|  |
| --- |
| http://static.thomasmore.be/huisstijl/images/TM_logo_wit_web.jpg **Campus Geel** **Kleinhoefstraat 4 | 2440 Geel | België Tel. 014 56 23 10** |
| **lesvoorbereiding** |
| **Naam lesgever:** Jakob Robijns | **Naam begeleider TM:**  |
| **Datum:**  | **Bibliografie :** Ondernemen en IT - Scratch |
| **Onderwijsvorm, studierichting en graad + leerjaar:**  | **Lokaal+uur:**  |
| **Leerplan:** D/2011/7841/045 | **Leervak:** Informatica |
| **Lesonderwerp (agenda):** "Scratch: Grafische opdrachten, Huis met kruis, Dialoog, Raak de kat en wijzig de kleur, stuur de kat met de pijltjestoetsen" |
| **Leerplandoelstelling/eindterm:** Deelcompetentie 11.1 - Inzien wat een algoritme is. Het verschil tussen een algoritme en een programma kennen11.1.3Deelcompetentie 11.2 - Een probleemstelling omzetten in een werkend programma. De verschillende controlestructuren kennen en gebruiken11.2.1, 11.2.6, 11.2.7 |
| **Beginsituatie:** De leerlingen hebben in de vorige lessen al met Scratch gewerkt. Ze kennen de volgende blokjes code: Alle bewegingen, gebeurtenis wanneer er op vlag wordt gedrukt, herhalen en tekenen met de pen.Ze kunnen ook de x en y coördinaten lezen van het speelveld. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DOELSTELLINGEN/FASEN** | **Didactische werkvorm** | **Gr** | **LEERSITUATIES** **Onderwijs- en leeractiviteiten / feedback** | **Media** | **Tijd** |
| I Motivatiefase |  |  |  |  |  |
| 1. Vereiste voorkennis nagaan
 | Doceren/OLGZelfstandig leren | KI | *(Herhaling: lln kunnen de x en y coördinaten van een punt in een veld vinden)*Op het scherm staat het veld zoals in het handboek, de lln zeggen welke coördinaten ze denken dat elke bol is:* Rood: (240,0)
* Wit: (240, -180)
* Grijs: (-240, 180)
* Groen: (-240, 0)
* Zwart: (-240, -180)

*Als het minder vlot gaat, kan de lkr enkele punten aanduiden op het veld waarbij ze een schatting van de coördinaten kunnen maken.* De lln gaan nu aan de oefening ‘Huis met een kruis” werken. Voor ze op scratch het gaan maken, beginnen ze het huis op papier te maken.Stappenplan:1. De lln proberen eerst hoe ze het huis kunnen tekenen zonder over dezelfde lijn te gaan en dezelfde lijn te tekenen.
2. Als dat lukt zetten ze pijlen en een nummer bij elke lijn.
3. Zet bij elk punt de X en Y coördinaten.
4. Als alles lukt maken ze het in scratch.
5. Test de code en maak wijzigingen op papier indien nodig.

*Als er al lln klaar zijn kunnen ze aan de extra oefeningen in het handboek werken. Ze mogen er ook thuis aan werken.**Als het bij iedereen goed lukt gaan we verder, anders overloopt de lkr de oefening met behulp van de PowerPoint.*De lkr legt het belang uit van eerst je code op papier uit te werken en voor te bereiden. | Cursus: “ondernemen en IT – Scratch”PowerPoint+ Website: ScratchOefening 3 | 5”15” |
| 1. Motiveren/belangstelling wekken

 | DocerenRollenspel | K | De lkr legt nu het concept van ‘Signalen’ uit. Dit door middel van een kort rollenspel.Rollenspel:De hele klas bestaat uit verschillende sprites, en we hebben allemaal een stuk code dat naar de lkr luistert.De lkr zegt nu bepaalde taken zoals: “sta recht”, “ga zitten”, “Steek rechterhand omhoog”,… |  | 5” |
| 1. Belang van de les
 | Doceren |  | In het dagelijks leven is het heel gemakkelijk om signalen te kunnen sturen. Als ik aan de lln iets zeg om te doen, is dat een soort van signaal sturen. In programmeren is er ook zo een concept, en dat gaat heel handig zijn als we naar andere sprites een signaal kunnen sturen. |  | 3” |
|  |  |  |  |  |
| **Leerdoel 1 (classificatie)**De lln kunnen door middel van signalen communiceren naar andere sprites. (Toepassen) | Demonstratie | K | Informatieverwerving 1: Zoals ze bij de motivering hebben gezien, is het handig om te communiceren naar andere sprites.De lkr demonstreert waar ze de blokjes kunnen vinden in een nieuw project:Je kan een signaal verzenden met het blokje ‘Zend signaal x’ bij gebeurtenissen. Bij een andere sprite (of ook bij de zelfde sprite) kan je dan het blokje ‘wanneer ik signaal x ontvang’ gebruiken om de code te starten zodra het signaal is verzonden. | Scratch | 5” |
| Zelfstandig leren | I | Informatieverwerking 1:De lln werken zelfstandig aan opdracht 4 in het handboek.Vermeld zeker dat ze de oefeningen op tijd en goed opslaan,De lkr checkt of het bij iedereen is gelukt.*Als er lln sneller klaar zijn kunnen ze de uitbreiding maken.**Als de lln het idee van signalen nog niet snappen, kan de lkr een soortgelijke oefening als voorbeeld demonstreren.* | ScratchOpdracht 4 | 10” |
|  |  | Feedback 1: De lkr loopt tijdens de oefening door de klas en bekijkt bij wie het wel/niet goed gaat.Als het bij meerdere lln niet goed lukt kan er klassikaal een extra voorbeeld worden gedemonstreerd. |  | 6” |
| **Leerdoel 2 (classificatie)**De lln kunnen het ‘als dan’ blokje gebruiken in hun code. (Toepassen) | RollenspelDemonstreren | K | Informatieverwerving 2: De lkr legt het concept van het als-dan blokje uit door middel van een kort rollenspel.1 lln is vrijwilliger en staat recht. Op het bord staan enkele als-dan situaties.* Als de lkr de hand omhoog steekt, dan gaat de lln zitten
* Als de lkr ‘rechtstaan’ zegt, dan staat de lln recht
* Als de lkr naar het bord kijkt, dan praat de lln
* Als de lkr naar de lln kijkt, dan is de lln stil

De lkr doet enkele van deze situaties en de lln moet dan gepast reageren.De lkr toont op de PowerPoint hoe dit in code er kan uitzien.Hierna toont de lkr dat je de blokjes vind bij ‘besturen’. | PowerpointScratch | 10” |
| Zelfstandig leren | I | Informatieverwerking 2:De lln maken opdracht 5 en 6De lkr overloopt de opdrachten mondeling voor ze beginnen.Als de lln klaar is met een oefening mogen ze niet vergeten op te slaan.*Als een lln sneller klaar is, zijn ze vrij om de 3e en 4e oefening verder te maken.* *Als een oefening bij een lln onduidelijk is, is een tip om de code stuk per stuk op te bouwen. Bv bij oef 5:** *We beginnen met het vlaggetje, dus dat kunnen we er al zetten*
* *We willen het uiterlijk veranderen als de muisaanwijzer de sprite raakt*
* *Dit gaan we dan telkens herhalen*
* *Voeg er nog een wacht blokje bij*
 | ScratchOefening 5Oefening 6 | 20” |
|  |  | Feedback 2: De lkr loopt tijdens de les rond en helpt de lln.We overlopen de oplossing op bord en zien dat iedereen het juiste heeft. Mocht het bij meerde lln niet lukken, kan de lkr altijd de oefening klassikaal maken |  | 5” |
| Slot/Lesafronding: |  | K | Als laatste testen we de kennis door middel van een Kahoot quiz.<https://create.kahoot.it/details/5b864fe6-45ab-4cc5-880b-6b4c006a74b2>Als laatste wordt er nog 3 vlugge vragen gesteld:* Wat ging er goed?
* Wat kon er beter?
* Wat neem je mee uit de les?
 | Kahoot | 7” |
| III Evaluatiefase  |  |  |  |  |  |
|  |  |  | Tijdens de oefeningen bekijkt de lkr bij wie het wel/niet goed gaat, maar er is geen evaluatie of taak gepland voor deze les. |  |  |
| IV Differentiatiefase |  |  |  |  |  |
|  |  |  | In de cursus staan er uitbreidingen bij de meeste oefeningen. De lln kunnen als ze klaar zijn met het belangrijkste van de oefening de uitbreidingen zelfstandig maken.Als er lln een probleem hebben met iets te begrijpen of niet op tijd klaar zijn met alle oefeningen, dan kunnen ze zelfstandig aan de oefeningen thuis werken door middel van het handboek.Bij de oefeningen is altijd een voorbereiding en oplossing om de leerlingen eventueel via het bord te helpen en/of meer uit te leggen. |  |  |

|  |
| --- |
| **Bijlagen in LV-mapje (vink aan):****󠆡** kopie uit leerplan**󠆡** kopie leerinhoud (uit handboek, cursus…)**󠆡** hand-outs PPT**󠆡** oefeningen, werkblaadjes, toets + oplossleutel**󠆡** didactisch materiaal (vb. stappenplan) |

|  |
| --- |
| OPMERKINGEN/BIJSTURING |
| Alle oplossingen vind je online terug:3) Huis met kruis: <https://scratch.mit.edu/projects/469364297>4) Dialoog: <https://scratch.mit.edu/projects/469369807>5) Raak de kat: <https://scratch.mit.edu/projects/469383707>6) Stuur de kat: <https://scratch.mit.edu/projects/469387156> |

|  |
| --- |
| LEERPLANUITTREKSELCompetentie 11: Algoritmisch denkenAlgemene didactische wenken* “Eerst denken en dan doen” is het belangrijkste bij algoritmisch denken.
* Bij complexere problemen een algoritme schematisch (laten) voorstellen door een Nassi-Shneiderman diagram (NSD). In de andere gevallen volstaat een uitgebreide beschrijving van de verwerkingsfase in de analyse.
* Gebruik een actuele, motiverende ontwikkelomgeving waarover de leerlingen ook thuis kunnen beschikken. Het is niet de bedoeling de leerlingen een programmeertaal aan te leren. Er zijn voldoende gratis alternatieven die voor deze doeleinden geschikt zijn zoals Greenfoot, Scratch, Small Basic, Alice …
* Toon eens de werking van een gemaakt programma in een professionele ontwikkelomgeving.
* De complexiteit van en de soort oefeningen is afhankelijk van de studierichting. In een meer wiskundige richting komen uiteraard meer wiskundige probleemstellingen aan bod.
* Geef de leerlingen de tijd en de ruimte om zelfstandig oefeningen uit te werken. Succeservaringen zijn heel belangrijk.
* Vertrek vanuit de fouten van de leerlingen bij het aanleren van foutanalyse. Leer hen zelf hun fouten te analyseren. Gebruik hiervoor onder meer de mogelijkheden van de ontwikkelomgeving.
* Leer de leerlingen ook een werkend programma aanpassen.
	1. Inzien wat een algoritme is. Het verschil tussen een algoritme en een programma kennen
		1. Het begrip algoritme kaderen in het dagelijkse leven.
		2. Omschrijven wat een algoritme en een programma is.
		3. De verschillende stappen in het oplossen van een probleem kennen en continu toepassen bij het oplossen van problemen nl. probleemdefinitie, analyse, algoritme, programma, testen en documenteren.

Didactische wenken* Vertrekkend vanuit een aantal algoritmes uit het dagelijkse leven, de leefwereld van de leerlingen, het begrip algoritme aanbrengen. Het is niet de bedoeling om hier heel lang bij stil te staan en de leerlingen zelf algoritmes uit het dagelijks leven te laten opstellen.
	1. Een probleemstelling omzetten in een werkend programma. De verschillende controlestructuren kennen en gebruiken
		1. Ongeacht de eenvoud of de complexiteit van een probleem, een analyse maken en vooraleer tot het gebruik van de computer over te gaan, minimaal voor zichzelf het principe van een oplossing formuleren.
		2. De verschillende elementen en mogelijkheden van de gebruikte ontwikkelomgeving doelgericht aanwenden.
		3. Met variabelen en constanten werken.
		4. De toekenningsoperator gebruiken.
		5. Rekenkundige-, vergelijkings- en logische operatoren integreren.
		6. De controlestructuren met hun kenmerken kennen en toepassen waaronder de sequentie, de selectie en de iteratie.
		7. De eenzijdige, tweezijdige en geneste selectie of keuze toepassen.
		8. De voorwaardelijke en begrensde herhaling gebruiken.

Didactische wenken* Laat de leerlingen inzien wat de betekenis is van een toekenningsoperator. Onderstreep het verschil met de wiskundige gelijkheidsoperator. Maak duidelijk dat de richting van deze operator van rechts naar links gaat. Gebruik ook constructies zoals var = var + 1, var1 = var1 + var2 om de werking van een toekenningsoperator te verduidelijken.
* Beperk de voorwaardelijke herhaling tot de voorwaardelijke herhaling met aanvangsvoorwaarde of afbreekvoorwaarde.
* Het algoritmisch denken van de leerling worden niet aangescherpt door voorgekauwde probleemstellingen en oplossingen te presenteren. Aanvankelijk is de voorbeeldfunctie van de leraar van groot belang. De leerlingen moeten de wijze waarop de leraar systematisch algoritmisch denkt, leren nabootsen. De leraar kan dit bevorderen door als het ware ‘luidop na te denken’. De leerlingen moeten oplossingen gaandeweg zien ontstaan en hieraan meer en meer actief participeren. Hun zelfwerkzaamheid is van groot belang en moet voortdurend groeien.
 |