

# Code Design

Herkansingsopdrach: Uitbreidingen op Indiana Jones

## Samenvatting

<b>Wat</b>	C# Console applicatie
<b>Wie</b>	Implementeer de nieuwe requirements individueel.
<b>Deadline</b>	Zie brightspace
<b>Inleveren</b>	Presentatiedocument + Code op Brightspace Code dient tevens op Github Classrooms te staan.
<b>Toetsing</b>	(Digitaal) Assessment. Voor moment zie Brightspace



[https://en.wikipedia.org/wiki/Indiana\\_Jones\\_and\\_the\\_Last\\_Crusade](https://en.wikipedia.org/wiki/Indiana_Jones_and_the_Last_Crusade)

## Casus

In deze opdracht dien je individueel verder te ontwikkelen aan de code die je bij de eerste deelopdracht hebt gemaakt. In de eerste deelopdracht heb je een console applicatie gemaakt waarin je als Indiana Jones door een tempel heen loopt om de Holy Grail te bemachtigen. Verschillende obstakels moesten daar overwonnen worden voordat je bij de grail was.

Dit keer zijn er nieuwe uitdagingen toegevoegd aan het level en ook nieuwe wensen aan de applicatie. Jouw doel is om je applicatie te refactoren en de nieuwe requirements te realiseren.

## Opdracht

Deze opdracht dien je alleen te maken en te verdedigen in een assessment (dus zonder je partner van het eerste deel). Voordat je start met deze opdracht dien je...

- ...je eigen branch te maken, houd de volgende naamgeving aan:  
C\_{studentnaam}

**Let op:** We accepteren geen commits / pushes meer op de main-branch. We accepteren enkel commits van jezelf op je eigen branch.

### 1. Verbeteren op basis van feedback

Nadat je jouw eigen branch hebt gemaakt kan je de feedback op deelopdracht 1 gebruiken om je code te verbeteren. Deze verbeteringen worden meegenomen in de beoordeling van de eindopdracht.

## 2. Nieuwe vijanden

Bij de inleverlink op blackboard kan je een library vinden met gedrag voor vijanden in de tempel. Zorg er voor dat deze vijanden ook in jouw game voorkomen.

Gebruik daarvoor het nieuwe JSON bestand dat ook bij de inleverlink is gevoegd.

Laad de nieuwe JSON file in waarin zich vijanden bevinden.

- Gebruik de library die meegeleverd is op Brightspace. Aan het einde van dit document zie je een klassendiagram van de code in de library.
- Toon een vijand simpelweg door de letter **E** te plaatsen op de plek waar hij zich bevindt.
- Na elke stap die jij zet moeten ook alle vijanden lopen (Move methode).
- Als je tegen een vijand aan loopt krijg je schade.
- Door op spatie te drukken kan de speler Indiana Jones laten schieten.
  - o Elke vijand die op 1 afstand ten Noorden, Oosten, Zuiden of Westen van je staat verliest dan één van zijn 3 levens.
  - o Elke vijand zal na het 'schietincident' een stap doen (zonder dat Indiana Jones zich beweegt)

```
"enemies": [{  
  "type": "horizontal",  
  "x": 2,  
  "y": 5,  
  "minX": 2,  
  "minY": 5,  
  "maxX": 8,  
  "maxY": 5  
}]
```

De nieuwe JSON structuur is als volgt:

**Room-object**, nieuwe properties:

- **Enemies** (array), properties:
  - o **type** (string): Het type van de vijand;
  - o **x** (int): de start-x-locatie in de ruimte (0-based);
  - o **y** (int): de start-y-locatie in de ruimte (0-based);
  - o **minX** (int): de minimale x-locatie in de ruimte waarna hij omdraait (0-based);
  - o **maxX** (int): de maximale x-locatie in de ruimte waarna hij omdraait (0-based);
  - o **minY** (int): de minimale y-locatie in de ruimte waarna hij omdraait (0-based);
  - o **maxY** (int): de maximale y-locatie in de ruimte waarna hij omdraait (0-based);

**Tip:** Kijk nog eens naar het *Adapter patroon* om deze implementatie goed te laten verlopen.

### 3. Switch door in de tempel

Er is een nieuwe kamer aan het level toegevoegd waarin zich 2 pressure plates bevinden en een nieuw type deur (Switch door) die zich ontgrendeld/vergrendeld wanneer op alle pressure plates in dezelfde kamer iemand staat. Er bevindt zich een vijand in deze kamer die zich over beide pressure plates beweegt. Als de vijand op 1 van de pressure plates staat en Indiana Jones op de andere dan gaat (en blijft) de deur open. Het JSON-bestand is daarom uitgebreid met een nieuw type (switched) voor een Connection-object.

```
{
  "WEST": 4,

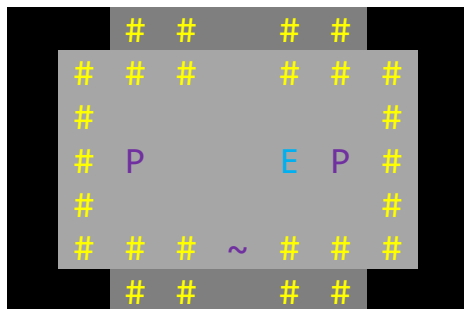
  "EAST": 7,
  "door": [{
    "type": "switched"
  }]
}
```

#### Type deuren

##### - “switched”

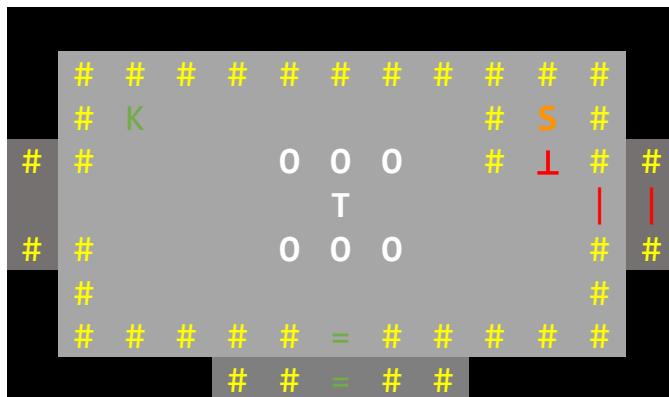
Initiële staat: Gesloten

*Interactie:* Opent enkel als alle pressure plates in dezelfde kamer gelijktijdig worden ingedrukt (en blijft dan open).



#### 4. Binnenkamers

Er kunnen zich nu ook kamers binnen kamers bevinden.



Het JSON-bestand is daarom uitgebreid **en aangepast** met de nieuwe functionaliteit:

**Room-object**, nieuwe properties:

- **specialFloorTiles** (array), de speciale vloertegels:
  - o **type** (string): het type vloer (in dit geval "wall" of "innerdoor")
  - o **x** (int): x-locatie in de ruimte (0-based);
  - o **y** (int): y-locatie in de ruimte (0-based);

```
{
  "specialFloorTiles": [
    {
      "type": "wall",
      "x": 3,
      "y": 3
    }, {
      "type": "innerdoor",
      "x": 4,
      "y": 3
    }
  ]
}
```

- **"wall"**

*Interactie:* Een speler kan niet op/door een binnenmuur bewegen.

- **"innerdoor"**

De positie van de inner door is gedefinieerd bij de kamer, zoals hierboven beschreven. Het gedrag van de deur staat echter bij de connections, waar ook alle andere deuren beschreven zijn. Er kan **altijd maar één innerdoor in een kamer** zijn, je kan hem matchen op basis van de nieuwe property **"within"**. En als laatste toevoeging hebben we bij alle deuren aangegeven of ze horizontaal of verticaal zijn.

*Interactie:* Een speler kan niet zich enkel door de deur heen bewegen als aan de voorwaarden van de deur voldaan is. Elke type deur die als Connection gedefinieerd is kan nu ook als 'binnendeur' gebruikt worden. In de JSON is dit een colored deur en een toggle deur maar ook andere type deuren moeten mogelijk zijn.

```
"connections": [
  {
    "within": 4,
    "horizontal": true,
    "doors": [
      {
        "type": "colored",
        "color": "red"
      }, {
        "type": "toggle"
      }
    ]
  }
]
```



## Bijlage: Klassendiagram van de vijanden-library

