

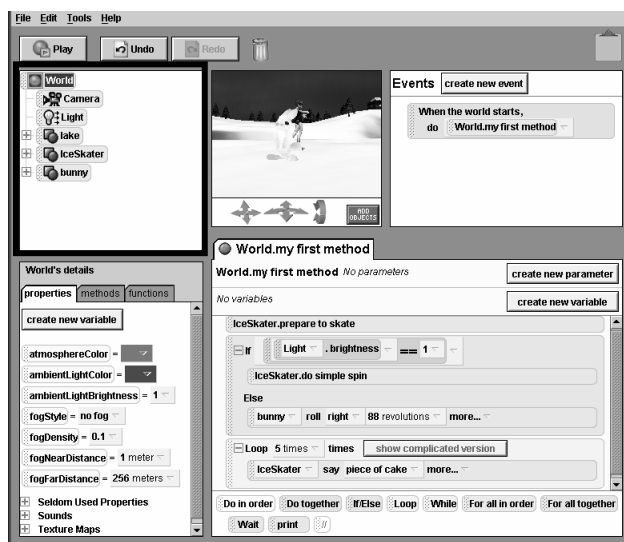
Les 5: De structuur van condities in controlestructuren in Alice



Wouter Tavernier
18/12/2006

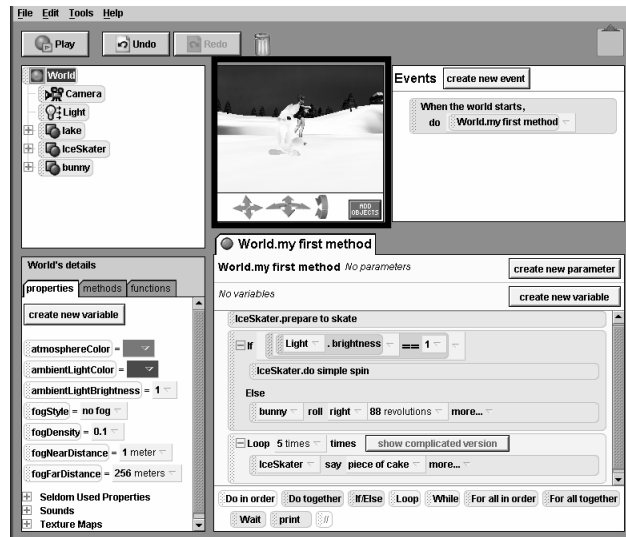
Oprisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



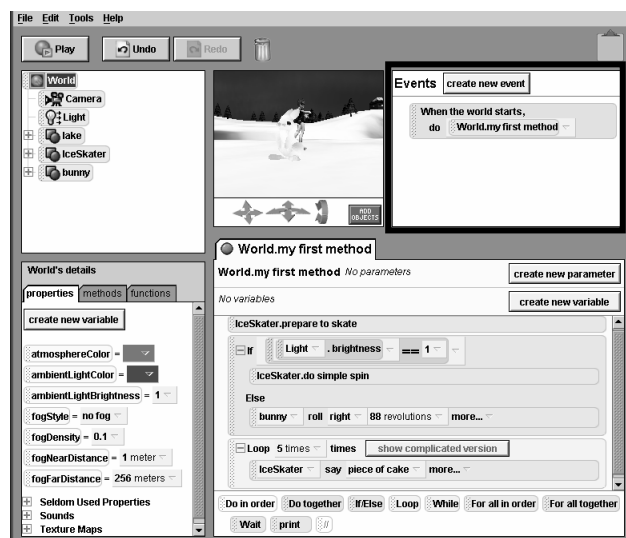
Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



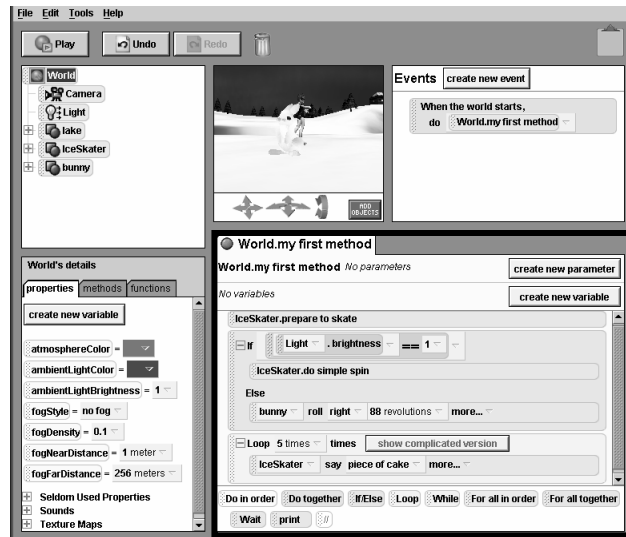
Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



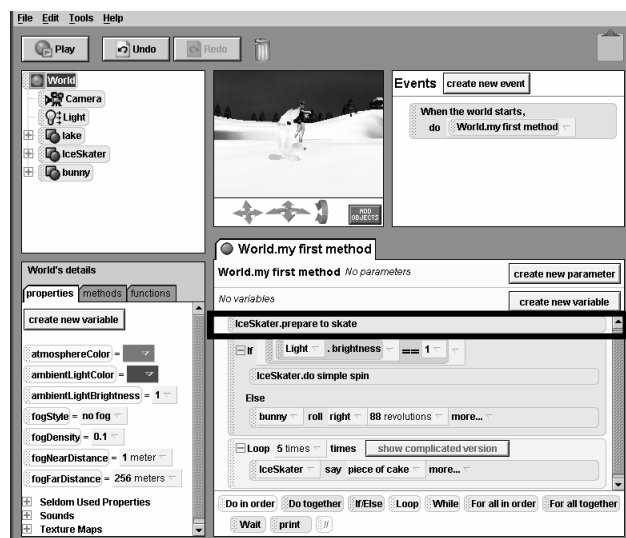
Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



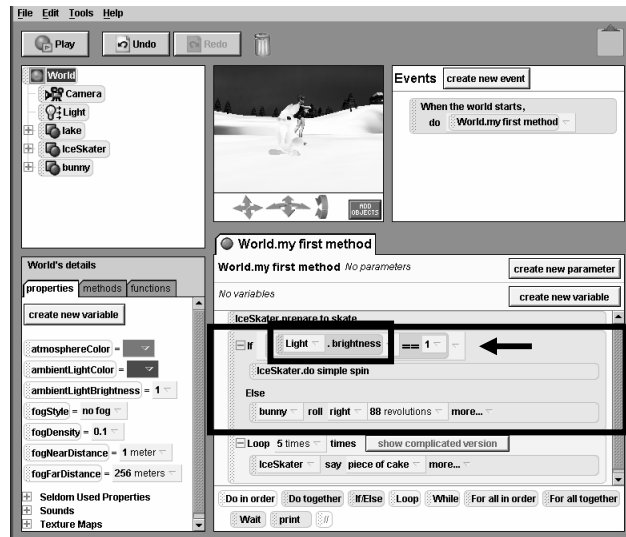
Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



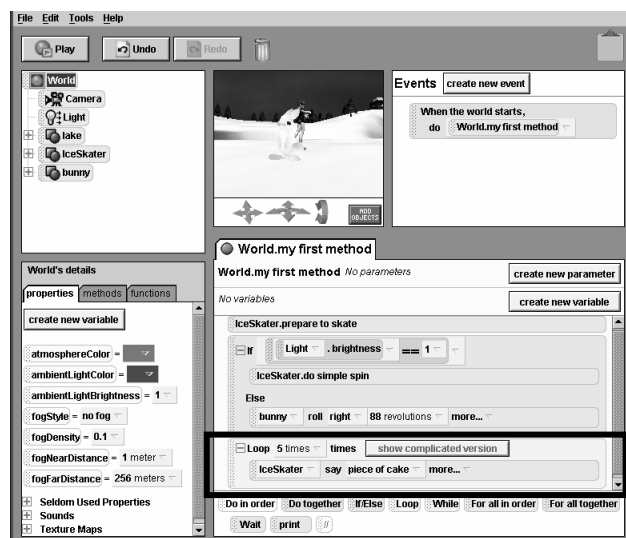
Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



Opfrisoefening

- Geef de betekenis & functie van het onderdeel waarnaar het pijltje/kader verwijst
- Te beginnen met ...



Herhaling – Zet in twee kolommen

While

Conditionele herhaling

Blijf lopen tot het regent

Loop tot je 3 km hebt gelopen

Loop

Begrensdde herhaling

Ik weet een voorwaarde om de herhaling te doen stoppen

teller

Ik weet in het begin precies hoeveel stappen er uitgevoerd moeten worden vooralleer de herhaling moet stoppen.

voorwaarde

Nieuwe materie

Operatoren en functies

Ingewikkelder condities

Events

Oefening – operatoren

1. Open de wereld “IceSkaterWorld.a2w”
2. Vergelijk in het detailvenster de “functions” van de objecten
 - Lake
 - World
 - IceSkater
3. Welke verschillen merk je?

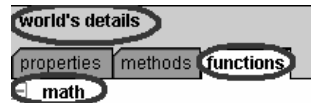
Operatoren – Wat zijn dat?

- Operatoren zijn een soort gereedschap waarmee je objecten met elkaar in verband kunt brengen
- Omdat die operatoren niet specifiek zijn aan het object zelf maar meer gerelateerd aan het kader waarin ze gebruikt worden, staan ze onder “functions” bij World.
- VOORBEELDEN:
 - Relationele operatoren
 - Hiermee kunnen we verschillende objecten vergelijken (vandaar “relationeel”) in de opbouw van condities
 - Boolse operatoren
 - Hiermee kunnen we condities over verschillende objecten combineren

Relationele operatoren

- Vorm-eisen:

- 2 objecten van het zelfde type
- “geen appels vergelijken met peren”



- Gebruik:

- In de conditie van de “if/else” of “while”
- Door de operator te slepen binnen het conditie-gedeelte

<code>a == b</code>	<i>is equal to</i>
<code>a != b</code>	<i>is not equal to</i>
<code>a > b</code>	<i>is greater than</i>
<code>a >= b</code>	<i>is greater than or equal to</i>
<code>a < b</code>	<i>is less than</i>
<code>a <= b</code>	<i>is less than or equal to</i>

Oefening

- Maak een nieuwe wereld
- Voeg een kip en een koe toe als objecten
- Vergelijk de hoogte van de koe en de kip adhv relationele operatoren en laat de grootste zeggen “ik ben de grootste”
- Extra oefening:
 - Bij het tabblad “functions” van world vind je ook de functie “random number”
 - Gebruik deze functie om de grootte van beide objecten willekeurig te veranderen voordat bovenstaande test wordt uitgevoerd.
 - Hoe verander je de onder- en bovengrens van het random number?
 - Hoe verander je de grootte?

STORYBOARD:

- .
- .
- .

Boolse operatoren

- Logische (voor-)waarden combineren kunnen we adhv EN, OF en NIET
- Voorbeelden:
 - Ik krijg een playstation 3 als
 - ik een goed rapport heb
 - EN
 - ik thuis help bij het afwassen
 - Ik krijg een pak rammel als
 - Ik NIET geslaagd ben
 - OF
 - wanneer ik teveel kabaal maak

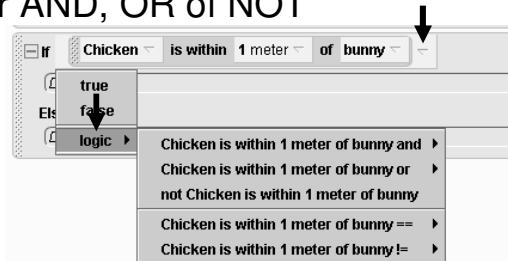
Boolse operatoren

- Neem de conditie A en B die WAAR of VALS kunnen zijn
- Dan kunnen we de volgende “waarheidstabel” opstellen:

<i>conditie A</i>	<i>conditie B</i>	<i>A EN B</i>	<i>A OF B</i>	<i>NIET A</i>
WAAR	WAAR			
VALS	WAAR			
WAAR	VALS			
VALS	VALS			

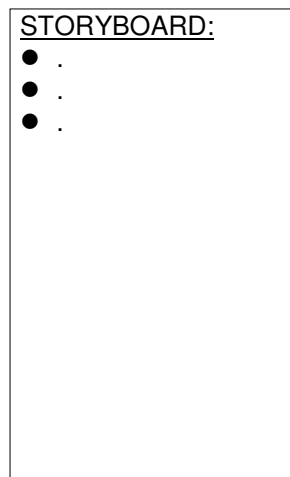
Boolse operatoren gebruiken in Alice

- Voor een enkelvoudige voorwaarde in de conditie-positie van de if of while-controlesruktuur
- Klik op het pijltje rechts
- Klik op “logic”
- Kies voor AND, OR of NOT



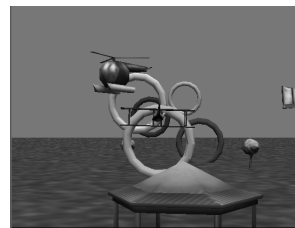
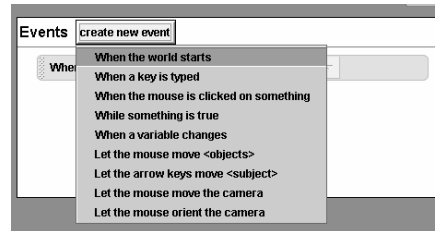
Oefening

- Pas de vorige oefening zo aan, zodat ook getest wordt of het ene dier breder (width) is dan het andere, en dan pas roept “ik ben de grootste”



Events

- Event = (systeem-) gebeurtenis
- Voorbeelden in Alice:
- Nut:
 - Wanneer die gebeurtenis zich voor doet, kun je het programma bepaalde acties laten ondernemen = INTERACTIVITEIT
 - Bijvoorbeeld de pijltjes bij de besturing van spelletjes



Events

- Events laten toe dat de gebruiker bepaalt hoe het programma verder loopt
- De code die je schrijft die moet worden uitgevoerd wanneer een event zich heeft voorgedaan = **event handler**
- Voorbeeld-storyboards:

Event: Spacebar press

Response:

Do together

roll biplane a full revolution
play biplane engine sound

Event:: Up Arrow key press

Response:

Do together

move biplane forward
play biplane engine sound

Oefening

- Schrijf 2 event handlers in het programma met de koe en de kip zodat je met de pijltjes de kip groter en kleiner kunt maken
- Herschrijf het oorspronkelijke programma op die manier zodat de test (wie is de grootste?) continue wordt uitgevoerd zodat meteen na het vergroten van de kip de juiste persoon zegt “ik ben de grootste”

STORYBOARD:

- .
- .
- .

Afsluitende oefening

- Het konijn en de danseres zijn beiden erg “in the mood”. Dit tonen ze door terzelfdertijd eerst (elk voor zich) “yeeha” te zeggen en daarna een spin te doen. Het konijn doet dit sowieso 15 keer en loopt dan naar de schaatster, terwijl de danseres dit blijft doen tot het donker wordt en het konijn naar haar toegelopen is.
- Maak een programma waarin dit gebeurt, en laat het donker worden door op het pijltje naar boven te drukken.

STORYBOARD:

- .
- .
- .