

1 Diepgaande Beschrijving van het project

1.1 Diepgaande uitleg

In deze sectie wordt op hoog niveau de IDE beschreven. Dit als korte inleiding voordat de algoritmes en datastructuren vermeld worden. Hierdoor kan verschillende terminologie verduidelijkt worden.

Zoals al eerder vermeldt kan een gebruiker de programma flow kunnen opbouwen in het deel van de IDE dat we het Frame-view noemen. Hierin kunnen instanties van klassen die eerder gedefinieerd werden de mogelijkheid worden geboden om informatie aan elkaar door te geven. Deze informatie noemen we een Event. Wanneer een instantie een Event verstuurd en wat hij met een ontvangen Event doet is beschreven in de klasse waartoe hij behoort.

In de volgende paragrafen zal een diepgaande uitleg worden geven over wat een Events is en hoe de gebruiker er gebruik van maakt alsook hoe hij een klasse kan definiëren en welke standaard eigenschappen een klasse in de IDE bevat.

1.1.1 Events

Om onderlingen communicatie tussen Instanties van Klassen voor te stellen, gebruiken we een **Event**. Een Event kan al dan niet informatie bevatten. Een Event zonder informatie kan beschouwd worden als een trigger. De informatie dat een Event kan bevatten kan uit meerdere delen bestaan. De informatie kan dus bestaan uit meerdere primitieve types (int, string of boolean). Elk deeltje in die informatie noemen we een variable. Met elke variable wordt een naam geassocieerd. Deze kan de gebruiker dan gebruiken om de variabele uit een Event op te vragen. Ook het Event zelf moet een ID hebben dat als type geldt.

Eens de gebruiker een Event heeft gedefinieerd in de daarvoor voorziene omgeving kan hij er verder in de IDE gebruik van maken. Dit doet hij dan door deze Event te selecteren in een dropdown menu van bepaalde blokken, dit maakt een EventInstance aan. Hij kan dus vanaf dan een Event het eerder gedefinieerde type aanmaken en invullen met de informatie die hij wenst mee te geven. Het zenden van een EventInstance door een klasse noemt een emit. Naar welke instanties van klassen het EventInstance wordt verzonden, kan worden bepaald in het Frame-view van de IDE.

Er is een aparte view waarin alle Instanties van Klassen als blokjes getoond worden, dit view noemen we het **Frame-view**. Deze blokjes bevatten inkomende en uitgaande poorten. Deze stellen respectievelijk de evenementen voor die een Instantie wil ontvangen en de evenementen die het uitzendt. Er kunnen verbindingen gemaakt worden tussen de uitgaande poorten van een instantie en de inkomende poorten van een andere instantie, waarbij de respectievelijke

evenementen hetzelfde type hebben.

Dit aparte view is echter de begin positie van alle gewenste Instanties van de aangemaakte Klassen. De gebruiker heeft de optie om de verbindingen al dan niet te tonen. Een extra view, het **Canvas-view** het bewegen van de instanties toont. De gebruiker kan zo de flow van Events bekijken.

Nieuwe **Events kunnen aangemaakt worden** door de gebruiker in een aparte sectie van de IDE. Een Event moet een type hebben, vervolgens kan er informatie meegegeven worden aan dit Event. Deze informatie is een POD (plain old data) die opgebouwd wordt door de gebruiker. Hierin zal elke variable een unieke naam en specifiek type hebben.

Er zijn **standaard Events** beschikbaar zoals oa. KeyA, MousePress, Start, enz. Deze events zijn voorgedefinieerd en dienen om interactie te hebben met het visuele canvas.

1.1.2 Klassen

Een Klasse kan worden vergeleken met eenderzijds een Sprite in de visuele programmeeromgeving Scratch [?] en anderzijds een klasse uit een object geïoriënteerde taal zoals Java. Het verschil in deze applicatie is dat de Instanties van een Klasse expliciet aangemaakt worden in de Frame-view. Een Klasse bestaat uit: input Events, Handlers voor die Events, functie definities en member variabelen. Een Klasse kan worden voorgesteld in het Frame-view. Deze appearance kan door een functie in de Klasse worden veranderd. Een Instantie kan dan in het Frame-view beslissen van welke andere Instanties het die input Events ontvangt of naar welke instaties hij Events verstuurd.

Een Klasse kan **Events ontvangen**. Dit werd besproken 1.1.1. Het afhandelen van een Event gebeurt door een handler die het event binnen krijgt. Het raadplegen van de inhoud van een Event, kan doormiddel van een accessblok. Deze kan een variable accessen die in het binnenkomende event zit.

Een Klasse kan **Events emitten**. Dit kan met behulp van een Emit blok. Hierin moet een Event worden geselecteerd. Als een Event informatie bevat zal deze ook hier moeten worden ingevuld.

Een Klasse zal ook een overzicht hebben met alle Events die erdoor worden geëmit, dit is getoond in het Wire-view, indien een instance bestaat van die Klasse.

Een uitbreiding van de visuele omgeving ten opzichte van andere IDE's is toe te laten om **functie aanroepen** te maken binnen een Klasse. Eerst was het idee om dit voor te stellen met een lijn die twee functieblokken zou verbinden. Bij een Klasse met veel interne functie aanroepen wordt dit echter onoverzichtelijk.

Een functie oproep vanuit een andere functie zal worden voorgesteld door blokje. Dit blokje bevat de naam van de functie die zal worden opgeroepen. Alsook zijn input parameters en return waarde. De parameters worden by-value doorgegeven aan de functie. Hierin kunnen variable gebruikt worden die als constante worden doorgegeven. Een variable is data dat een bepaald type heeft zoals een number of string. Er kan geschreven worden naar een variable en de variable kan gelezen worden. De onderkant van een functieaanroep blok bevat een leeg vakje voor de return waarde. Hier kan een variabele aan gekoppeld worden om deze waarde op te vangen.

Member Variabelen zijn variabele die gelden per Instantie van een Klasse. Deze kunnen bijvoorbeeld de toestand van van de Instantie van een Klasse die een lamp voorstelt in het canvas voorstellen.

1.1.3 Blokken

Een blok is een blokje dat de gebruiker kan plaatsen in het programmeer venster van de IDE. Deze blokken kunnen alles voorstellen, bv. variabelen, types, control-flow, functions, enz. . Deze zijn onderverdeeld in verschillende categorieën. De verdere uitleg met betrekking tot deze categorieën en de blokken die erbij horen staan in bijlage op sectie ??.