# Projekttagebuch

Lukas Mahir Valeria

22.06.2022: Brainstorming von Projektideen + Klären erster Spielgrundlagen

24.06.2022: Erstellen der Lasten-/Pflichtenhefte

29.06.2022: Erstellen eines vorläufigen Klassendiagramms und Beginn der Programmierung mit Erstellung der benötigten Klassen und den leeren Methoden

1.06.2022: Beginn der eigentlichen Implementierung:

* Aufteilung der Spielfunktionen durch das MVC-Entwurfsmuster:
  + GameEngine als Controller für Spiellogik: Steuert den Spieler und berechnet Kollisionen etc.
  + GameRenderer als View: Bekommt alle zum Anzeigen benötigte Daten von der GameEngine geliefert und zeigt diese bei Bedarf an
  + GameObject als Model: Primitives Object was Eigenschaften der Spielobjekte speichert z.B. Position, Form, Image, CollissionType
* Implementierung von GameObject und dessen Unterklassen wie z. B. Spieler, Wand, Boden mithilfe des Kompositums
* Beginn des Leveldesigns

2.06.2022: Implementierung von weitern Funktionen der GameEngine:

* Simulation von physikalischen Eigenschaften des Spielers
* Korrekte Berechnung von Kollisionen des Spielers + Benachrichtigung der GameObjects durch die Observer-Pattern
* Erstellen eines WindowManagers zum einfachen Tauschen von verschieden Scenes wie z. B. MainMenu und GameScene/GameRenderer
* Fertigstellen des vorläufigen Level Designs
* Beginn des Texture-Designs

6.06.2022-7.06.2022: Implementierung wichtiger Funktionen für den Spielablauf:

* Wenn der Spieler sich außerhalb des Bildschirmes befindet, wird er auf die andere Seite teleportiert wenn sich dort kein Objekt mit dem der Spieler kollidiert befindet
* Erstellen von Collectibles:
  + Werden aufgesammelt, wenn der Spieler sie berührt
  + Können selbst die Aktionen danach bestimmen 🡪 Collectibles sind sowohl das model, als GameObject, haben aber auch eigene wichtige Funktionen wie z.B. onCollect();
* Messung der Dauer bis zur Levelbeendigung
* Implementierung eines Scores:
  + Berechnung durch Dauer & Score (Durch gesammelte Collectibles bestimmt)
* Beginn der Implementierung von einer Rampe: Slope()
* Erstellen einer Bananen-Texture zur Beendigung des Spiels
* Erste Implementierung des zuvor erstellten Level Designs

10.06.2022: Hinzufüge von Schrägen

* Kollisionserkennung für Physik-Simulation verbessert

13.06.2022:

* Implementierung des Level-Designs
* Hinzufügen von einem Hauptmenü

17.06.2022:

* Fertigstellen der Implementierung des Levels
* Erstellen eines Back-Ends für die Speicherung der Scores mithilfe von JavaScript und MySQL

19.06.2022:

* Erstellen eines Score-Menüs
* Erstellen einer Datenbankverbindung in Java
* Speichern des Scores + Zeit
* Textures geupdatet
* Level-Beendet-Overlay erstellt