Kennis-check Blok 1

Kennis-check Blok 1

datasource

Om je voor te bereiden op de multiple choice kennis-check kun je voor je zelf proberen de ondersrtaand evragen te beantwoorden.

Als je op de vraag klikt dan verschijnt het antwoord. Op die manier kun je zelf jou antwoord controleren.
Begrip je de uitleg niet, vraag dan aan de docent om extra uitleg.

Scratch 1

Wat is een lus in programmeren en waarom is het handig?

Een lus (of "loop") is een blokje in Scratch dat zorgt dat een stukje code steeds opnieuw wordt uitgevoerd. Bijvoorbeeld: als je wilt dat een sprite (Giga) blijft bewegen zolang het spel bezig is, kun je een "herhaal" of "herhaal zolang" blok gebruiken. Dit is handig, omdat je dan niet elke stap apart hoeft te programmeren – het gebeurt automatisch steeds opnieuw.

Wat is een als-dan-anders blok (if-then-else)?

Een als-dan-anders-blok in Scratch controleert of iets waar is. Als dat zo is, doet het programma het eerste stukje code. Anders doet het iets anders. Bijvoorbeeld: *als* de muis is ingedrukt, *dan* laat een sprite (Giga) iets zeggen, *anders* doet hij niets of zegt hij iets anders. Zo kan je programma reageren op wat er gebeurt.

Wat is een variabele en hoe kun je die gebruiken om de score bij te houden?

Een variabele is een blokje waarmee je informatie kunt opslaan, zoals een getal dat kan veranderen. In een spel kun je een variabele maken die "score" heet. Aan het begin zet je die op 0. Elke keer dat Giga bijvoorbeeld een sleutel aanraakt, verhoog je de score met 1. Zo weet het spel hoeveel punten je hebt.

Wat is het verschil tussen gewoon “als…dan” en “als…dan…anders”?

“Als…dan” voert alleen code uit *als* de voorwaarde waar is. Als het niet waar is, gebeurt er helemaal niks. **"Als…dan…anders”** voert óf de ene actie uit als de voorwaarde waar is, óf een andere actie als de voorwaarde niet waar is. Dus:

- Met “als…dan” doe je iets óf niks.
- Met “als…dan…anders” doe je altijd iets: óf de ene actie óf de andere.

Scratch 2

Wat betekent het als een sprite op x: 0 en y: 0 staat?

Dat betekent dat de sprite precies in het **midden van het speelveld** staat.
In Scratch is (0, 0) het **middelpunt van het scherm**.

- De x-coördinaat (horizontaal) is dan in het midden van links naar rechts.
- De y-coördinaat (verticaal) is in het midden van boven naar beneden.

In welke richting beweegt een sprite als je de x-coördinaat groter maakt?

Dan beweegt de sprite **naar rechts**.

- Hoe groter de x-waarde, hoe verder naar rechts de sprite op het scherm staat.
- Als je de x-coördinaat juist kleiner maakt, beweegt de sprite naar links.

Wat gebeurt er als je bij x een heel grote waarde gebruikt, zoals 5000?

De sprite verdwijnt **buiten beeld**.

- Het Scratch-scherm heeft grenzen: ongeveer van x = -240 (helemaal links) tot x = 240 (helemaal rechts).
- Als je een waarde gebruikt zoals x = 5000, dan staat de sprite ver buiten het zichtbare scherm, dus je ziet hem niet meer.

Hoe kun je zorgen dat een sprite niet buiten het scherm beweegt?

Gebruik een **“als…dan”**-blok om te controleren of x of y binnen een bepaalde grens.

```
als x-positie < 240 dan
   wijzig x met 10
```

Van Scratch naar Python

Waarom is inspringen (indentatie) belangrijk in Python, terwijl Scratch dat niet nodig heeft?

Omdat Python aan de hand van inspringen (spaties) bepaalt welke code bij elkaar hoort.
Als je dat niet goed doet, begrijpt Python niet wat je bedoelt, en krijg je een foutmelding.

□□ *Voorbeeld:*

```
if x > 10:
print("x is groot")  # □ fout: geen inspringing
```

```
if x > 10:
    print("x is groot")  # □ goed: ingesprongen
```

Leg uit wat een `#` in Python doet

Een `#` wordt gebruikt om commentaar toe te voegen in je code. Alles wat na het `#` staat, wordt genegeerd door de computer. Het is alleen bedoeld voor de programmeur zelf, om uit te leggen wat de code doet.

□□ *Voorbeeld:*

```
# Dit is een commentaarregel
x = 5  # We geven x de waarde 5
```

Hoe ziet een if-then-else er uit in Python?

In Python gebruik je `if`, `else` (en soms `elif`). De code die bij elke voorwaarde hoort moet ingesprongen staan.

Voorbeeld:

```
if x > 10:
    print("x is groter dan 10")
else:
    print("x is 10 of kleiner")
```

Wat doet een for-loop in Python?

Een `for`-loop herhaalt een blokje code een vast aantal keren.

Voorbeeld:

```
for i in range(5):
    print("Hallo")
```

Noem verschillen tussen Python en Scratch

Scratch	Python
Werkt met blokken die je sleept	Werkt met tekst die je zelf typt
Geen spaties of haakjes nodig	Spaties en dubbele punten zijn belangrijk
Visueel en kleurrijk	Tekstgebaseerd, je moet meer onthouden
Makkelijk te starten zonder fouten	Foutgevoelig bij typefouten of inspringen

Wat leert de student bij 'Pak de Kaas'?

In dit Python/Pygame Zero-project leren studenten onder andere:

- Sprites op het scherm tonen met `screen.blit()`
- De muis laten bewegen via variabelen en `update()`
- Botsingen detecteren tussen muis en kaas
- Willekeurige verplaatsing van de kaas met `random`

- Score bijhouden bij het verzamelen van kaas
- Tijdslimiet instellen zodat het spel eindigt ("Game Over")

Hoe detecteer je dat de muis de kaas raakt?

Gebruik een `if`-statement met `collide` of vergelijk coördinaten van de muis en kaas, bijvoorbeeld:

```
if mouse_x == cheese_x and mouse_y == cheese_y:
    score += 1
```

Waarom gebruik je willekeurige getallen bij het verplaatsen van de kaas na botsing?

Zodat de kaas steeds op een nieuwe plek verschijnt. Bijvoorbeeld met:

```
import random
cheese_x = random.randint(0, WIDTH)
cheese_y = random.randint(0, HEIGHT)
```

Wat doet `random.randint()` precies?

De functie `random.randint(a, b)` geeft een willekeurig geheel getal terug tussen **a** en **b**, inclusief beide grenzen.

□□ *Voorbeeld:*

```
import random
getal = random.randint(1, 10)
print(getal)
```

Dit print een willekeurig getal tussen 1 e

Pak de Kaas

Hoe detecteer je dat de muis de kaas raakt?

Gebruik een `if`-statement met `collide` of vergelijk coördinaten van de muis en kaas, bijvoorbeeld:

```
if mouse_x == cheese_x and mouse_y == cheese_y:
    score += 1
```

Waarom gebruik je willekeurige getallen bij het verplaatsen van de kaas na botsing?

Zodat de kaas steeds op een nieuwe plek verschijnt. Bijvoorbeeld met:

```
import random
cheese_x = random.randint(0, WIDTH)
cheese_y = random.randint(0, HEIGHT)
```

Wat doet `random.randint()` precies?

De functie `random.randint(a, b)` geeft een willekeurig geheel getal terug tussen **a** en **b**, inclusief beide grenzen.

□□ *Voorbeeld:*

```
import random
getal = random.randint(1, 10)
print(getal)
```

Dit print een willekeurig getal tussen 1 en 10, zoals 3 of 9. Handig voor bijvoorbeeld het willekeurig verplaatsen van een sprite in een spel.

Wat gebeurt er als je de muis buiten het scherm laat bewegen?

Als je de variabelen `mouse_x` of `mouse_y` zo aanpast dat de sprite buiten het scherm komt, dan **verdwijnt de muis uit beeld**.

- Het spel zelf blijft werken, maar de speler ziet de muis niet meer.
- De sprite staat dan op een coördinaat die buiten het zichtbare venster valt (bijvoorbeeld x = -50 of y = 1000).

Hoe zorg je ervoor dat de muis niet buiten het spel beweegt? Gebruik je een loop of een if-then?

Om te voorkomen dat de muis buiten het scherm beweegt, gebruik je een `if`**-statement**, geen `loop`. Daarmee controleer je of de muis nog binnen de grenzen is voordat je de positie verandert.

Voorbeeld:

```
if mouse_x < WIDTH - 50:
    mouse_x += 5
if mouse_x > 0:
    mouse_x -= 5
```

?? Opdracht

Maak nu de kennis-check.

? Inleveren

Aan het einde van de kennis-check ontvang je een certificaat. Maak een schermafdruk en lever deze in.

Kennis-check Blok 1 (poging 2)

?? Opdracht

Maak nu de kennis-check.

? Inleveren

Aan het einde van de kennis-check ontvang je een certificaat. Maak een schermafdruk en lever deze in.

datasource

--

Revision #25
Created 2025-06-14 17:30:35 UTC by Max
Updated 2026-07-05 08:41:26 UTC by Max