Ruimtelijke Tour 1.1

Doel

Activeren voorkennis

Inleveren

Link naar je github repository voor dit project. Let op: Maak pas een repository als je in een groepje gaat werken.

In de github een readme MET:

een link naar een werkende tour.

Opdrachtomschrijving

Maak m.b.v. Agile methode een prototype van een tour binnen de school. Agile houdt in dit geval dat er veel iteraties zijn, d.w.z. meerdere momenten waarop je iets klaar moet hebben.

Per sprint worden hieronder tips gegeven over hoe je het kan doen, maar je bent vrij om het geheel anders, of met een andere technologie te doen. Het opgeleverde moet wel overeenkomen met de **dikgedrukte** omschrijving.

Doe eerste paar sprints doe je alleen, later ga je samenwerken. Je kan dan het werk verdelen.

Aan het eind van sprint 7 lever je per groepje een link in naar je github. In de readme beschrijf je wie er meedoet EN waar de live versie staat!

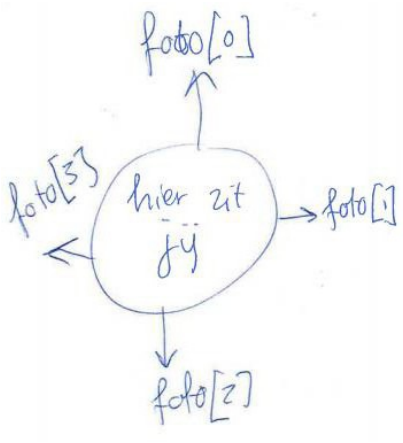
Aan de slag

Individueel

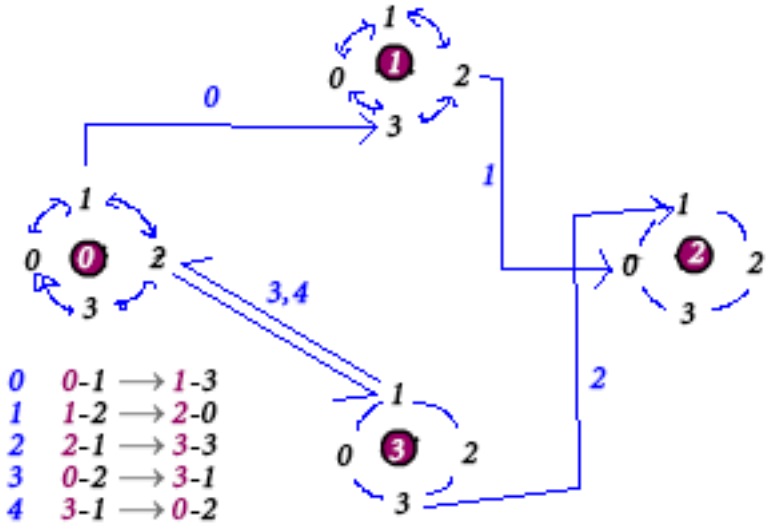
Sprint0: **Voorbereiding en test-foto's**

maken Plekken (nodes) & Connecties,

zie voorbeeldschets 1



- Maak foto's van 4 richtingen vanaf je huidige locatie. Dit Deze verzameling foto's noemen we een "plek" of node.
- De foto's staan allemaal voor een richting waarin je kan kijken vanaf deze plek.
- Denk alvast aan goede naamgeving van je foto's (bv. img0_0.jpg, img0_1.jpg,…) (of gebruik directories/mappen)



Sprint 1: **Maak een webpagina en laat daarin 1 foto zien, zo groot mogelijk.**

- Maak een html-bestand aan
- Iets met <img src ?
- css: maak de foto zo groot mogelijk, zonder het aspect ratio te veranderen.
    - backgroundsize-property contain..
    - object-fit... http://jsfiddle.net/danield770/n8qxcp4b/1/

Sprint 2: **Display functie**

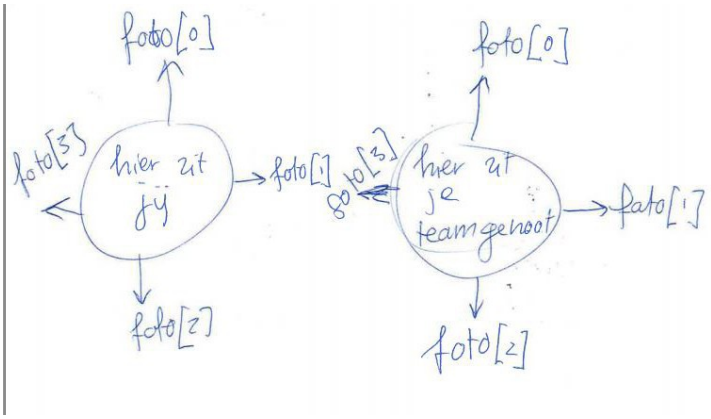
Maak een javascript functie die een image displayed.

- `document.getElementById("main_view").src = ?`

Sprint 3: **Maak de eerste node/plek werkend**

- Zorg nu dat je op de node “rond kan kijken”:
- Voeg pijltjes naar links en naar recht toe en zorg dat je van foto0 naar foto1 gaat, etc, als je je op rechts drukt…en andersom.
- Doe dit door middel van arrays (in JavaScript).
- Iets met
- `addEventListener op de pijltjes?`

Sprint 4: **Maak foto's van een aansluitende node**

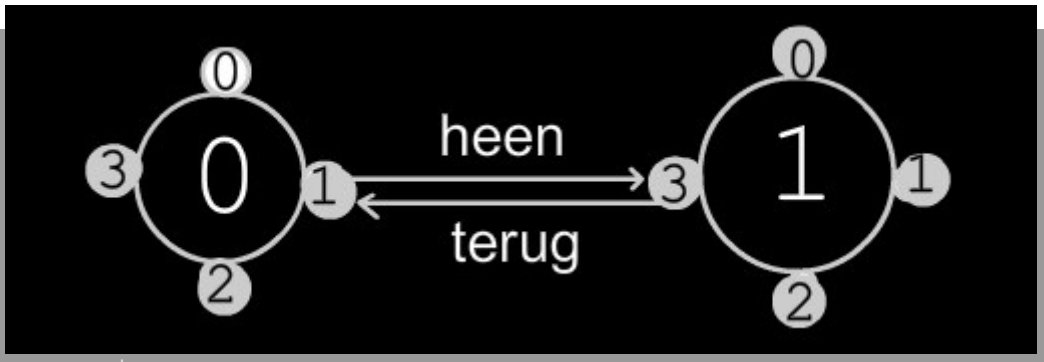
- Maak vanaf het verlengde 1 van de richtingen (een nieuwe node) ook 4 fotos's. Misschien heeft 1 van je collega's die al…
- 

Zorg dat je tour-applicatie kan wisselen tussen twee nodes/plekken met een klik. De kijkrichting moet hetzelfde blijven!

Sprint 5: **Voeg de connectie toe in code**

- Zorg nu dat je door kan klikken naar een volgende node. Maak een nieuwe JavaScript functie, waarbij als je *op de foto* klikt, je de juiste foto van de volgende node laat zien.
    - Iets met onclick/addeventlistener op een foto?

    Of wellicht een forward button? (Hoe deed je ook weer een mobile friendly button?)



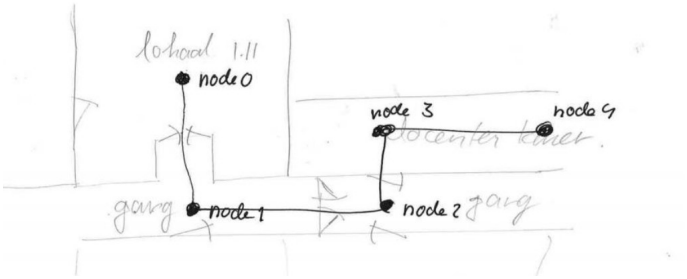

```
var connections=
[
      {
            from:{node: 0, dir: 1},
            to:{node: 0, dir: 1}
      },
      {
            from:{node: 1, dir: 3},
            to:{node: 0, dir: 3}
      }
];
```

Sprint 6: **Voeg de connectie toe in code**

- Voeg een interface toe waarmee, je links, rechts, vooruit en achteruit kan.
- Klik nu heen en weer door de twee of meer nodes/plekken,
- de kijkrichting moet hetzelfde blijven bij veranderen van plek..

Samenwerken

Nu heb je een framework om je echte tour te gaan maken. We gaan verder in groepjes van maximaal 4.



Sprint 7: **Schets nu een plattegrond van je echte tour(zie voorbeeld), maak vervolgens alle foto's op alle nodes**

- Schets plattegrond van deel van de school met een route en nodes
  - Maak op elke node 4 foto's van de 4 richtingen
- Zorg dat je alle nodes in een array hebt. Een nodes_array is dus een array van arrays met 4 waardes.

Sprint 8: **Voeg alle content (=foto's) en connecties toe**

- Bedenk van welke node je naar welke node kan lopen(= hetzelfde als van welke foto naar welke foto) en zet deze connecties in een connecties_array. Dat is dus een array van een array met 2 waardes (namelijk [fotos_van, foto_naar])

Je kan nu als het goed is van elke node naar elke node lopen en overal rondkijken…

Verbeter de UXD d.m.v toevoegen UI-elementen

- Zorg dat het zichtbaar is als een foto een connectie heeft (bv. d.m.v. een handje)
- Bekijk hiervoor eventueel andere online tours voor UI-ideeën.
- Iets met **D**ocument.getElementById("my_img").className = "clickable"?

Verbeter de UXD d.m.v. front-end verbeteringen.

- Maak de tour af door hem visueel aantrekkelijk te maken d.m.v. bv. CSS-animatie, een extra map, een inventaris, wat dan ook.

Sprint 8: **Test, test, test**

- Laat je tour testen door minstens 2 uit een andere groep. Breng verbeteringen aan.

Uitbreidingen:

Ben je klaar?

- Maak dan je CSS responsive, zodat het ook op een mobiel werkt.
- Maak CSS overgangen met `transition:transform 0.2s ease-in-out;`
- Doe andere dingen dan alleen vooruit/achteruit, kun je een voorwerp manipuleren?
- Kijk of je filmpjes erin krijgt of animated gif, ipv alleen een image.

Technologieën:

HTML, CSS, JavaScript (evt. als je echt per se wilt.. PHP, JQuery)

Hoe

In groepjes van 4.

Hulp

Een opzet van de uiteindelijk code zou kunnen zijn:

```
var node0 = ["img0_1.jpg", "img0_2.jpg","img0_3.jpg",
"img0_4.jpg"];
var node1 = ["img1_1.jpg", "img1_2.jpg","img1_3.jpg",
"img1_4.jpg"];
….
```

var node_array = [node0, node1,…..];

```
var connections = [ {node: 0, dir:1},{node: 1, dir:1}], // heen
    [ {node: 1, dir:3},{node: 0, dir:3}], // terug
…..
]
```

<script>

```
function change_image(direction) {
…
}

function getConnection() {

}
</script>

<body>

<div>
…<img onclick=getConnection(); …..

…<img id="left" onclick=change_image("previous"); ….

…<img id="right" onclick=change_image("next") …
</div>
```

CSS hulp:
https://jsfiddle.net/ya6f7xwa/