

Verslag Project Vleesjongen

Bert Livens - 20204681

27 augustus 2023

Wanneer het spel gestart wordt, worden dynamisch de levels en hun namen ingelezen en op het scherm getoond. Met behulp van de pijltjestoetsen kan de speler een level kiezen. Dan wordt dit ingeladen en begint het spel. Wanneer een level gedaan is, zou de speler moeten terugkeren naar het menu. Tijdens het spelen van een level zorgt de ESC-toets ook dat de speler terug in het menu terechtkomt, waar hij een ander level kan kiezen of hetzelfde kan verderzetten. Het venster kan in het menu of tijdens het spel van grootte veranderd worden, het level zal meeschalen.

Er is een klok-singleton en er is veel met inheritance en polymorfisme gewerkt, bijvoorbeeld bij events en objects/observers. Ook is het model, het view en de controller van de speler compleet gescheiden. Het model bevat eventuele informatie (zoals de snelheid in de twee richtingen), het view (het projectiescherm) zorgt enkel voor het tekenen op het scherm en de controller krijgt alle keyboard-events binnen en laat dit weten aan het model, deze vraagt aan de "Botseröf er muren geraakt worden en past indien nodig de snelheid aan.

Er zijn ook portalen die een speler omhoog kunnen transporteren maar pas op, de snelheid van een speler kan draaien door de orientatie van de portalen.

Het spel heeft een maximaal fps van 60, er wordt steeds gewacht met updaten tot er minstens een 60ste van een seconde (een "tik") is gepasseerd. Dit met dank aan het Meyer's singleton klok. Elke tik berekent de jongen waar hij terecht komt, handelt hij op basis van mogelijke botsingen en herberekent hij zijn snelheden, deze worden namelijk wat gedempt (behalve verticale snelheid naar beneden, die zal door de zwaartkracht meestal toebnemen). Dan wordt het hele scherm opnieuw getekend.

Alle logica die te maken heeft met input ontvangen, zit in de controller, alle logica die te maken heeft met op het scherm verschijnen, zit in het view. Zo is het eenvoudig componenten, zoals bijvoorbeeld SFML, te verwisselen. Dat is ook een reden om niet de klok van SFML te gebruiken maar een eigen

te ontwerpen.

Specifiek zijn de enige componenten die aangepast zouden moeten worden om een andere grafische library dan SFML te gebruiken: projectiescherm, screen, jongencontroller, game en sfmlevent (telkens .cpp en .h natuurlijk). Alle andere code is compleet grafisch agnostisch.

Een ander patroon dat gebruikt is, is dat van Observable (Object) en Observer. Als bijvoorbeeld de speler wint of sterft, wordt dat als een event gemeld door de botser, respectievelijk het projectiescherm en ontvangen *jongen* en *game* dit.