

Gebruiksaanwijzing Active Writer- programmeertaal

Versie 1

Pieter Gijpen

1. Scriptformattering

De commando's worden herkend door de tekens `<` en `>`. Tussen deze tekens wordt de instructie getypt. De argumenten worden als

De basisformattering van een Active writer-commando is als volgt.

```
<(instructie):(argument1):(argument2):(argument3A$argument3b)>
```

2. Instructies

Het is belangrijk om de commando's ZONDER hoofdletters te typen. Anders worden deze niet herkend.

2.1. 'music'

Voor de instructie 'music' heb je twee argumenten nodig: de naam van het audiofragment en 'true', indien het lied zich moet herhalen of 'false' als het lied maar één keer moet afspelen. Het audiofragment moet een MP3-fragment zijn (.mp3). De audiofragmenten moeten gekopieerd worden in de map 'music'.

Syntax-voorbeeld:

```
<music:(naam van het audiofragment):('true' of 'false')>
```

Praktisch voorbeeld met gebruik van een audiofragment 'abba.mp3':

```
<music:abba:true>
```

Als de muziek moet stoppen, dat typ je:

```
<music:none:true>
```

2.2. 'sfx'

Voor de instructie 'sfx' heb je maar één argument nodig: de naam van het audiofragment. Het audiofragment moet een WAV-fragment zijn (.wav). De audiofragmenten moeten gekopieerd worden in de map 'sfx'.

Syntax-voorbeeld:

```
<sfx:(naam van het audiofragment)>
```

Praktisch voorbeeld met gebruik van een audiofragment 'sfx.wav':

```
<sfx:sfx>
```

Als alle geluidseffecten moeten stoppen, typ dan:

```
<sfx:none>
```

2.3. 'image'

Deze instructie zorgt ervoor dat een afbeelding verschijnt op de voorgrond of achtergrond. Het formaat maakt hier niet uit.

Syntax-voorbeeld:

```
<image:(naam van de afbeelding):('bkg' voor achtergrond, 'fog' voor voorgrond):(fade-in- en out-tijdsduur in miliseconden)>
```

Afbeelding 'background.jpg' op de achtergrond weergeven:

```
<image:background.jpg:bkg:10>
```

Afbeelding 'foreground.png' op de voorgrond weergeven:

```
<image:foreground.jpg:fog:10>
```

Afbeelding op de achtergrond verwijderen:

```
<image:nobkg:bkg:1000>
```

Afbeelding op de voorgrond verwijderen:

```
<image:nofog:fog:1000>
```

Alle afbeeldingen moeten in de map 'textures' geplaatst worden.

2.4. 'video'

De instructie 'video' zorgt ervoor dat er een videofragment wordt weergegeven. Deze overlapt alleen de achtergrond. Het videoformaat maakt niet uit.

Syntax-voorbeeld:

```
<video:(naam van het videofragment):( fade-in- en out-tijdsduur in miliseconden)>
```

Videofragment 'video.mp4' weergeven:

```
<video:video.mp4:20>
```

Videofragment stoppen:

```
<video:none:20>
```

Alle videofragmenten moeten in de map 'videos' geplaatst worden.

2.5. 'title'

Instructie 'title' geeft een titel weer in het midden van het scherm.

Syntax-voorbeeld:

```
<title:(titeltekst):(fade-in- en out-tijdsduur in miliseconden)>
```

Weergave text 'Intro' als titel:

```
<title:intro:15>
```

2.6. 'voice'

Voice stelt een stem in wanneer er tekst getypt wordt. Voorlopig zijn er maar twee opties: pieter en buna.

Voorbeeld:

```
<voice:pieter>
```

2.7. 'txtcolor'

Deze instructie verandert de tekstkleur. De tekstkleur moet voluit worden geschreven in het Engels.

Syntax-voorbeeld:

```
<txtcolor:(kleur in engels)>
```

Voorbeeld witte tekst:

```
<txtcolor:white>
```

2.8. Geavanceerde commando's

2.8.1. 'options'

Met options kun je de lezer dingen vragen met telkens een Ja- en nee-optie. Je kunt zelf de vragen en antwoorden kiezen, alsook wat er gebeurt indien ja of nee.

Syntax-voorbeeld:

```
<options:(vraag):(antwoord ja):(antwoord nee):(callback indien  
ja):(callback indien nee):(argumenten voor ja-callback):(antwoorden voor  
nee-callback)>
```

2.8.1.1. Callback-instructies

2.8.1.1.1. continue

Met de callback-instructie 'continue' gaat alles normaal verder. De extra argumenten zijn altijd null en null.

Voorbeeld:

```
<options:Is alles oké?:Ja, het gaat wel.:Neen, niet  
echt...:continue:stop:null>null:buna$Dan moeten we afronden...$10>
```

2.8.1.1.2. stop

De stop-callback beëindigt het script na een boodschap. Deze werkt met dezelfde argumenten als de 'text'- 'voice'-instructie.

Voorbeeld:

```
<options:Is alles oké?:Ja, het gaat wel.:Neen, niet  
echt...:continue:stop:null>null:buna$Dan moeten we afronden...$10>
```

2.8.1.1.3. skip

Deze callback-instructie slaat regels over.

Voorbeeld, ik wil 5 regels verder:

```
<options:Is alles oké?:Ja, het gaat wel.:Neen, niet  
echt...:continue:skip:null$null:buna$Dan moeten we afronden...$10$5>
```

2.8.1.1.4. confirm

Confirm doet het tegenovergestelde van skip. Hiermee ga je een gewenst aantal regels terug. Alleen wordt er geen bericht getypt.

Voorbeeld 3 regels terug:

```
<options:Is alles oké?:Ja, het gaat wel.:Neen, niet  
echt...:continue:confirm:null$null:3$null>
```

2.8.2. 'input'

De input-instructie vraagt de lezer om tekst in te voeren. Deze kan je ergens anders laten terugkomen in je verhaal. Dit werkt met een verwijswoord naar de tekst die de lezer opgegeven heeft.

Syntaxis:

```
<input:(verwijswoord):(vraag naar invoer)>
```

Voorbeeld:

```
<input:name:What's your name?>
```

Daarna kun je dat antwoord in je verhaal verwerken door:

```
<text:Hello, €name€!:20>
```

3. Inhoud verhaal (platte tekst)

De tekst die weergegeven wordt, moet niet getypt worden als een commando. M.a.w. Je typt de inhoud van je verhaal gewoon tussen de commando's in.