



FuzeField Scratch handleiding

Wat is Scratch	2
Layout FuzeField	3
Scratch in FuzeField	4
Beheer	4
Weergave	4
Scratch blocks	5
Tips en tricks	7
Aandachtspunten	8
Geavanceerd	9
Schermgrootte	9



Wat is Scratch

Scratch is een visuele ontwikkelomgeving waar kinderen, maar zeker ook volwassenen, ervaring kunnen opdoen met het bouwen van spellen en applicaties. Abstract logisch en probleemoplossend denken worden maximaal aangesproken tijdens het bouwen van eigen spelletjes.

En wat is er mooier om je eigen creatie vervolgens op het FuzeField gespeeld te zien worden door alle kinderen van je school?

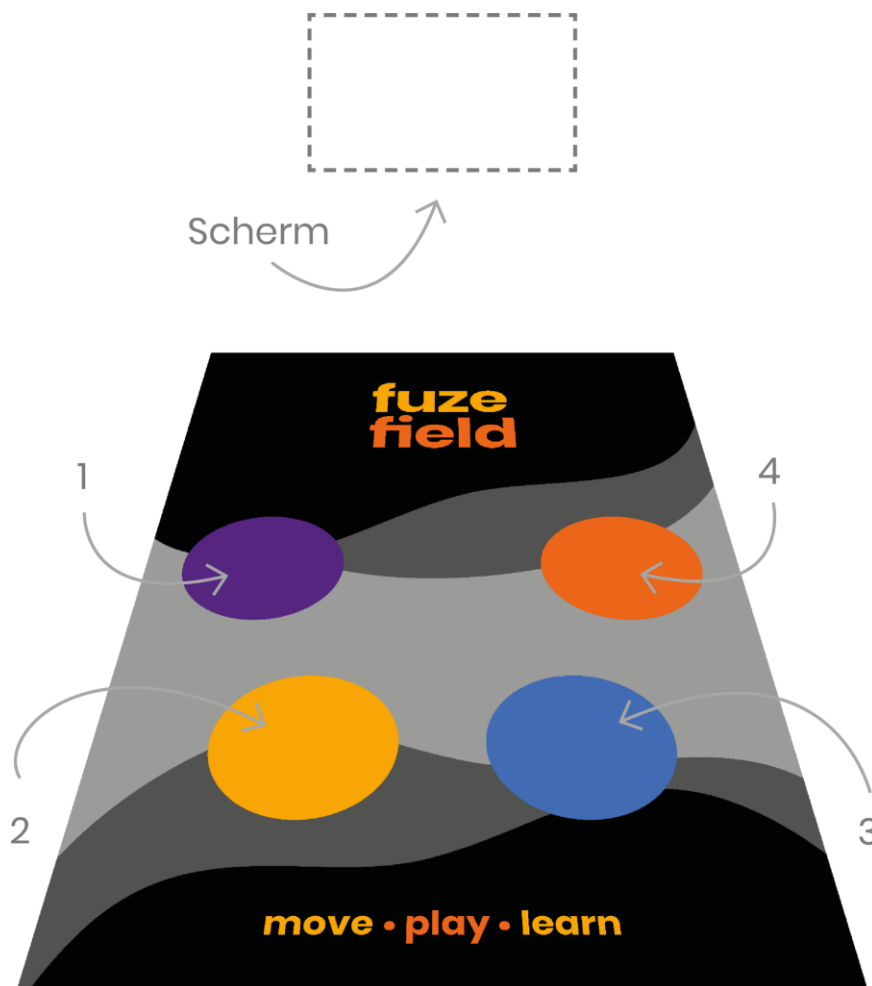
Je vindt Scratch hier: <https://scratch.mit.edu/>

Ons Scratch account vind je hier: <https://scratch.mit.edu/users/FuzeField/>

Hier kun je ook de voorbeelden van FuzeField Scratch games bekijken ter inspiratie van je eigen games.

Layout FuzeField

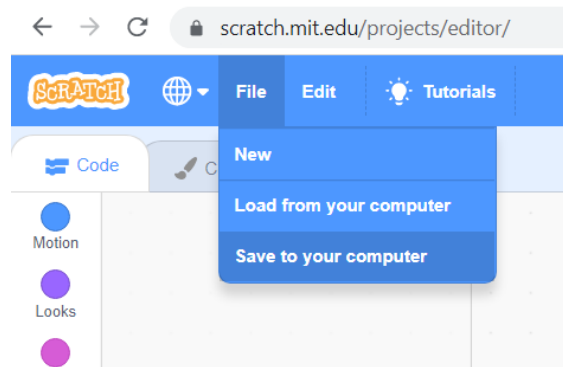
Het FuzeField bestaat uit vier tegels, genummerd van 1 tot en met 4 zoals onderstaand weergegeven. Dit zijn tevens de cijfers waarop je kan controleren binnen je Scratch game om te zien of er gesprongen is.



Scratch in FuzeField

Beheer

Om je Scratch spel te draaien op een FuzeField moet deze eerst gecontroleerd en geschikt gemaakt worden. Hiervoor moet je je Scratch spel exporteren/downloaden via de menuoptie binnen Scratch:



Export dan uploaden naar beheer... nader in te vullen als beheer zover is...

Weergave

Op het FuzeField zal een Scratch game draaien op een blanco scherm (enkel een bovenbalk is zichtbaar alwaar tijd en score getoond kunnen worden (op basis van variabelen welke door het FuzeField uit het spel zullen worden uitgelezen).

De spelbouwer is verantwoordelijk voor de interface voor de gebruiker. Spelontwikkeling zal binnen Scratch kunnen plaatsvinden en de sensoren kunnen uitgelezen worden op basis van de nummers van de sensoren (zie layout FuzeField). Er zijn geen extra plug-ins nodig in Scratch en het spel is op de PC te testen met behulp van het toetsenbord.

- Het FuzeField leest een tweetal variabelen uit voor het tonen van informatie op de bovenbalk en om mee te nemen als resultaat naar het score scherm:
 - "score": aantal behaalde punten tot dan toe;
 - "timer": aflopende tijd in seconden
- Toetsen 1 t/m 4 worden automatisch vertaald naar de sensoren
- Zorg dat een spel eindigt, ook als het FuzeField wordt verlaten om te voorkomen dat het FuzeField "hangt" op het betreffende spel.

Scratch blocks

De volgende blocks van Scratch zijn geïmplementeerd, verdeeld per groep zoals ook in Scratch (*in cursief een eventuele opmerking*):

- Motion blocks
 - move ... steps
 - turn ... degrees *(beide richtingen op)*
 - go to [...]
 - go to x ... y ... *(willekeurige plek of andere object)*
 - glide ... secs to x ... y ...
 - glide ... secs to [...]
 - point in direction ... *(willekeurige plek of andere object)*
 - change x by ...
 - set x to ...
 - change y by ...
 - set y to ...
 - if on edge, bounce
 - set rotation style ...
 - x position *(variabele)*
 - y position *(variabele)*
 - direction *(variabele)*
- Looks blocks
 - switch costume to ... *(naam en nummer)*
 - next costume
 - switch backdrop to ... *(naam en nummer)*
 - next backdrop
 - change size by ...
 - set size to ...
 - show
 - hide
 - go to [front/back] layer
 - go [forward/backward] ... layers
 - costume [...]
 - background [...]
 - size *(variabele)*
- Sound blocks
- Events blocks
 - when flag clicked *(starten game)*
 - when ... key pressed

- when I receive ...
- broadcast ... *(stuurt de 'when I receive' aan)*
- broadcast ... and wait *(stuurt de 'when I receive' aan)*
- Control blocks
 - wait ... seconds
 - repeat ... times
 - forever
 - if ... then
 - if ... then ... else ...
 - wait until
 - repeat until ...
 - stop [...]
 - when I start as a clone *(script dat wordt gestart bij ontstaan clone)*
 - create clone of ...
- Sensing blocks
 - touching ... *(alleen objecten en randen, bounding box)*
 - distance to ... *(alleen objecten)*
 - key ... pressed
 - [...] of [...] *(alleen objecten)*
- Operators blocks
 - add
 - subtract
 - multiply
 - divide
 - pick random ... to ... *(inclusive!)*
 - greater than
 - less than
 - equal
 - ... and ...
 - ... or ...
 - not ...
 - join
 - letter ... of ...
 - length of ...
 - ... contains ... ?
 - mod
 - round
 - [...] of ... *(wiskundige functies)*
- Variables blocks *(variabelen en lijsten)*
 - set ... to ...

- change ... by ...
- add ... to [...]
- delete ... of [...]
- delete all of [...]
- insert ... at ... of [...]
- replace item ... of [...] with ...
- item ... of [...]
- item # of ... in [...]
- length of [...]
- [...] contains ... ?
- My Blocks
 - Eigen blocks worden ondersteund en 'inline' uitgevoerd. Eigen blocks kunnen alle ondersteunde blocks bevatten, inclusief calls naar (andere) eigen blocks (incl. recursie)

Blocks die niet geïmplementeerd zijn, kunnen niet gebruikt worden, deze zullen een foutmelding geven tijdens de conversie om te voorkomen dat niet ondersteunde zaken op het FuzeField terechtkomen.

Tips en tricks

- De stippen van het FuzeField staan in een 2x2 opstelling, daardoor zijn ze minder intuïtief te gebruiken als 4 pijltjes toetsen. Logischere opties zijn:
 - 2x een up-down combi voor twee elementen (zoals bij spel "pong")
 - Diagonale bewegingen
 - Geel en blauw gebruiken voor beweging naar links en rechts, de andere stippen gebruiken voor aanvullende acties
- De broadcast van een bericht kent geen volgorde in afhandeling, zorg dat "when I receive" blokken niet afhankelijk zijn van elkaar
- Afbeeldingen als bitmap geven het beste conversie resultaat, vector wordt geconverteerd tijdens de import maar kan soms afwijken. Zorg tevens dat afbeeldingen gecentreerd zijn rond het "nulpunt", op het FuzeField worden costumes gecentreerd toegepast.

Aandachtspunten

Naast het niet kunnen gebruiken van enkele niet geïmplementeerde blocks zijn er nog een paar aandachtspunten:

- Combineren van event “when ... key pressed” en sensing “key ... pressed” op dezelfde toets is niet mogelijk doordat de eerste die voorbij komt de toets zal afvinken als ‘gezien’. Ditzelfde geldt voor twee gelijktijdig lopende stukken code die beide op “key ... pressed” controleren, dan zal de eerste de toets afvinken en de andere code de toets dus niet zien.
- Hou er rekening mee dat “key pressed” kortstondig is, ingedrukt houden bestaat niet op het FuzeField. Een manier om dit te simuleren in je Scratch spel is door direct na de controle, te wachten totdat die knop niet meer ingedrukt is:



- Variabelen worden niet getoond binnen het FuzeField behalve de variabelen die door het FuzeField worden uitgelezen en buiten het speelveld getoond worden.
- Variabelen zijn niet case-sensitive. Een variabele met naam “TIMER” is dezelfde variabele als “timer”.



Geavanceerd

Schermgrootte

Naar de toekomst toe is te voorzien dat de volledige breedte van het scherm gebruikt kan gaan worden, om hierop voor te sorteren kan je in plaats van harde afmetingen gebruik maken van variabelen “screenwidth” en “screenheight” (standaard te vullen met respectievelijk 480 en 360 in je Scratch project).