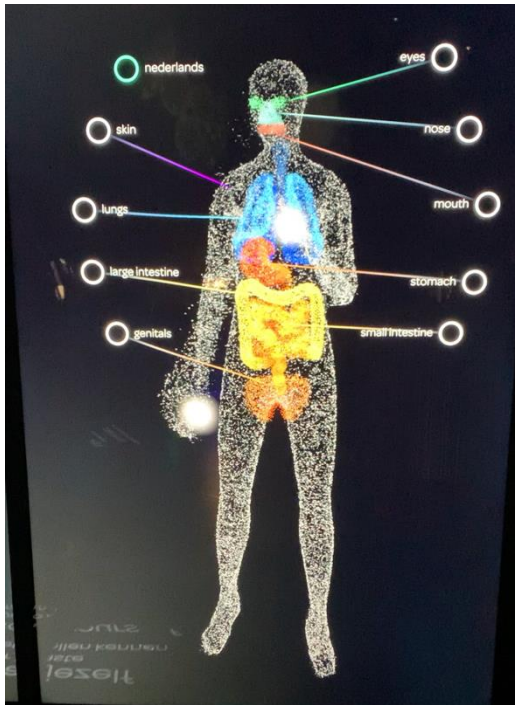


Vælg hvad du vil høre om

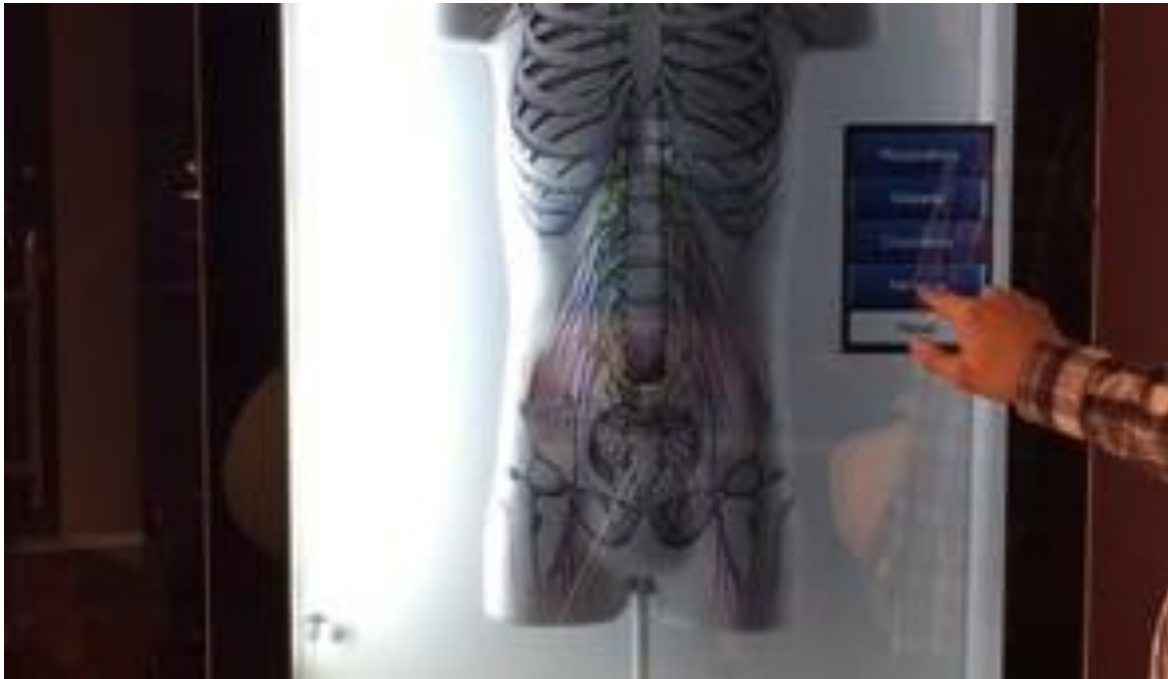
Sender visuelt ud og påvirker rummet ved hjælp af touchskærm

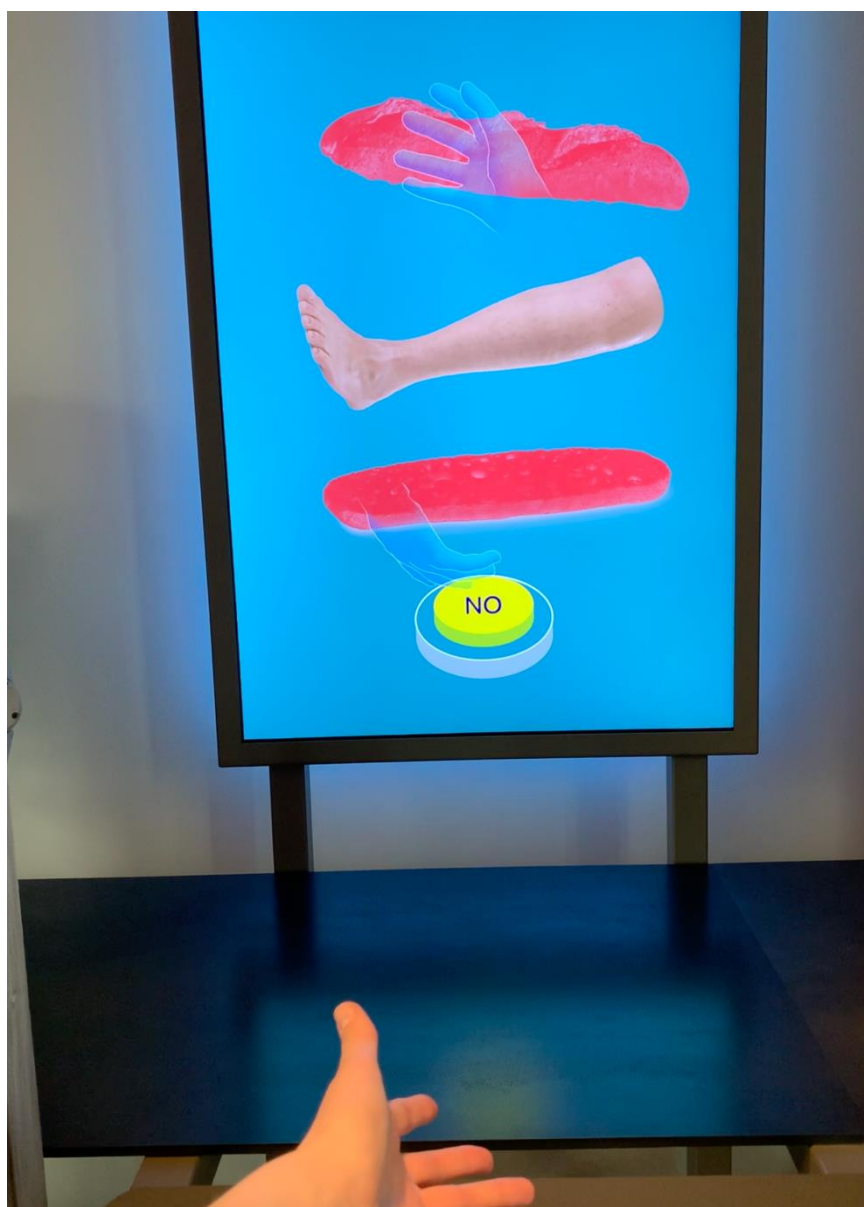


Din krop aflæses på skærmen, og hænder bruges til at vælge del af kroppen



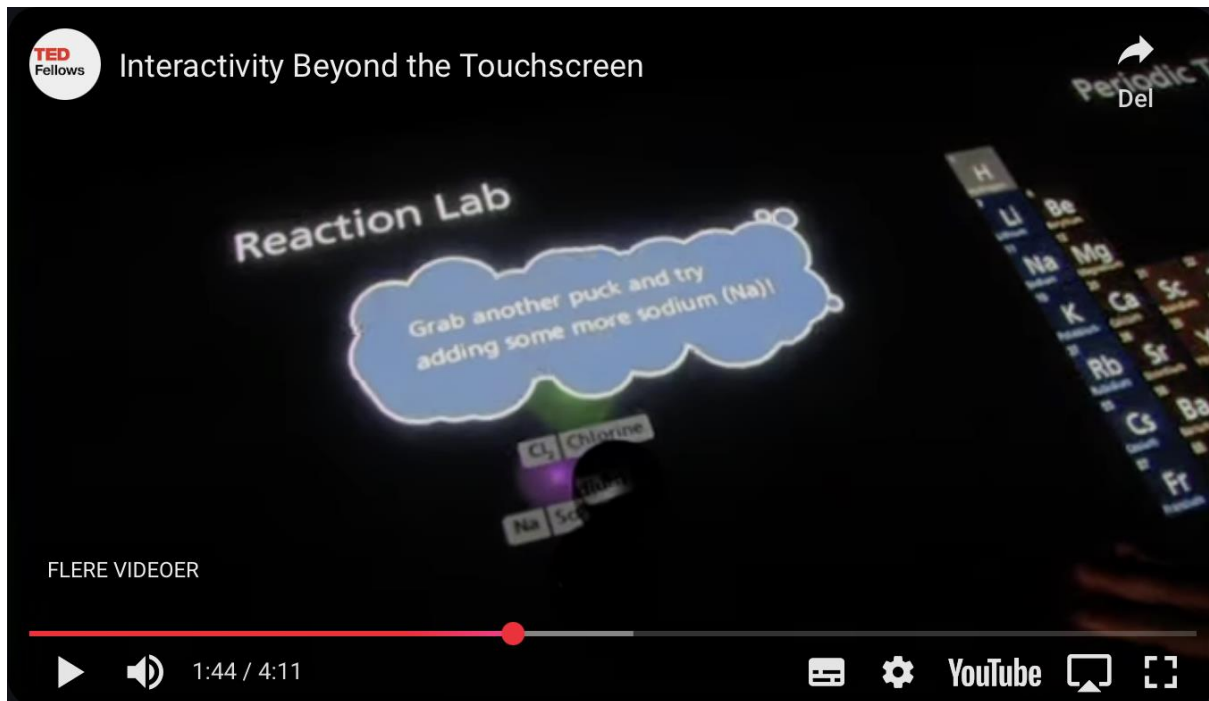
[Se website](#) - Kroppen





Hånd tager etiske beslutning (ja/nej)

Træk grundstofferne sammen på skærmen, og skab noget nyt



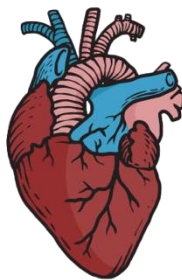
Interaktive kort eventuelt et der viser før og efter en begivenhed (atombombe eller andet) Eventuelt slider for glidende effekt



“Swipe” effekt med planeter, hvor man kan se f.eks. info om planterne og deres størrelse

Fase 2: Arbejdsdeling

Drag organer på plads i kroppen (tegnet skitse af kroppen)

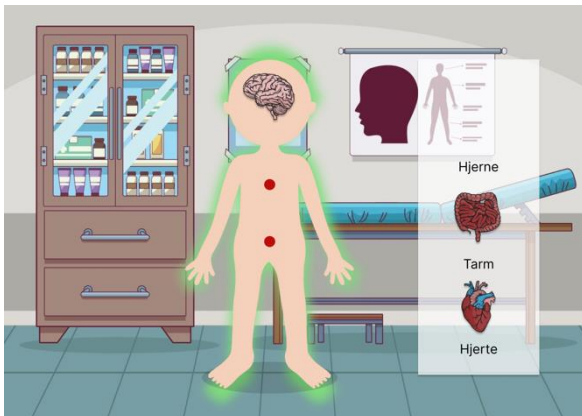


Problemformulering:

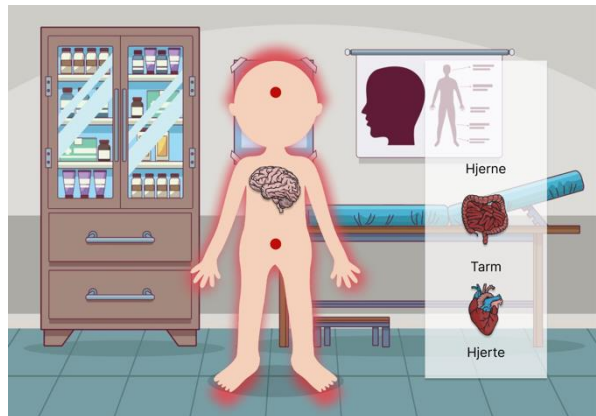
Vi vil gerne lave en slags “spil” hvor man skal trække organer hen på et menneske. Organerne inkluderer hjerne, hjerte og tarm. Når man trækker et organ hen til det rigtige sted, f.eks. hjernen til hovedet, vil kroppen, vil der komme en slags confirmation-besked, over at man har gjort det korrekt. Hvis man derimod trækker et organ hen til et forkert sted, f.eks. hjertet til hovedet, vil man få en “fejlmeddelelse”, og blive bedt om at prøve igen. Confirmation-besked og fejlmeddelelse skal understøttes med lyd.

Rapport

Som et interaktivt element har vi valgt at lave et ‘puslespil’ med kroppens organer. Brugeren skal trække de forskellige organer hen til deres rette placering. Når brugeren slipper et organ i en dropzone, responderer sitet enten med en bekræftelse i form af grønt lys og en bekræftende klokke eller en fejlmeddelelse med rødt lys og en buzzer (figur 1 & 2).

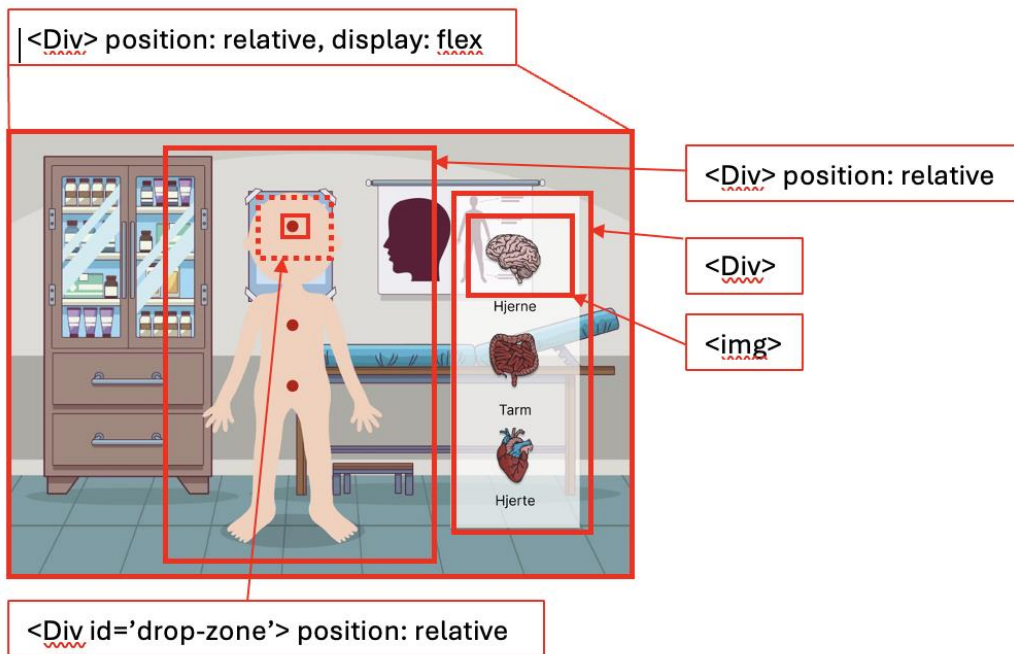


Figur 1: Korrekt gæt



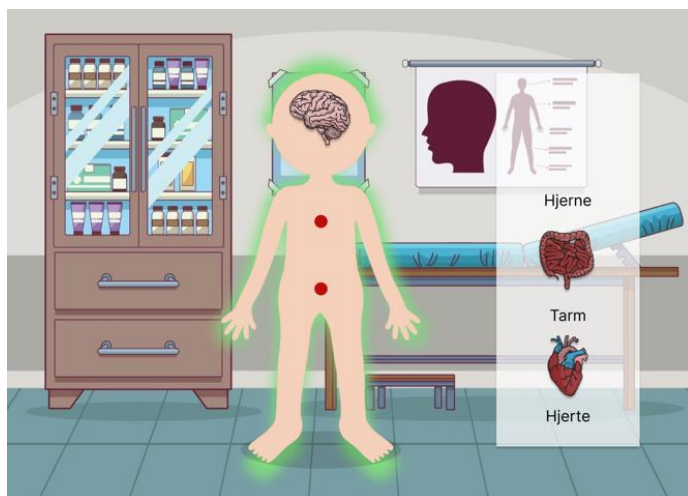
Figur 2: Forkert gæt

Sitet er i HTML bygget op af div'er, med henholdsvis position relative og position absolute. Div'erne med id'et 'drop-zone' har positionen absolute, for at vi har fuld kontrol over, hvordan de placeres på kroppen. For at denne absolute position forholder sig til billedet af kroppen, skal dens parent have position relative. Derfor har div'en med kroppen position relative. (se figur 3)



Figur 3: Opbygning i HTML og CSS

Drag-and-drop featuren



Drag-and-drop-funktionen gør det muligt for brugeren at klikke på et organ, trække det med musen og slippe det på en drop-zone på sitet. Featuren er lavet i JavaScript. Når brugeren tager fat i et organ, registreres det med 'dragstart', hvor ID'et på organet gemmes med `dataTransfer`.

```

10  organs.forEach(organ => {
11    organ.addEventListener('dragstart', e => {
12      e.dataTransfer.setData('text/plain', organ.id);
13    });
14  });

```

Figur x, script.js linje 10-14

Når organet holdes hen over en drop-zone, aktiveres dragover, hvor preventDefault() gør det muligt at slippe organet.

```
16     dropZones.forEach(zone => {
17         zone.addEventListener('dragover', e => {
18             e.preventDefault();
19         });
```

Figur x, script.js linje 16-19

Når organet slippes, aktiveres drop-eventet, hvor e.dataTransfer.getData('text/plain') henter ID'et på det trukne element. Herefter sammenlignes organElement.dataset.organ med zone.dataset.organ for at tjekke om organet er korrekt placeret.

Hvis det matcher, bliver elementet flyttet til dropzonen med zone.appendChild(organElement), lyden successSound.play() afspilles, og animationen korrekt.style.animation = 'lys 1s' viser et grønt lysglimt. Hvis det ikke matcher, afspilles errorSound.play() og animationen forkert.style.animation = 'lys 1s' viser et rødt lysglimt.

```
21     zone.addEventListener('drop', e => {
22         e.preventDefault();
23         const draggedOrganId = e.dataTransfer.getData('text/plain');
24         const organElement = document.getElementById(draggedOrganId);
25         const correctOrgan = zone.dataset.organ;
26
27         if (organElement.dataset.organ === correctOrgan) {
28             successSound.play();
29             zone.appendChild(organElement); // 💡 flyt organet ind i dropzonen
30             organElement.setAttribute('draggable', 'false');
31             organElement.classList.add('placed');
32             korrekt.style.animation = 'none';
33             void korrekt.offsetWidth; // Trigger reflow
34             korrekt.style.animation = 'lys 1s';
35
36             placedCount++; // ✅ tæl korrekt placering
37             if (placedCount === totalOrgans) {
38                 // 🎉 vis slut-popup når alle er på plads
39                 setTimeout(() => {
40                     document.getElementById('slutPopup').style.display = "block";
41                     document.getElementById('overlay').style.display = "block";
42                 }, 0);
43             }
```


Pop-up feature

Vi har en startskærm der dukker op inden spillet begynder.



Den har vi lavet ved hjælp af “setTimeout”, som gør at den vises et halvt sekund efter man kommer ind på siden.

```
setTimeout(function () {  
  document.getElementById('startPopup').style.display = "block";  
  document.getElementById('overlay').style.display = "block";  
}, 500);
```

Funktionen tager fat i “boksen”, ved hjælp af getElementById, som tager fat i div’ens id ‘startPopup’. Herefter ændres dens CSS fra display = none til display block, så startskærmen vises.

```
function closePopup() {  
    document.getElementById('startPopup').style.display = "none";  
    document.getElementById('overlay').style.display = "none";  
}
```

Herefter har vi en funktion som gør, at startskærmen fjernes når man trykker på “start”. Her tager funktionen igen fat i div’en med id’et ‘startPopup’ ved hjælp af getElementById, og ændrer CSS’en tilbage til none.