

Projekt Beschreibung CT

- Thema: Jump ,n' Run
- Name: ?
- Ziele:
 - Physikberechnung
 - Laufender Charakter
 - Mehrere Level
 - Powerups
 - Gegner
 - Level selber erstellen (Level Editor)
 - Spielstand speichern
- Vielleicht:
 - Netzwerkfähigkeit (Mapsstore, Multiplayer)
 - Gegenstände, verschiedene Charaktere
 - Ordentliches Design

Wie das/unser Jump ,n' Run später aussehen soll:

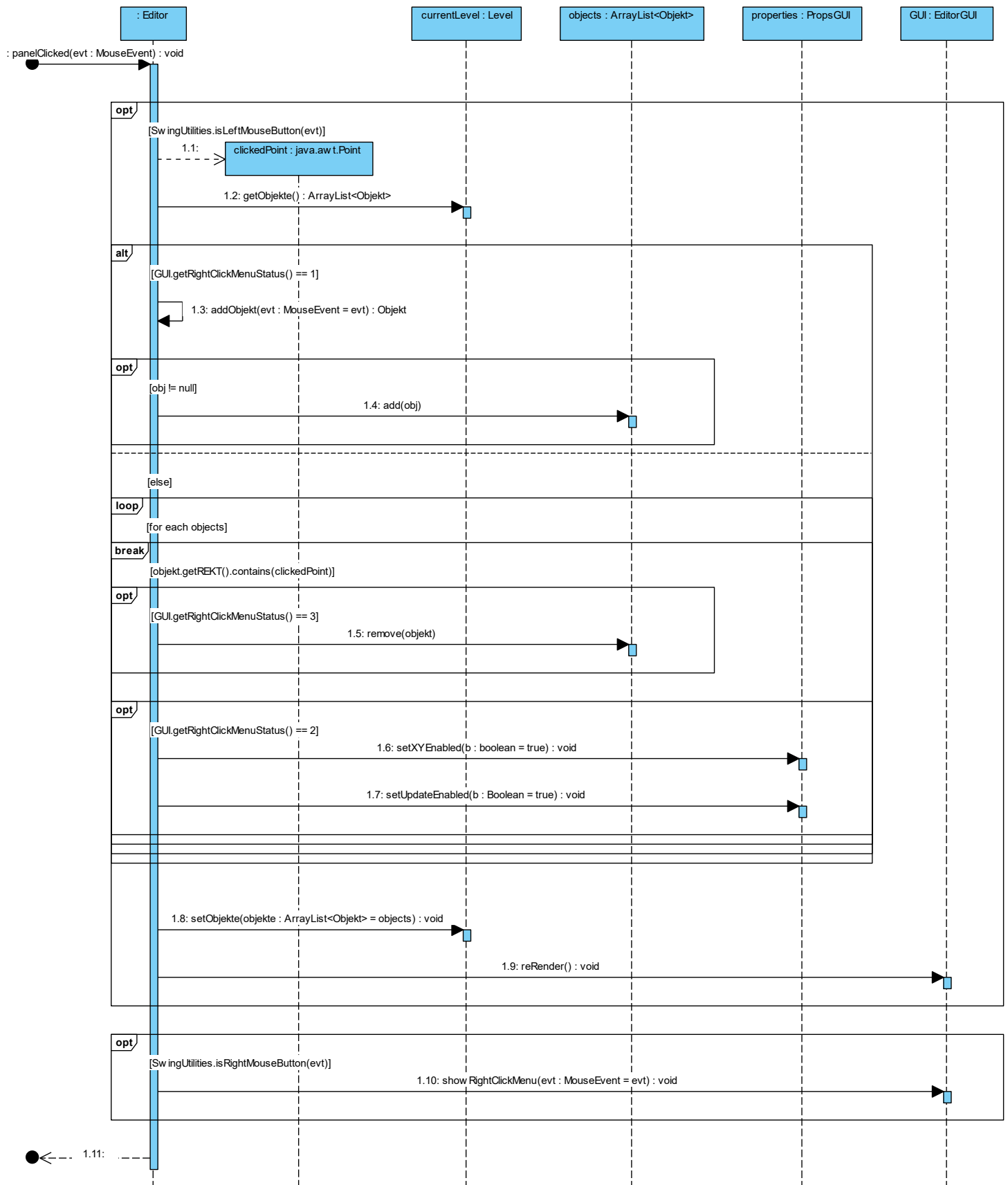
Selbstverständlich sollte es zunächst einmal ein Menü geben, womit man mit unterschiedlichen Buttons navigieren kann.

Letztendlich soll das Spiel wie ein Typisches Jump ,n' Run aufgebaut sein. Das heißt, es soll eine Figur geben die durch das Drücken von Tasten laufen und springen kann. Hierbei sind die Physikberechnungen wichtig. Natürlich soll die Figur einen Weg zurücklegen um ans Ziel zu gelangen. Durch das Erlangen des Ziels soll dann, dass nächste Level freigeschaltet werden. Dieser Prozess soll dem Spieler natürlich möglichst schwer gemacht werden. Wobei der Spieler dann mit der Spiel-Figur schwierige Sprünge zwischen Plattformen meistern muss. Falls dies nicht klappt soll die Spiel-Figur dann Schaden erleiden oder ein Level von vorne beginnen, was typisch für ein Jump ,n' Run ist. Wenn möglich soll es später auch Gegner, Powerups, Gegenstände oder unterschiedliche Charaktere geben. Die verschiedenen Levels sollen durch einen Level Editor errichtet werden. Falls es zeitlich reicht, soll es natürlich auch noch ein Ordentliches Design und andere zusätzliche Funktionen geben.

Bis zur ersten Abgabe geschafft (06.07.2017):

Bei Programm Start öffnet sich ein Fenster indem man gefragt wird, ob man das Spiel oder den Level Editor starten will. Bei der Wahl des Buttons „Spiel“ öffnet sich das Spiel und man kann zwischen dem „Einstellungen“ und „Spiel starten“ Button unterscheiden. Durch das Klicken des Buttons „Spiel starten“ soll später eine Level-Auswahl erscheinen, in der man zwischen unterschiedlichen Levels auswählen kann. Bis dahin sind die Levels per „direkten“ Klick auf die Datei aufrufbar. Wenn dies geschieht ist nun ein Frame mit einem Hintergrund zu erkennen. In diesem kann man ein Kästchen hin und her bewegen. Dies gelingt durch die Tasten a,s,d und w. Wobei man mit letzterer springen kann. Durch die Physikberechnung kann sich das Kästchen in der „Luft“ etwas hin und her bewegen und gelangt nach kurzer Zeit wieder zu „Boden“. Levels lassen sich durch den vorhin erwähnten Level Editor erstellen. Hierbei kann man verschiedene große Plattformen mit unterschiedlichen Farben und Bildern herstellen und auf dem Frame hinzufügen. Bei einem Fehler kann man diese auch wieder entfernen. Außerdem kann man sowohl die Start-Position der Spielfigur als auch Ziele festlegen. Die hergestellten Levels werden natürlich separat auf Dateien gespeichert und sind über diese bei Spiel Start auch wieder aufzurufen. Durch die Kollisions-Abfragen erkennt das Programm wo sich eine Plattform befindet. Dadurch wird dem Kästchen ermöglicht sich auf einer Plattform zu bewegen, ohne dass es durchfliegt. Hierbei müssen noch Kleinigkeiten im Bezug zur seitlichen Kollisionserkennung verbessert werden. Durch das Drücken der Taste „P“ soll man später das Spiel pausieren lassen können. Zusätzlich kann man zwischen verschiedenen Buttons auswählen und somit wieder ins Menü gelangen oder ein Level neustarten. Generell haben wir schon einige von unseren „Haupt“ Zielen erreicht. Jedoch müssen wir vor allem in der Richtung des Spielsinns noch etwas arbeiten. Also z.B. wie die Figur durch das Herunterfallen „Schaden“ erleidet.

Sequenzdiagramm (platformereditor.Editor.panelClicked(MouseEvent))



Sequenzdiagramm (platformer.GameLoop.run())

