

BOMBERBOT

LES 10

PROGRAMMA FLOW

LESDOELEN

- De leerlingen kunnen programma's herstructureren met behulp van control statements (loops, functies en conditionals).
- De leerlingen begrijpen het effect van herstructureren op de programma flow.
- De leerlingen kunnen conditionals, loops en functies gebruiken om betere algoritmes te schrijven om problemen op te lossen.

NIEUWE BEGRIPPEN

Programma flow = de volgorde waarin instructies in het programma worden uitgevoerd.

Herstructureren = het proces waarbij de kwaliteit van de code wordt verbeterd door onnodige instructies te verwijderen en de structuur van de code aan te passen.

BEKENDE BEGRIPPEN

Programmeren, sequentie, for loop, loop teller, absolute richtingen, relatieve richtingen, algoritme, functie, functie definiëren, functie aanroepen, until loop, oneindige loop, variabele, conditional, if statement.

BENODIGDHEDEN

- Programma Flow Presentatie (Powerpoint)
- Programma Flow Antwoordengids (PDF)

LESPLAN (60 minuten)

Informatie voor de leerkracht

Introductie nieuw concept (pagina 1)
Herstructureren (pagina 2)

Lesinhoud

Herhaling (pagina 3)
Uitleg nieuw concept (pagina 5)
Speel Missie (pagina 6)
Afsluiting (pagina 6)

INFORMATIE VOOR DE LEERKRACHT

INTRODUCTIE NIEUW CONCEPT

In de afgelopen lessen hebben de leerlingen geleerd over conditionals, functies, variabelen en loops - programmeerconcepten waarmee we onze programma in kunnen korten en efficiënter kunnen maken. Hiernaast veranderen deze programmeerconcepten ook de **programma flow** door de volgorde waarin instructies worden uitgevoerd te veranderen. Deze programmeerconcepten zijn net wegwijzers die ons programma een andere weg of route wijzen - ze veranderen de flow (stroom) van ons programma.



Je kunt een programma zien als een lange rivier, een stroom van allerlei instructies. De rivier heeft een startpunt en stroomt in de richting van het eindpunt. Programma's 'stromen' normaliter van begin tot einde, van boven naar beneden of van links naar rechts, tenzij een wegwijzer ze de andere kant op stuurt. Deze wegwijzers zijn programmeerconcepten als loops, functies en conditionals. Deze wegwijzers noemen we in het programmeren ook wel '**control statements**', omdat ze controle uitoefenen op de richting waarin het programma stroomt. Laten we eens kijken hoe deze control statements de programma flow beïnvloeden.

Extra informatie: De rivier is het programma met instructies. Bomberbot vaart over de rivier en voert de instructies uit, net als een computer. Bomberbot volgt dus de 'flow' van de rivier.

LOOPS

Wanneer het programma (de rivier) een loop bevat, vaart Bomberbot vanaf het punt dat de loopt begint telkens hetzelfde rondje met instructies. Bomberbot **herhaalt** dit rondje totdat de loop teller op 0 staat (for loop) of totdat een bepaald doel is bereikt (until loop).



FUNCTIES

Functies verwijzen Bomberbot als het ware naar een ander deel van de rivier. De rivier stuurt Bomberbot naar een vijver waar de instructies in de functie zijn opgeslagen. Bomberbot voert deze instructies uit en keert hierna weer terug naar de rivier. Iedere keer wanneer Bomberbot de wegwijzer (F1 vlag of instructie) tegenkomt, keert Bomberbot terug naar de vijver om de opgeslagen instructies uit te voeren.

CONDITIONALS

Conditionals laten Bomberbot telkens checken of een if statement waar is of niet. Als de if statement waar is, dan volgt Bomberbot een andere stroom met instructies, waarna hij terugkeert naar de rivier. Als de if statement niet waar is, volgt Bomberbot de normale stroom van de rivier.



HERSTRUCTUREREN

Programmeurs duiken vaak opnieuw in de al geschreven code om deze te herstructureren en verbeteren met behulp van loops, functies, conditionals en andere programmeerconcepten. Dit proces van het verbeteren van de code, terwijl het doel en de uitkomst van de code gelijk blijft, noemen we de **herstructurering** (of *'refactoring'*) van code. In dit proces veranderen programmeurs de structuur van de code met behulp van programmeerconcepten als loops, functies en conditionals, zodat het programma beter en efficiënter wordt. Het herstructureren maakt het programma overzichtelijker en voorkomt bugs.

Hieronder een aantal veelvoorkomende technieken die worden gebruikt bij het herstructureren van code en waarmee de **programma flow** wordt aangepast:

- **Code opdelen in kleinere stukken**

Functies spelen hierin een belangrijke rol, omdat programmeurs kleine stukken code kunnen opslaan in functies en in andere delen van het programma kunnen aanroepen. Het herstructureren van code met behulp van functies maakt de code overzichtelijker voor programmeurs.

- **DRY: "Don't repeat yourself!"**

Code herhalen is een van de grootste nachtmerries van programmeurs. Zij hebben daarom zelfs een nieuwe term in het leven geroepen: goede programma's zijn DRY. DRY staat voor **Don't Repeat Yourself** (herhaal jezelf niet). DRY programma's zijn dus efficiënt. Programmeurs kunnen onder andere loops, functies en conditionals gebruiken om hun programma te herstructureren en DRY te maken.

- **Onnodige instructies verwijderen om het programma te verkorten**

Alle computerprogramma's nemen ruimte op in de computer. Hoe meer instructies, hoe langer het programma en hoe meer tijd het kost om het programma uit te voeren. Soms wordt zelfs de gehele computer hier trager door! Verwijder dus altijd onnodige instructies.

HERSTRUCTUREREN IN BOMBERBOT

In deze Missie zullen leerlingen bestaande programma's ook gaan herstructureren om zo efficiëntere algoritmes te schrijven. Tijdens dit proces zullen de leerlingen nogmaals merken welke invloed bepaalde programmeerconcepten op de programma flow hebben. De levels in deze Missie zijn afkomstig uit eerdere Missies. De oplossingen die al klaar staan in elk level zijn echter niet de meest efficiënte oplossingen. De leerlingen zullen de programma's moeten verbeteren en herstructureren om tot de 3 sterren oplossing te komen.

(Optioneel) Indien er genoeg tijd is, kunnen leerlingen ook terug gaan naar levels uit eerdere Missies waar zij minder dan drie sterren voor hebben gekregen. Kunnen zij inmiddels wel de beste oplossing vinden?

LESINHOUD

HERHALING (15 minuten)

Dia 1: LESAFBEELDING	 Les 10 PROGRAMMA FLOW
Dia 2: PROGRAMMEREN	Doen: Herhaal wat programmeren is. Herinner leerlingen aan de unplugged activiteit van de bijbehorende les. • Programmeren: het creëren van instructies (opdrachten) die door een computer kunnen worden uitgevoerd om een bepaalde taak te voltooien. <i>Unplugged Activiteit: Menselijke Bomberbot (één instructie tegelijk)</i>
Dia 3: SEQUENTIES	Doen: Herhaal wat sequenties zijn. Wat was het verschil tussen absolute en relatieve richtingen ook alweer? Herinner leerlingen aan de unplugged activiteiten van de bijbehorende lessen. • Sequentie: een opeenvolging van instructies die stap voor stap wordt uitgevoerd, van begin tot eind. <i>Unplugged Activiteit: Menselijke Bomberbot (gehele reeks instructies in één keer)</i> • Relatieve richting: de beweging die een object maakt uitgaande van de positie waar het object zich op dat moment bevindt. <i>Unplugged Activiteit: Menselijke Bomberbot (relatieve richtingen)</i>
Dia 4: LOOPS	Doen: Herhaal wat een loop en loop teller is. Wat is het verschil tussen for loops en until loops? Herinner leerlingen aan de unplugged activiteiten uit de bijbehorende lessen. • For Loop: een set instructies die een bepaald aantal keer wordt herhaald. <i>Unplugged Activiteit: for loop dans</i> • Loop teller: geeft aan hoe vaak een sequentie nog herhaald moet worden. • Until loop: een loop die blijft herhalen totdat er aan een bepaalde voorwaarde voldaan wordt. <i>Unplugged Activiteit: until loop dans (stoelendans of stopdans)</i>

Dia 5: ALGORITMES	Doen: Herhaal wat algoritmes zijn. Herinner leerlingen aan de unplugged activiteit uit de bijbehorende les. • Algoritme: een proces dat stap-voor-stap wordt gevolgd door de computer om een taak te verrichten; een soort recept of formule om een probleem op te lossen. <i>Unplugged Activiteit: Frisbee</i>
Dia 6: FUNCTIES	Doen: Herhaal wat functies zijn. Wat weten leerlingen nog van functie definiëren en aanroepen? Herinner leerlingen aan de unplugged activiteit uit de bijbehorende les. • Functie: een sub-sequentie van instructies die opgeslagen en hergebruikt kan worden in een programma; een routine die een programma opdeelt in kleinere delen. <i>Unplugged Activiteit: Functie orkest</i> • Functie definiëren: het definiëren van een sub-sequentie van instructies (functie); aangeven welke instructies binnen een functie worden opgeslagen. • Functie aanroepen: Het uitvoeren van instructies binnen een functie in het programma.
Dia 7: VARIABELEN	Doen: Herhaal wat variabelen zijn. Herinner leerlingen aan de unplugged activiteit uit bijbehorende les. • Variabele: Informatie kan worden opgeslagen in variabelen. Een variabele bevat een waarde die kan veranderen. <i>Unplugged Activiteit: De variabelen pot</i>
Dia 8: CONDITIONALS	Doen: Herhaal het concept conditionals, if statements en hoe deze gebruikt worden in Bomberbot. Herinner leerlingen aan de unplugged activiteit uit de bijbehorende les. • Conditionals: Instructies die alleen uitgevoerd worden bij bepaalde condities (bijvoorbeeld wanneer er een geel blok is). <i>Unplugged Activiteit: If Games</i> • If statement: Een set instructies welke alleen uitgevoerd wordt als aan een bepaalde conditie voldaan wordt (de vraag wordt met “ja” beantwoord). We gebruiken if statements in Bomberbot om de robot te vertellen welke instructies hij moet uitvoeren als hij voor een bepaald spelstuk staat. Zo kunnen we een korter en efficiënter programma schrijven.

UITLEG NIEUW CONCEPT

(10 minuten)

Dia 9: PROGRAMMA FLOW	Vertel: We kunnen een programma vergelijken met een rivier. Een rivier waarbij de instructies vanaf het begin naar het einde van de rivier stromen. Dit jaar hebben we verschillende programmeerconcepten gebruikt om ons programma - onze rivier - korter te maken. Met behulp van deze concepten veranderen we de flow, oftewel de route van het programma. We veranderen dan de volgorde waarom instructies worden uitgevoerd en de stroom van onze rivier. We noemen de programmeerconcepten die de programma flow veranderen ook wel controle statements (controle stellingen), omdat deze controle uitoefenen op de flow van het programma. Op het bord zien we een voorbeeld van een 'normaal' programma. Het programma begint bij het begin en eindigt bij het einde, zonder dat de flow van het programma ergens wordt veranderd. We kunnen dit programma, of onze rivier, korter maken door de flow te veranderen met behulp van control statements. Vraag: Stel we moeten deze rivier (het programma) korter maken. Welke controle statements kunnen we hiervoor gebruiken? <i>Antwoord: for loops, functies, until loops en conditionals.</i>
Dia 10: FOR LOOPS	Vertel: Een for loop vertelt het programma om een bepaalde set instructies iedere keer te herhalen, totdat de loop teller op nul staat. Je kunt een programma met een for loop zien als de rivier op het bord. De vlag vertelt Bomberbot dat hij vier keer de instructies in het rondje uit moet voeren. Doen: Laat zien dat Bomberbot uiteindelijk hetzelfde programma uit als in de vorige slide, maar dat deze rivier korter en dus efficiënter is.
Dia 11: FUNCTIES	Vertel: Functies verkorten ook de rivier (het programma), omdat we de instructies die we iedere keer herhalen nu opslaan in een soort vijver. Iedere keer dat we de functie moeten uitvoeren kunnen we naar deze vijver gaan en de instructies hierin opnieuw uitvoeren. We hoeven de instructies dus niet iedere keer uit te schrijven.
Dia 12: CONDITIONALS	Vertel: Conditionals zorgen ervoor dat ons programma de route splitst, wanneer een bepaalde voorwaarde waar is. Wanneer de conditie waar is (Bomberbot staat voor het spelstuk), dan volgt Bomberbot tijdelijk andere route voordat hij terug keert naar de reguliere route.

SPEEL MISSIE

(30 minuten)

Dia 13: BOMBERBOT MISSIE 10

Doen: Leerlingen kunnen inloggen op Bomberbot met hun gebruikersnaam en wachtwoord. Zijn de leerlingen hun wachtwoord en gebruikersnaam vergeten? Deze kun je als leerkracht terugvinden in jouw platform (klik 'Mijn Klassen' en open de klas). Klik op het oogje naast 'Wachtwoord' om de wachtwoorden te tonen.

Belangrijk!

Speel **alle levels** in deze Missie. In ieder level is al een oplossing te zien, maar deze oplossing is te lang! Leerlingen moeten de oplossingen herstructureren met behulp van loops, functies en conditionals. Leerlingen die snel klaar zijn kunnen ook levels uit eerdere Missies opnieuw spelen en proberen een hogere score te halen. Ook levels die nog niet voltooid waren kunnen nu worden gespeeld.

Tip! Vergeet niet de Missie van tevoren te ontgrendelen in jouw leerkrachtenplatform, zodat de leerlingen de levels in de Missie kunnen spelen.

AFSLUITING

(10 minuten)

Dia 14: RONDJE KLAS

Vraag: Waarom gebruiken programmeurs functies, loops en conditionals?
Antwoord: *Om de programma's korter, efficiënter en overzichtelijker te maken, tijd te besparen en bugs te voorkomen.*

Doen: Herhaal hoe functies, loops en conditionals de programma flow veranderen. Bespreek moeilijke levels klassikaal.