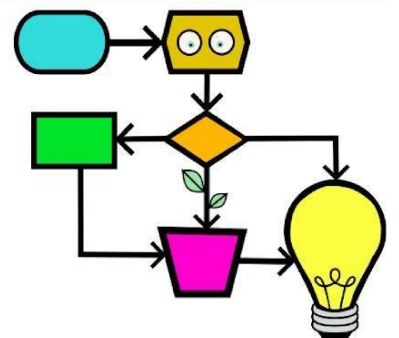
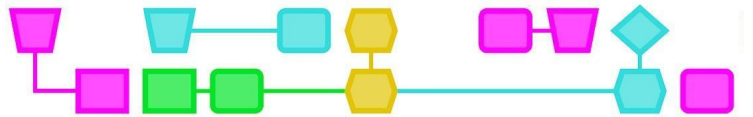


Offline
programmeren





Samenvatting

Een algoritme is een stapsgewijze instructie om een probleem op te lossen. Denk aan een recept, een script voor een toneelstuk of een programma voor de computer. In deze les leren de leerlingen wat een algoritme is. Het begint met een spel dat de leerling laat zien welke instructies een computer kan begrijpen. Hierna programmeren ze elkaar als danscomputer.

Doelgroep: 6-12 jaar (voor kinderen die niet gemakkelijk lezen en schrijven hebben we een differentiatie toegevoegd).

Duur: 55 minuten

Leerdoelen:

De leerlingen leren

- wat een algoritme is;
- hoe je een algoritme in het dagelijks leven kunt gebruiken;
- wat een lus en een voorwaarde zijn en hoe je die in een programma kunt gebruiken.

Online/offline: offline

Computational thinking:

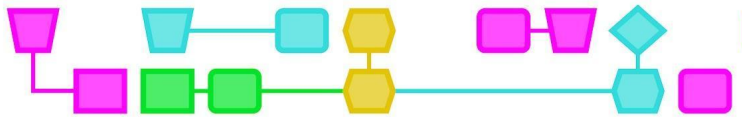
- General skills: creativiteit, samenwerken, logisch nadenken
- CT-foundations: algoritmen, patroonherkenning
- CT-concepten: variabelen, reeksen, lussen en voorwaarden

Materialen:

- Presentatie: De Danscomputer
- Dans Pictogrammen
- Pen
- Papier

Vorbereiding

Lees de les door en zet de presentatie klaar. Bedenk voor de eerste opdracht waar plek A en plek B in de klas zijn.



Lesbeschrijving

Introductie - Van A naar B (10 min)

Leg aan de leerlingen uit dat ze hun eigen algoritme gaan maken. Een algoritme is een proces of een reeks regels die gevolgd moeten worden bij berekeningen of andere probleemoplossende handelingen, voornamelijk door een computer.

Vertel de leerlingen dat jij in deze opdracht de computer bent en dat ze jou moeten 'programmeren' om van punt A naar punt B te komen in de klas. Kies een plek A, het beginpunt, en het eindpunt B. Zorg dat er een paar obstakels in de weg staan, zoals een tafel waar je onderdoor of overheen moet.

Kies 2-3 leerlingen die om de beurt proberen om jou naar punt B te krijgen. Ze moeten instructies geven zoals: 'Doe twee stappen vooruit' en: 'Draai naar rechts.' Vat de instructies extreem letterlijk op. Als leerlingen bijvoorbeeld zeggen: 'Ga vooruit', blijf dan naar voren lopen totdat je ergens tegenaan botst en blijf zelfs dan doen alsof je naar voren loopt. Zo leren de leerlingen dat ze ook 'Stop' moeten zeggen. Je zult zien dat leerlingen moeite hebben alles zo letterlijk mogelijk uit te leggen.

Vertel de leerlingen na afloop dat ze een algoritme hebben gemaakt: een proces of een reeks regels die gevolgd moeten worden bij berekeningen of andere probleemoplossende handelingen, om van punt A naar punt B te gaan. Ze hebben jou stap voor stap instructies gegeven en iedere stap heel letterlijk uitgelegd.

Choreografie - Algoritme (20 min)

Doel: algoritme, duidelijke instructie

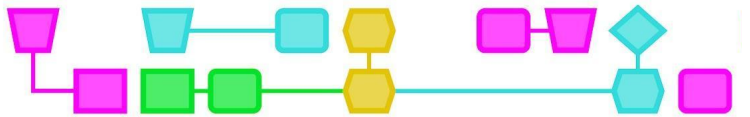
Leg uit dat algoritmes voor verschillende dingen gebruikt kunnen worden, zoals je veters strikken, een recept volgen of voor een danschoreografie. Tijdens deze opdracht bedenken de leerlingen een algoritme voor een dans die een ander groepje gaat uitvoeren. Verdeel de leerlingen in groepjes van 3-4 leerlingen. Elk groepje maakt een dansje (choreografie) van 1-2 minuten.

Stap 1: De leerlingen denken na over welke bewegingen ze willen gebruiken. Dit kan een bestaande dans zijn, zoals 'hoofd, schouders, knie en teen', of een zelfverzonnen dans.

Stap 2: De leerlingen schrijven de dans heel precies uit op papier, zodat een ander groepje hun dans kan uitvoeren als ze de instructies lezen.

Differentiatie: gebruik de danspictogrammen voor leerlingen die nog niet goed kunnen lezen of schrijven.

Stap 3: Leerlingen wisselen de choreografieën uit met een ander groepje. De groepjes krijgen vijf minuten om de choreografie te lezen en te repeteren en vervolgens voor de hele groep te dansen. Benadruk dat de leerlingen alleen mogen uitvoeren wat er op het briefje staat.



Stap 4: Bekijk elkaars dansen en bespreek de volgende vragen met de leerlingen:

- Voerde het groepje de dans precies hetzelfde uit?
- Wat ging er wel goed en wat niet?

De leerlingen hebben het dansje waarschijnlijk niet precies hetzelfde uitgevoerd. Benoem dit en leg uit dat algoritmes stap-voor-stapinstructies zijn die je kunt gebruiken om een probleem op te lossen. Je kunt een algoritme ook gebruiken om een computer iets voor jou te laten doen. In dit geval was het algoritme de beschrijving van de dans en het groepje dat de dans uitvoerde was de computer. Bij de beschrijving van de stappen is precisie belangrijk, omdat de onduidelijkheden fouten in het programma kunnen veroorzaken (= andere uitvoering van de dans). Precieze instructies geven is erg lastig, omdat je alles heel duidelijk uit moet schrijven. Dat is waarschijnlijk waarom de dansjes net iets anders gingen dan hoe ze bedacht waren.

Tip: Zet een muziekje aan bij de uitvoering van de choreografie.

Choreografie - Lussen en voorwaarden (20 min)

Doel: Leren wat lussen en voorwaarden zijn.

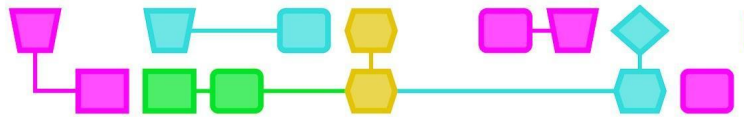
Bespreek met de leerlingen:

- Wat zouden jullie anders doen nu je de uitvoering van de dans hebt gezien?
- Hoe zouden we ervoor kunnen zorgen dat de dans precies zo wordt uitgevoerd als de bedoeling is?
 - Bewegingen versimpelen
 - Duidelijker omschrijven
 - Afspreken welke beweging je maakt
 - Dingen herhalen

Start de presentatie *De Danscomputer*. Laat de eerste dia zien en leg de leerlingen uit dat ze de opdracht nog een keer gaan doen, maar dan met behulp van pictogrammen. Leg uit dat elk pictogram een code is voor een beweging die de computer kan uitvoeren.

- Laat dia 2 zien en leg de pictogrammen uit.
- Laat dia 3 zien en voer dit dansje samen met de klas uit. Bespreek hoe je kunt laten zien dat ze het dansje drie keer moeten herhalen.
- Laat op dia 4 zien dat je de stappen allemaal kunt herhalen, maar dat dat heel onduidelijk zou worden. Bespreek hoe je het korter zou kunnen opschrijven.
- Laat dia 5 zien. Dit noem je een lus, een herhaling. Bespreek of ze nog een plek zien om een lus toe te passen (bij het klappen van de handen).
- Laat hierna dia 6 zien. Dit noem je een voorwaarde, een 'als..., dan...'-regel. Als iets gebeurt, dan stopt iets anders. *Als* de muziek aan is, *dan* wordt het dansje herhaald. *Als* de muziek uitgaat, *dan* stopt het programma.

De leerlingen gaan nu in groepjes een choreografie maken met behulp van de pictogrammen. Zorg dat ze ten minste twee lussen en een voorwaarde gebruiken.



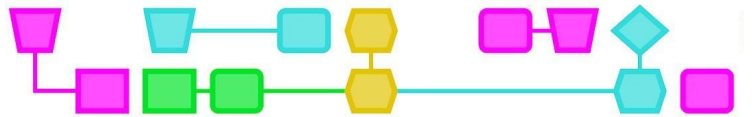
Bespreek na de opdracht wat de leerlingen vandaag hebben geleerd. Ze kunnen nu een danscomputer programmeren met een functie, lus en voorwaarden. Bij de danscomputer hebben ze gebruikgemaakt van pictogrammen als codetaal. Op de computer kunnen ze andere codetalen gebruiken om dingen te programmeren, zoals Scratch, MakeCode of Kodetu. In deze codetalen kunnen ze ook functies, lussen en voorwaarden gebruiken. In de informatica ('computerwetenschap') zijn algoritmes nodig om computerprogramma's te coderen.

Algoritme	Een reeks logische instructies voor het uitvoeren van een taak. In de informatica zijn algoritmen nodig om computerprogramma's te ontwerpen.
Programma	Opeenvolging van instructies voor een computer.
Functie	Een blok georganiseerde, herbruikbare code die wordt gebruikt om een enkele, verwante actie uit te voeren.
Instructie	Een specifieke opdracht die door een computer uitgevoerd kan worden.
Lus	Een herhaling van een (reeks) instructie(s).
Voorwaarde	Een 'als..., dan...'-regel die stelt dat er iets moet gebeuren voordat het andere gebeurt.

Tip: Laat in Scratch, Kodetu of MakeCode zien hoe een functie, lus en voorwaarde er in codetaal uitzien.

Afsluiting (5 min)

Vertel de leerlingen dat ze deze les zelf een algoritme hebben gemaakt! Leg uit dat er heel veel algoritmes in de wereld zijn. Bijvoorbeeld bij stoplichten, of op sociale media zoals TikTok. Maar ook bij tandenpoetsen, de tafel dekken en bij rekensommen gebruik je stap-voor-stapinstructies die dingen makkelijker laten gaan. Ook lussen en voorwaarden zie je terug in het dagelijks leven, bijvoorbeeld in een recept waar je iets moet herhalen, maar ook bij de ongeschreven regels dat je als je naar de wc bent geweest je handen moet wassen, of dat je als je op reis gaat een koffer moet inpakken. Algoritmes, lussen en voorwaarden zijn overal!



Colofon

© CTPrimED

This publication is a product of CTPrimED (2021-1-NL01-KA210-SCH-000031319), funded with support from the Erasmus+ Programme of the European Union. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use that may be made of the information contained therein.

Project Coordinator

NEMO Science Museum, The Netherlands

Partners

Universidad de la Iglesia de Deusto Entidad Religiosa, Spain
Stichting Children's Science Museum Curacao, Curacao



Co-funded by the
Erasmus+ Programme
of the European Union



Universidad de Deusto
University of Deusto

Deusto

