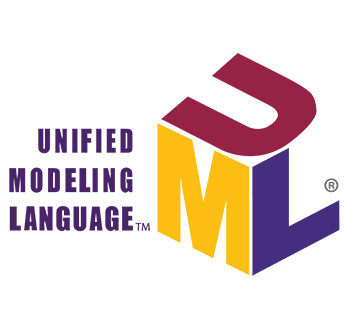
Ontwikkeldocument





Namen: Jasper Schinkel, Joost Simons en Daniel Zwamborn

Klas: SE 1D

Docenten: Cobie van der Hoek

Rob van der Krogt

Inleiding:

In dit document zal kort beschreven worden wat de eisen waren voor het te maken spel, hoe de opgestelde analyse en design klassendiagrammen eruit zien en waarom deze diagrammen ontworpen zijn. Tot slot komen de unittesten aan bod die gedaan zijn over de “Move” klasse.

Eisen:

Het speelveld is een vierkant vlak waarin een speler een weg moet vinden naar een eindveld. Een speler kan niet van het speelveld af.

De speler verplaatst zijn “poppetje” over het speelveld door middel van de pijltjestoetsen.

In dit speelveld zitten vaste muren en barricades. Een speler kan niet door een vaste muur heen.

Barricades kunnen geopend worden met een passende sleutel.

Een sleutel kan voor meerdere bijbehorende barricades gebruikt worden, maar slechts een sleutel kan in het bezit zijn van de speler.

Als een sleutel opgepakt wordt dan verdwijnt deze uit het doolhof. (Indien de speler al een eerdere sleutel in zijn/haar bezit had verdwijnt deze dus permanent)

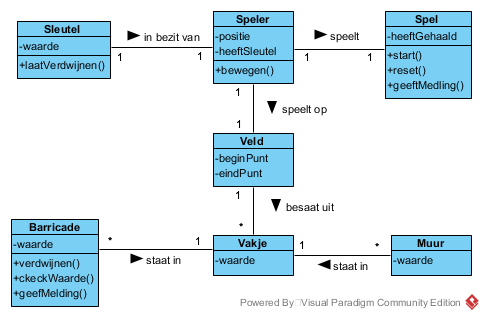
Als de speler een barricade probeert te openen met een sleutel die niet correspondeert met de barricade, dan krijgt de speler een foutmelding.

Correspondeert deze sleutel wel, dan verdwijnt de barricade.

Als de speler zijn “poppetje” op het eindveld weet te krijgen, dan verschijnt er een melding dat aangeeft dat de speler gewonnen heeft/het poppetje succesvol is begeleid naar het eindveld.

Het speelveld moet tijdens het spel opnieuw opgestart moeten kunnen worden, wat ervoor zorgt dat hetzelfde speelveld weer opgestart wordt.

Het spel moet geschikt worden gemaakt voor potentiele uitbreidingen.



Figuur 1. Analyse diagram voor het gemaakte sleutelspel

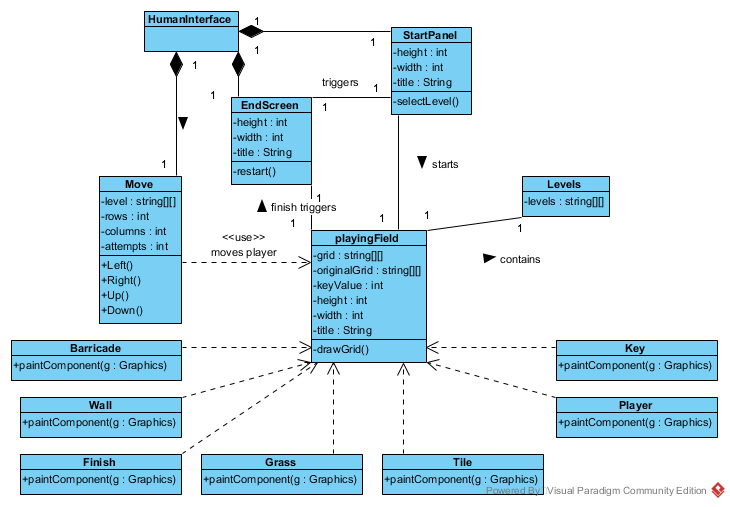
Toelichting:

In het ontwerp van de analyse klassendiagram is er in principe vanuit gegaan dat er maar een select aantal klassen gemaakt zouden worden. Een aantal van deze klassen zouden ook als objecten gedefinieerd worden, zoals de barricade, de sleutel, de muur en de speler. Deze objecten zouden dan hun eigen waardes krijgen om aan het speelveld aan te geven wat zij zijn.

De speler moest uiteraard kunnen lopen en een sleutel in zijn bezit kunnen krijgen. Voor de sleutels en barricades was het ook de bedoeling dat zij acties kregen. Sleutels moesten verdwijnen nadat zij opgepakt werden en barricades moesten ook verdwijnen nadat er een interactie plaatsvond tussen de speler en de barricade met de juiste sleutel.

Het speelveld zelf had nog een losse definitie wat simpelweg “Veld” genoemd werd. Hierin zou dan het speelveld gedefinieerd worden, met het beginpunt en het eindpunt voor de speler. In het uiteindelijke programma is dit veel uitgebreider geworden.

In het analyse diagram was het plan ook om alle knoppen in een klasse te stoppen, wat het spel zou worden, om zo het spel zo simpel mogelijk te houden. Ook was het de bedoeling dat de status van het spel bijgehouden zou worden in de klasse van het Spel, zodat het spel met deze informatie een melding kan geven om aan te geven dat het spel voorbij was.



Figuur 2. Design diagram voor het gemaakte sleutelspel

Toelichting:

Het uiteindelijke design diagram is heel anders geworden dan het analyse diagram. De klasses voor de objecten definiëren hier de graphics, dus hoe deze eruit zien, voor het spel. Deze worden nergens anders gedefinieerd dan in het playingField, dus deze zijn afhankelijk van deze klasse om te bestaan.

Het grid van het speelveld word gedefinieerd in playingField en gewijzigd in Move. Deze wijzigingen worden ook doorgegeven aan de klasse playingField, zodat de speler de wijzigingen kan waarnemen. De Levels worden geladen door het StartPanel, en de EndScreen gaat uiteindelijk terug naar het StartPanel.

In de klasse PlayingField word ook bijgehouden welke sleutel opgepakt is. Wanneer een andere sleutel opgepakt word, dan overschrijft deze de vorige sleutel.

Menselijk input kan de Move klasse beïnvloeden en de Move klasse beweegt de Speler op het Speelveld. Ook kiest de mens het level in het StartPanel en kan de mens terug gaan naar het StartPanel via de klasse van de EndScreen.

Dit diagram laat wel erg zien dat de klasses van PlayingField en Move veel attributen waarborgt. Voor een schonere design diagram had dit wellicht beter verspreid kunnen worden over meerderen klassen.

Unittesten: