

|  |
| --- |
|  |
| **Eindopdracht New Media** |
| **2015/2016 - 2NMCT gr1**  **Robin Menschaert – Ziggy Maes** |
|  |
|  |

# Eindopdracht New Media: Onderzoeksdocument

## Introductie en concept

Robin en ik hebben één van de allereerste arcadegames uitgewerkt met een unieke twist: Pong dat werd uitgegeven door Atari in 1972. Hoewel het concept identiek is als het originele spel hebben we de spelbesturing compleet aangepakt. Pong werd klassiek gespeeld met twee toetsen (op en neer), of een hendeltje die een soortgelijke werking had. Daar we tegenwoordig iedereen thuis een generieke controller heeft liggen, zijde van een Playstation of Xbox, of Nitendo, hebben we besloten om hiervoor functionaliteit in te bouwen. Daarbovenop kan de andere speler het spel bedienen met een Leap Motion zodat geen fysiek contact met een controller nodig is. Onze uitwerking van Pong kan dus gespeeld worden door twee spelers: één speler die speelt door zijn hand op of neer te bewegen, de andere door met de generieke controller hetzelfde te doen.

Om het spel te balanceren hebben we de speler die de gamepad bedient ook de mogelijkheid gegeven om power-ups te verzamelen. Een dergelijke power-up kan er potentieel voor zorgen dat deze speler voor enkele seconden een spelvoordeel krijgt, of net niet. Zo kan de tegenstander trager bewegen, maar ook kleiner of groter worden.

## Analyse

### Hardware

Er worden twee stukken externe hardware gebruikt, één voor elke speler. Speler 1 krijgt een LeapMotion controller waarmee hij het spel moet bedienen, speler 2 krijgt een algemene controller (lees: gamepad, zoals bijvoorbeeld een xBox controller). Beide stukken hardware moeten rechtstreeks op éénzelfde computer worden aangesloten, elk op een unieke USB poort.

## Technische implementatie

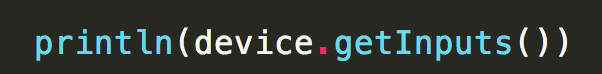
### Libraries

We kunnen de libraries, net zoals de hardware, in twee stukken verdelen: enerzijds hebben we de GameControlPlus library die ons toelaat om een generieke controller aan te sturen. Daarnaast maken we gebruik van de ‘onformative’ en ‘leapmotion’ libraries om onze Leap Motion aan te sturen.

We lichten enkel toe hoe we de generieke controller kunnen gaan aansturen daar de werking van de LeapMotion reeds in labolessen werd toegelicht.

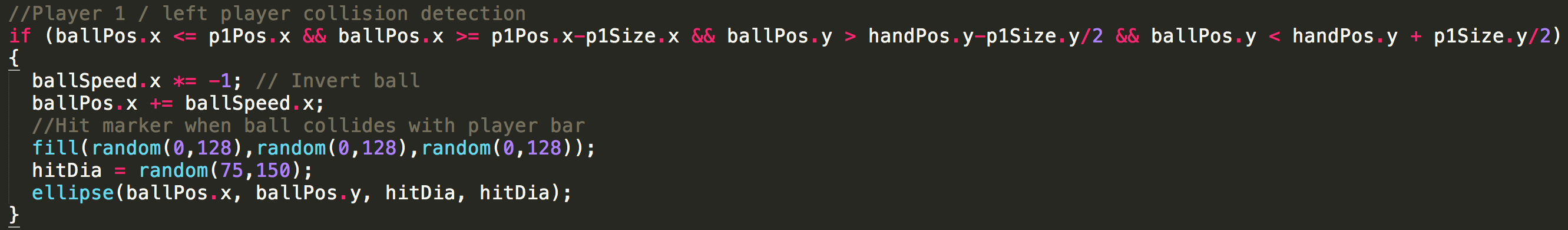
### Initialisatie van de controllers

De Leap Motion heeft geen verdere intialisatie nodig dan het connecteren van de USB-sensor. Het gamepad heeft echter het bijzonder vervelende probleem dat we een ID nodig hebben voor we dit kunnen gebruiken. Vreemde controllers zullen dus niet werken zolang hun ID niet aangepast wordt in de code. Dit kan op twee manieren: eenerzijds kan men de het unieke ID gaan traceren op de controller en deze string manueel ingeven. Anderzijds kan met een lijst generen met alle verbonden apparaten en hier het ID vinden dat gebruikt wordt in de controlIO.getDevice(#) functie. Dit kan door het volgende commando’s op te roepen:

### Botsingdetectie

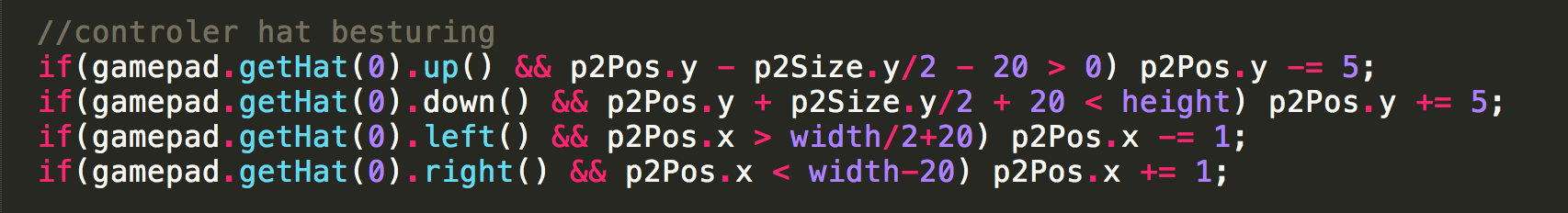
De bal wordt opgeslagen in een vector, en veranderd meerdere keren per seconden van positie (= snelheid). Een spelers hitbox bestaat uit twee aspecten: enerzijds de positie en anderzijds de grootte, deze worden ook ondergebracht in vectoren. We detecteren een ‘collision’ met de speler als de bal in de oppervlakte komt van de speler. Indien aan deze conditie, waarvan u hieronder een voorbeeld geïllustreerd ziet, voldaan wordt wordt de hoek waarin de bal zich beweegt geïnverteerd.



### Power-Ups

We hebben reeds 3 Power-Ups ingebouwd, maar deze functionaliteit kan makkelijk worden uitgebreid. Deze worden elke 20 seconden op een willekeurige plaats, en met een willekeurig resultaat, gegenereerd. Weliswaar komen deze enkel op het veld van speler 2 (de speler met het gamepad) voor. Speler 1 kan dus geen power-ups verzamelen, maar dit betekend niet dat deze speler enkel nadelen zal ondervinden. Een power-up kan ook voor speler 2 averechts werken.

Om de simpliciteit van power-ups op één verticale lijn te vermijden heeft speler 2 dan ook de mogelijkheid om dichter bij de tegenstander te komen en horizontaal te manouvreren. Dit kan het voor hem potentieel moeilijker maken om tijdig de bal te raken als deze terugkomt. Om het volledige potentieel van de controller te benutten is de keuze gemaakt om deze bewegingen te koppelen met de fysieke all-direction knoppen op het gamepad.



### Beweging

Beweging voor speler 1 is enkel verticaal. Deze speler kan zijn hitbox laten stijgen of dalen door zijn hand verticaal te bewegen boven de Leap Motion sensor. Hiervoor tekenen we bij elke draw cycle een rechthoek met een dynamische hoogte en vaste breedte van 10 pixels. Indien het hand hoger is dan de gemiddelde waarde zal dit rechthoek naar boven bewegen, anders naar beneden. De speler met de Leap Motion controller kan dus niet stil staan en moet steeds correcties blijven aanbrengen.

Speler 2 kan zoals eerder vermeld zowel horizontaal als verticaal bewegen, dit door middel van de ‘all-direction knobs’ op het gamepad. De rendering van zijn rechthoek is identiek als dat van speler 1, met de uitzondering dat speler 2 kan opteren om geen beweging uit te voeren.

### Score en weergave

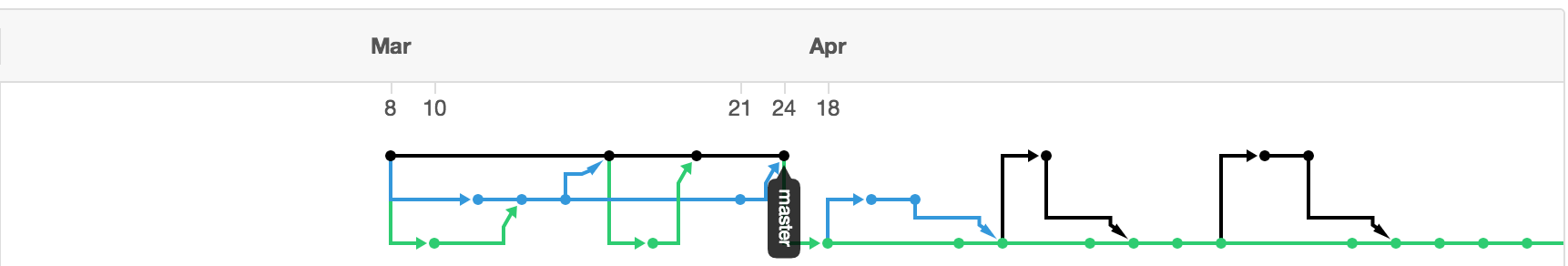
De score van een speler wordt met 1 punt vermeerderd indien deze erin slaagt om de bal buiten de speeloppervlakte van de tegenstander te krijgen. Het spreekt voor zich dat dit enkel lateraal is, indien de bal de boven- of onderkant van het spelbord raakt zal deze de andere richting uitkaatsen.

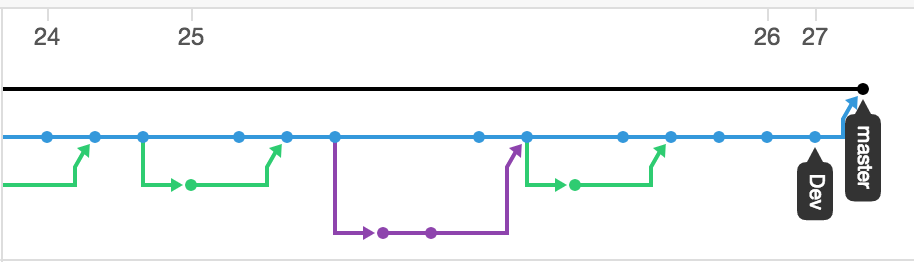
De score wordt bovenaan weergegeven. Het spel gaat oneindig lang door, en kan niet onderbroken worden.

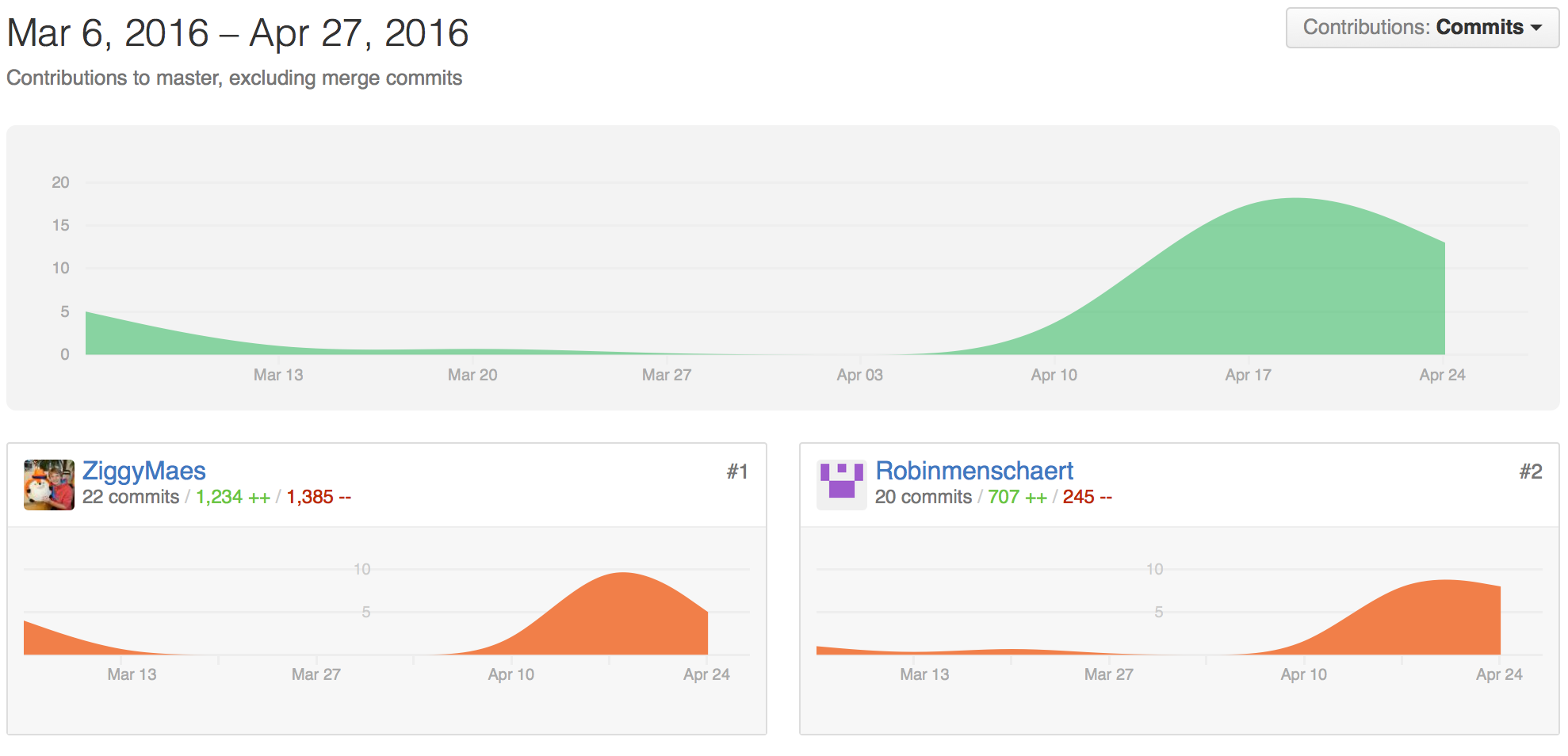
## Timemanagement en GIT

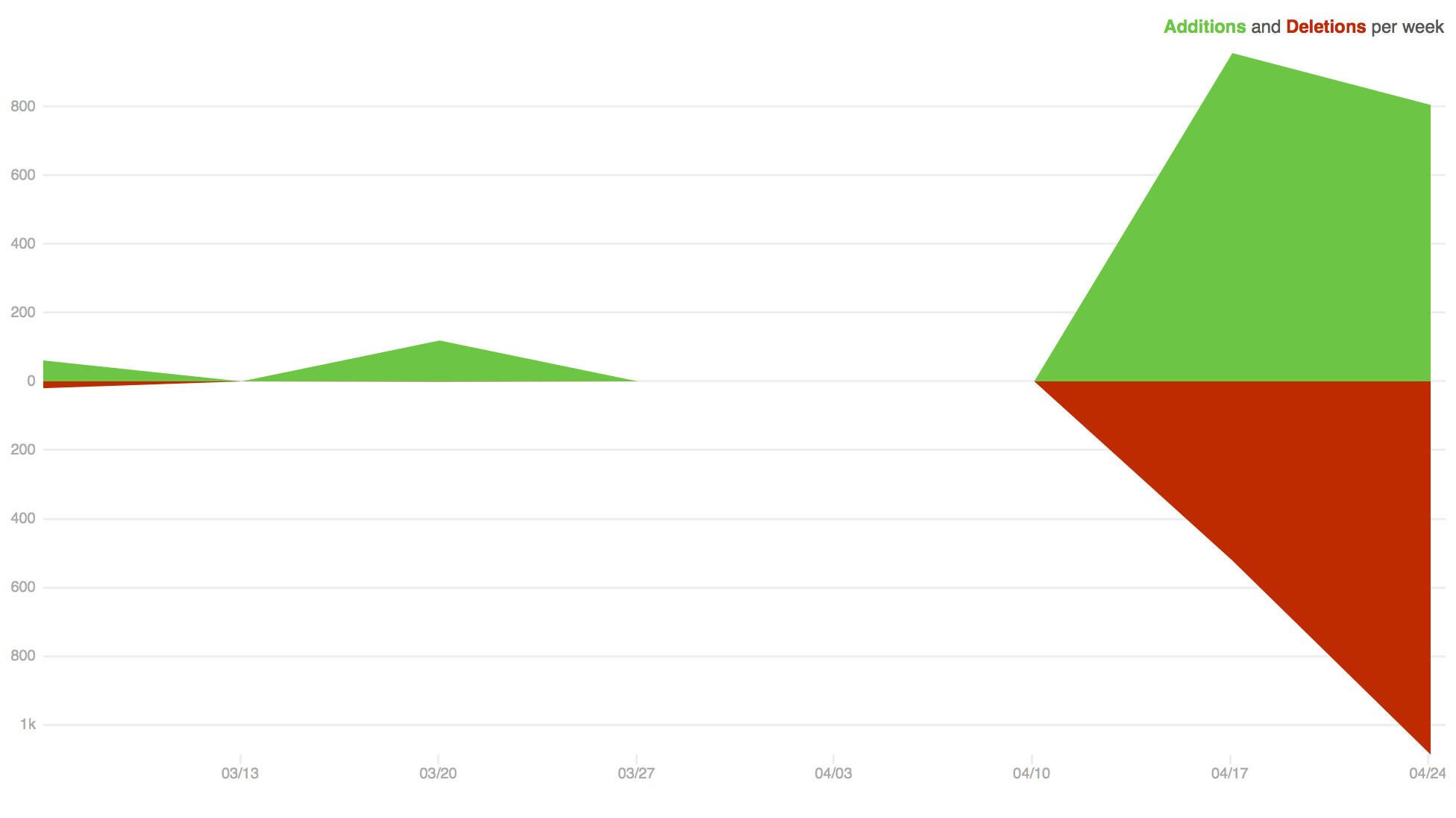
We hebben doorheen het ontwikkelingsproces gebruik gemaakt van GIT om onze code en documenten te versioneren. Door onze repository te koppelen aan Github kunnen hebben we enkele leuke grafieken die onze vooruitgang kunnen illustreren. Ook kunt u hierdoor een idee krijgen van hoeveel tijd we gespendeerd hebben om tot de huidige implementatie toe te komen.

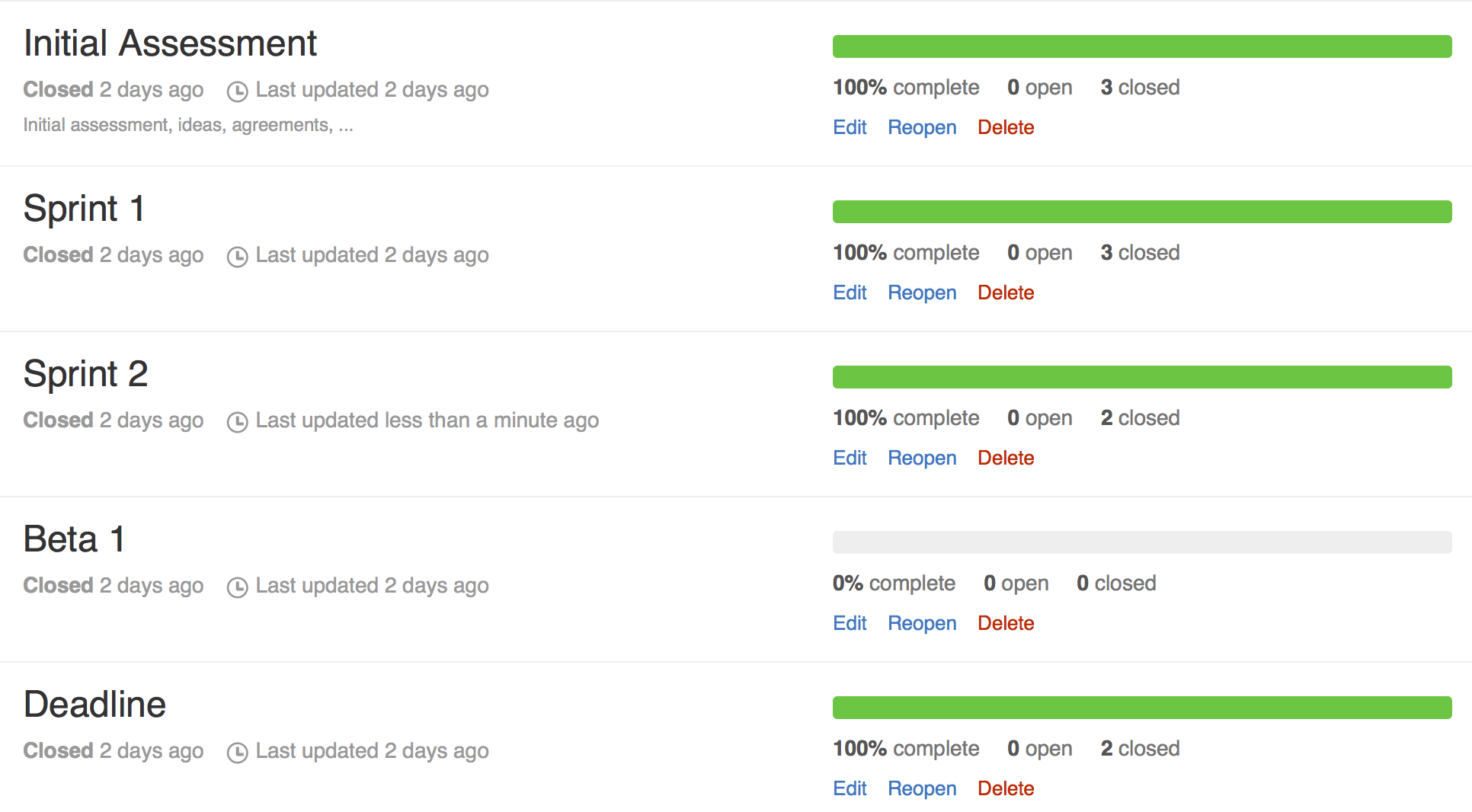
Een volledige lijst met taken vindt u in onze commit historiek terug: <https://github.com/ZiggyMaes/NMCT-New-Media-Eindopdracht/commits/master> .

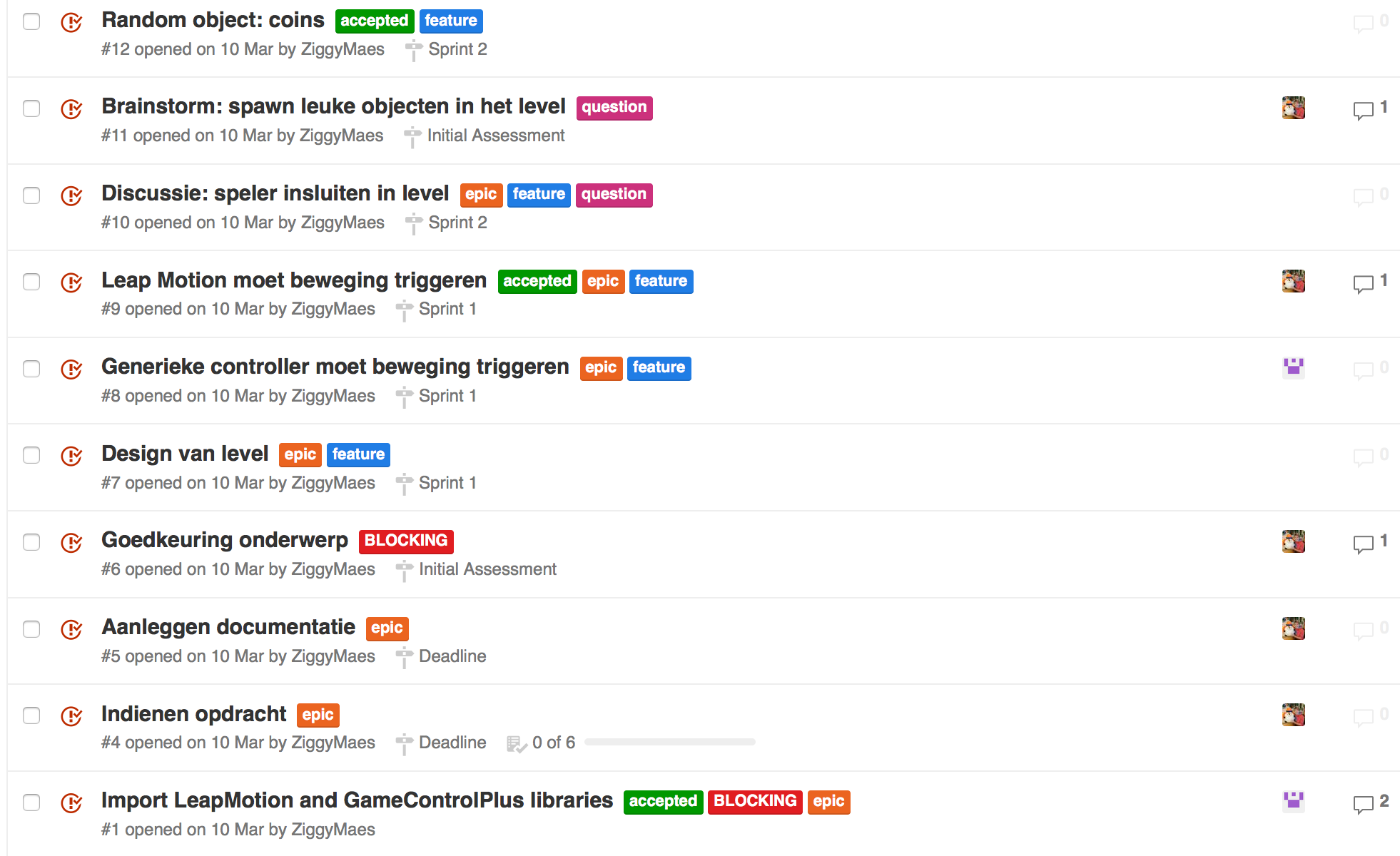


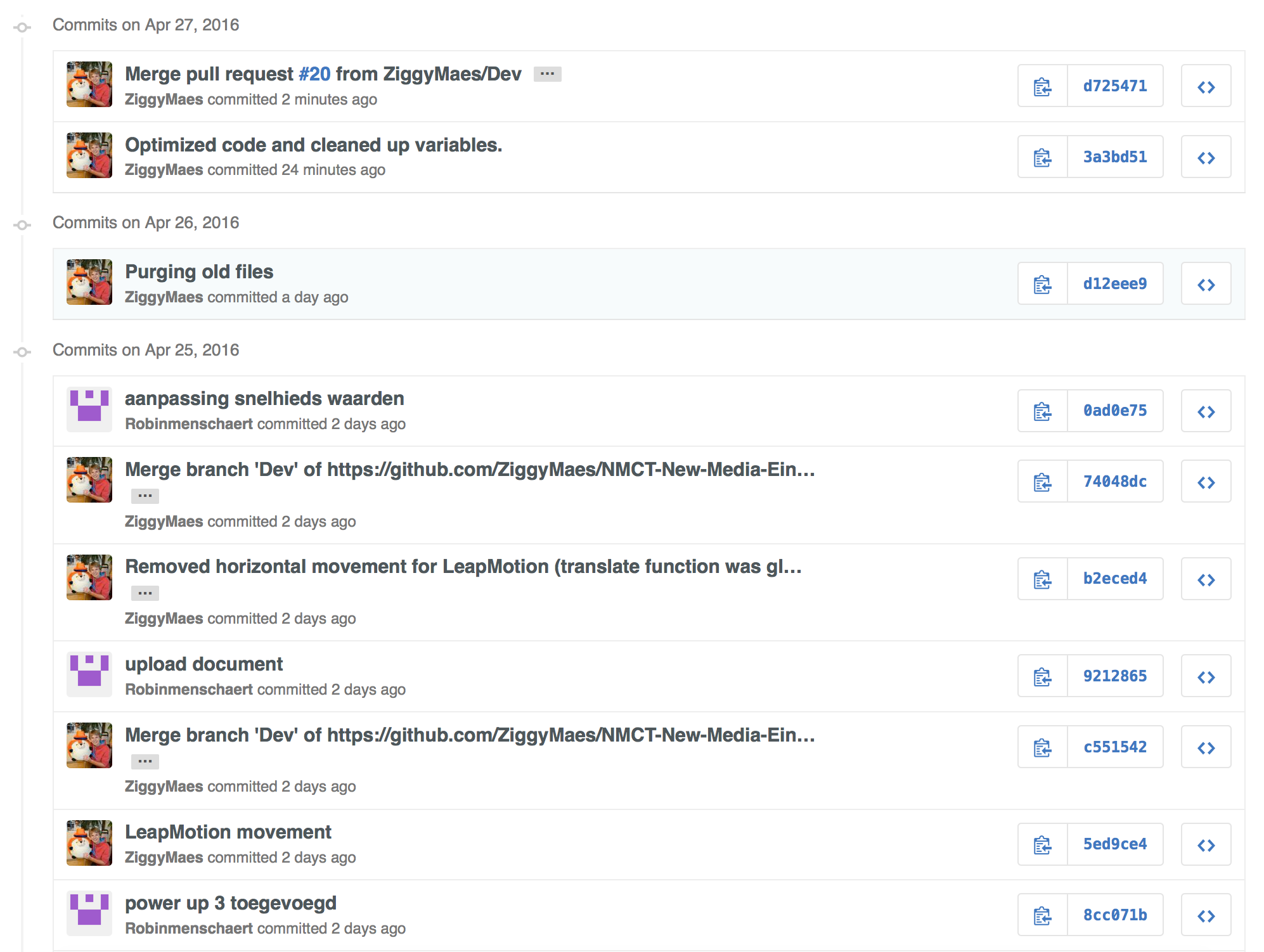












## Bronnen

Github - project repository: <https://github.com/ZiggyMaes/NMCT-New-Media-Eindopdracht>

Processing – reference guide: <https://processing.org/reference/>

Git – protocol documentation: <https://git-scm.com/doc>

## 