Space Invaders 2.0

Design

Observers

Omdat het mogelijk is dat objecten zowal willen andere objecten notify'en alsook zelf genotified willen worden zijn alle observables nu ook observers.

Notify neemt nu ook een type parameter. Dit zorgt ervoor dat het mogelijk is om slechts naar bepaalde types van notificaties te luisteren. Zo is het mogelijk om bijvoorbeeld iets te doen bij gameover.

Globale State

Er is nu een globale state. Deze houd bij in welke view we momenteel zitten. Enkele mogelijkheden zijn het menu, de game, scorebord, scoreinput, …

De global state word gemanipuleerd door de controllers. De voornaamste controllers die momenteel de state beheren zijn het menu en de scoreinput

Menu

Het menu is opgedeeld in een view en een controller. De view rendert de opties zodat deze kunnen worden weergegeven aan de gebruiker.
De controller is verantwoordelijk voor het luisteren naar userinput en daarbij de state te veranderen.

De menucontroller is vanaf nu ook het beginpunt voor de game. Er zijn 2 functies die de setup doen zoals het laden van het beginlevel en het aanmaken van de spelers en hun controllers.
In de vorige versie gebeurde dit in de main functie.

Magic numbers

Alle magic numbers worden geladen uit XML files. De xml files bevinden zich in the binary map, klaar voor packaging voor de eindgebruiker.

XML-Structuur

Levels

De level files bevinden zich in een levels map binnen de bin map.

De levels hebben een “Level” root element met attributen die voor het hele level bepaald zijn. Deze zijn de breedte en de hoogte van het level alsook welk level er op het huidige volgt.

Het root element heeft een enemies child dat alle enemies van het level definieert. De enemy children hebben parameters over hun initiële positie, schietsnelheid, thrustpower, vlieggedrag en hun kogels. De text inhoud van de enemy tag is hun texture.

Game en player

De game en player file staan rechtstreeks in de bin map.
De game file bevat globale game parameters zoals de lagg treshhold.
De player xml file bevat informatie over de player zoals zijn positie, textures, thrustpower en kogels.

Scores

De score file heeft het root element scores. De children zijn ongesorteerd en hebben score als tagnaam. Ze hebben een attribuut name die de naam van de gebruiker is en hun data waarde is een integer die hun score representeerd.

Scoreboard

Het scoreboard bevind zich bij de models. Het kan zichzelf laden en opslaan in een xmlfile. Tevens heeft het functionaliteit om het scoreboard te resetten of een score te adden.

Omdat scoreboard een model is dat door een view moet kunnen geobserveerd worden implementeert het de Observable base. Het heeft dus ook virtuele functies om de observer te notifyen en heeft de registerObserver hook.

Momenteel worden alle scores bijgehouden tenzij de gebruiker de scores manueel reset. Dit vormt echter geen probleem daar scores slechts een paar bytes innemen om op te slaan en er maar 1 score per beurt is. Het zou dus enorm lang duren om aan een te grote bestandsgrote te geraken.

Het sorteren van scores is representatie en gebeurt daarom in de scoreboardview.

Inputting scores

Wanneer een game eindigt stuurt deze een “gameover” event naar zijn listeners. De scorecontroller luistert hiernaar en zet vervolgens de globale state naar ‘scoreinput’. Vanaf dan begint de scorecontroller te luisteren naar keyinput en slaagt deze op.

Wanneer enter wordt ingedrukt wordt de score toegevoegd en gaat de state over naar ‘scores’ waardoor alle scores worden weergegeven.

Het renderen van de huidige input gebeurt in de scoreInputView.

Coop

Coop toevoegen was zeer eenvoudig. De keybindings zijn verhuist naar een file i.p.v. gehardcode te zijn. Daarna was het enkel nodig om bij de setup code 2 controllers aan te maken en 2 ships en deze aan de game toe te voegen.

Exceptions

De SpaceInvadersException heeft als base class de std::exception. Alle custom errors hebben een virtuele what functie. Deze geeft meer informatie over het probleem.

De errors worden voornamelijk gebruikt wanneer gamefiles niet kunnen worden ingelezen. Meer informatie word in dat geval geprint naar std::cerr waarna het programma met exit code 1 eindigt.