

Technical Description

Für die Simulation der Gravitation habe ich ein neues Component Script erstellt: **GravityComponent**. Dieses stellt Methoden bereit, um Gravitation zwischen diesem und allen anderen von Gravitation beeinflussten Objekten zu simulieren.

Die Massen der Planeten und der Sonne sind den tatsächlichen Massen nachempfunden und auf für Unity passende Größen verkleinert worden, wobei das Verhältnis der Massen beibehalten wurde. Auch die Entfernung zur Sonne wurde (so gut wie möglich) den echten Werten nachempfunden. Die Simulation wird von einem **Simulator** Objekt mithilfe des **SimulatorComponent** Scripts gesteuert. Dieses kümmert sich um das Finden aller zu simulierenden Objekte und auch um das Abfangen der Button UI Events.

Ein weiteres **PlanetComponent** kümmert sich noch zusätzlich um die Eigenrotation des Planeten. Die Eigenrotation wurde ebenfalls den echten Werten nachempfunden und an die Simulationsgeschwindigkeit angepasst.

Um die Rotation der Planeten um die Sonne richtig zu simulieren, hat jeder Planet eine **initiale Geschwindigkeit** (einen initialen Impuls) erhalten. Dies ist der echten Physik nachempfunden und wurde mithilfe dieser Formel berechnet (s.

[https://de.wikipedia.org/wiki/Bahngeschwindigkeit_\(Astronomie\)](https://de.wikipedia.org/wiki/Bahngeschwindigkeit_(Astronomie))):

$$v = \sqrt{G * (M + m) * \left(\frac{2}{r} - \frac{1}{a}\right)}$$

Das UI wurde mit TextMeshPro erstellt und sendet Events an das Simulator GameObject, welches die Events im SimulatorComponent bearbeitet.

Credits

Folgende Assets wurden vom Asset Store verwendet:

- UI - <https://assetstore.unity.com/packages/2d/gui/sci-fi-gui-skin-15606>
- Skybox - <https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717>
- Planets - <https://assetstore.unity.com/packages/3d/environments/planets-of-the-solar-system-3d-90219>