpawnPoint und PlayerSpawn Scripts erstellt um einen Respawn zu erstellen.

>> Es gibt Probleme bei der Logik

>>> Der Spieler wird an den letzten Spawnpoint teleportiert, jedoch wird der Spieler gleich wieder danach sofort an die Death Position teleportiert.

#### ▼ Lever

Nötig für PlayerInteractObject, öffnet eine Tür bei Interact auf E

#### ▼ DialogueTrigger

Collision für den Trigger vom Dialogue erstellt

>> Script hat sich genauso wie das vorherige zerschossen

>>> Der Text Input des Dialoges wird entweder einzeln in Buchstaben gezeigt, oder erst angezeigt wenn man den Skript > Verschwindet aber sofort

#### ▼ Dialogue Courtyard / Moonlight

Collision Blöcke für den Dialogue der bestimmten Orte im Level

#### ▼ HornAbility

Anpassung des Scriptes, dass auch Objekte mit dem Tag "Torch" angezündet werden können. Wenn zehn dieser Objekte angezündet sind wird das Moonlight Water gespawnt

>> Geringfügiges Problem in der Logik

>>> Counter läuft auch beim ausschalten der Torch weiter und mehrmals bei einer Torch.

#### ▼ 30.09 - Zusammenstellung des Spiels

