**Game Jam – Fundamentals**

**Inhaltsverzeichnis**

**Inhalt: Seite:**

[1 Informationen 2](#_Toc103930917)

[1.1 Allgemeines 2](#_Toc103930918)

[1.2 Hinweise/ Regeln vom Veranstalter 2](#_Toc103930919)

[2 Richtlinien zur Erleichterung für alle zum Arbeiten. 3](#_Toc103930920)

[2.1 Scripting 3](#_Toc103930921)

[2.2 Objekte und Texturen 3](#_Toc103930922)

[2.3 UI und Interface 3](#_Toc103930923)

[2.4 Sounds und Music 3](#_Toc103930924)

[3 Dokumentation 4](#_Toc103930925)

[4 Credits 4](#_Toc103930926)

[4.1 Team 4](#_Toc103930927)

[4.2 Weitere Credits 4](#_Toc103930928)

1. Informationen
   1. Allgemeines

* **Name: The GameDev.TV Jame Jam 2022**
* <https://itch.io/jam/gamedevtv-jam-2022?utm_source=Klaviyo&utm_medium=email&utm_campaign=Game%20Jam%20Announcement%20%28HB%29&_kx=VUu0H3pFilXvB6bGeIAZoupAvLUlctx89ITP_Ry2ZPw%3D.VVjemm>
* **Kampfzeit: 20.05 – 22Uhr bis zum 30.05 – 00:00 Uhrr**
  1. Hinweise/ Regeln vom Veranstalter
* Fehlerhafte Einsendungen sind in Ordnung! **Hotfix-Patches sind auch nach dem Einsendeschluss erlaubt, wenn ihr Fehler beheben wollt.**
* Alle Beiträge müssen den Quellcode als Teil der Einsendung enthalten oder einen Link zum extern gehosteten Quellcode (DropBox, GitHub, etc.)
* Die Verwendung von bezahlten Assets ist NICHT erlaubt, aber kostenlose Assets und solche unter einer Open-Source-Lizenz wie CC0 sind erlaubt.

1. Richtlinien zur Erleichterung für alle zum Arbeiten.

* Planen für die Hälfte der Zeit
* Playtests - das Spiel muss am Ende Spielbar sein
* Fokus sollte Fertigstellung liegen!
  1. Scripting
* ! Alles kommentieren! Auch wenn es nur Kleinigkeiten sind, denkt daran, jeder sollte stehts verstehen, was ihr wo implementiert.
* Möglichst aussagekräftige Variablennamen, Camelcase (bspw. int frameCount)
* Helper-Structs/Klassen immer zuerst im Skript definieren
* Function-Encapsulation: Funktionen kurzhalten und Hilfs-Methoden mit wiederverwendeter Funktionalität anlegen
* Singletons für Scene-Managers wird empfohlen (nur eine Instance notwendig)
* Zu Beginn möglichst oft [Range(x,y)] verwenden um variablen im Inspektor anzupassen
* Für states/Behaviourtrees/etc. Verhalten mit Debug.Log analysieren, um Bugs und unerwünschtes Verhalten zu spotten
* Für states zunächst primitives Verhalten, erweitern geht immer(Abstraktion)
  1. Objekte und Texturen
* keine Bodenplane
* Texturatlas -> Eine Textur für mehrere Objekte – spart Ladezeit in Unity
* Texturen so klein wie möglich in der Auflösung halten
  1. UI und Interface
  2. Sounds und Music

1. ****Dokumentation****

**Verwendete Programme:**

* Git -> Filesharing und Versionskontrolle (hier hört bitte alles auf Jojo)
* Unity -> **Version 2020.3.29f1** - Bitte an alle dies betrifft **vorher** herunterladen.
* Blender 3D -> für 3D und unwarping ◉ Texturen -> Libresprite, Gimp, Canva
* für Texturen und Sounds, jeder was er gerne mag. (ich würde das gerne für die Dokumentation des ganzen hier mit aufführen)
* Kommunikation: Discord
* Planung: <https://webwhiteboard.com/>
* Organisation: <https://trello.com/>

**Bezeichnungen von Assets**

* obj\_name – Objekte
* txt\_name – Texturen
* mat\_name – Materialien
* son\_name – Sounds
* scr\_name – Scripts
* bgm\_name - Music

1. Credits
   1. ****Team****

kümmert euch hier bitte selbst darum wie ihr in den Credits stehen möchtet

* cartooni Story/ Dialoge/ Texte
* Jojo Programmierung (lead)
* tobster90 Programmierung
* LvR Prod. Musik- und Sounddesign
* PhilterPhil Texturen und Gamedesign
* Prayer90 3D-Objekte und Gamedesign
* RLP UI - Design/ Game Design
* WildRikku UI – Design
  1. Weitere Credits

verwendete Assets fortlaufend mit aufführen