

Empowering Creative Thinking Through Programming

Les 11: Collision & animation

Wat gaan we doen

Raak of mis

Animatie (lijst met images) afbeelden

Feitelijk if-statements 2.0 en arrays 2.0

Programmeren =

- Niet (alleen) toevoegen van nieuwe functionaliteit
 - “probleem” kunnen omzetten in code
- herkennen wanneer je voor een probleem al een keer eerder een code oplossing hebt geschreven
 - hergebruiken (delen van) code

Raak of Mis

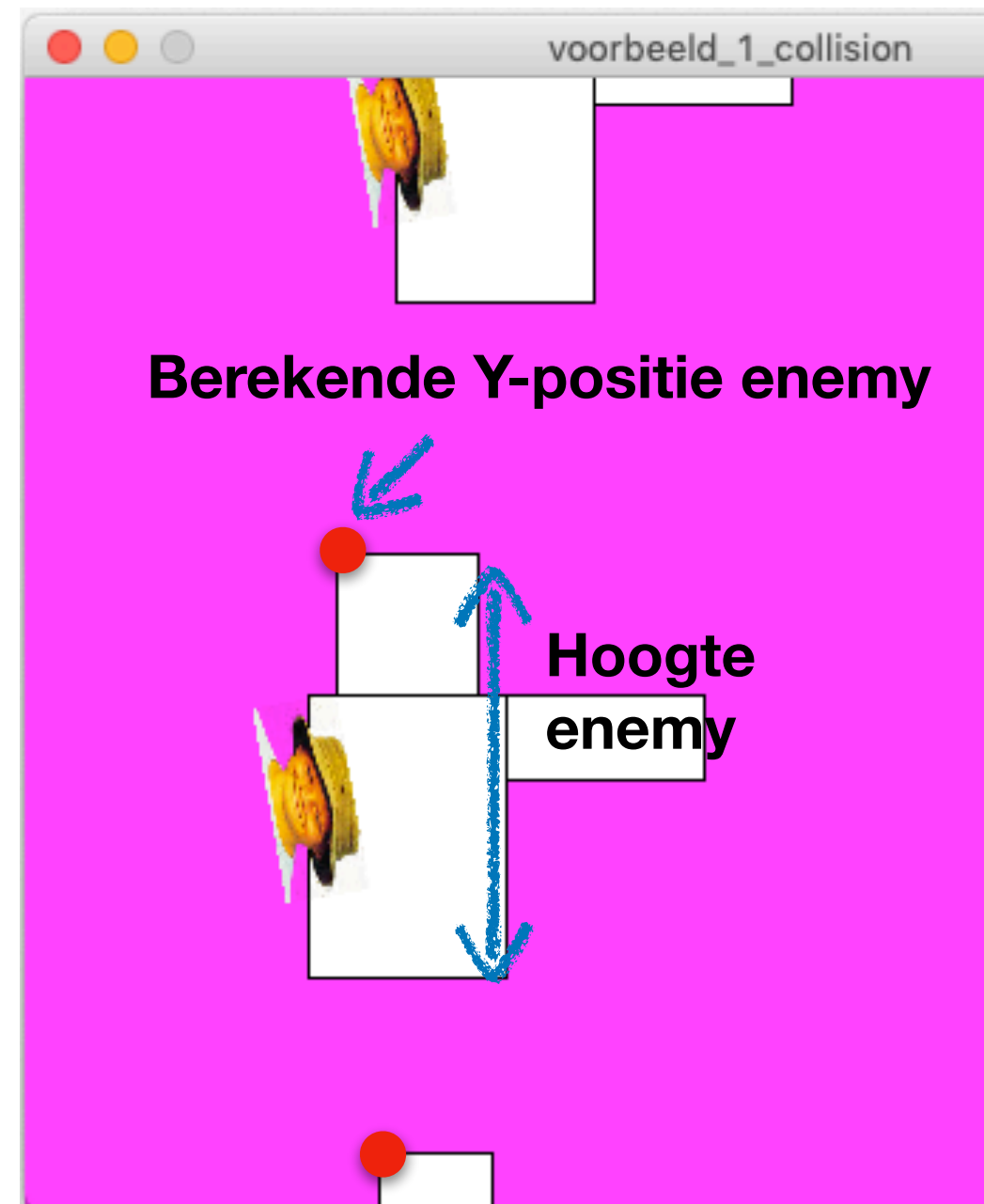
- Meer dan de helft vorige week gedaan
 - Positie enemy bepaald
 - Raak/mis moment bepaald
- Deze week
 - positie van de player bepalen
 - afstand enemy-player bepalen
 - Afstand + visuele breedte = raak/mis

Berekend vs Visueel

Rekening houden met verschil tussen berekende y-positie en visuele y-positie.

```
int benIkAlBeneden() {  
    if (y_positie >= height) {  
        return x_positie;  
    } else {  
        return 0;  
    }  
}
```

Visuele 'aankomst':
 $\text{height} - \text{hoogte enemy}$



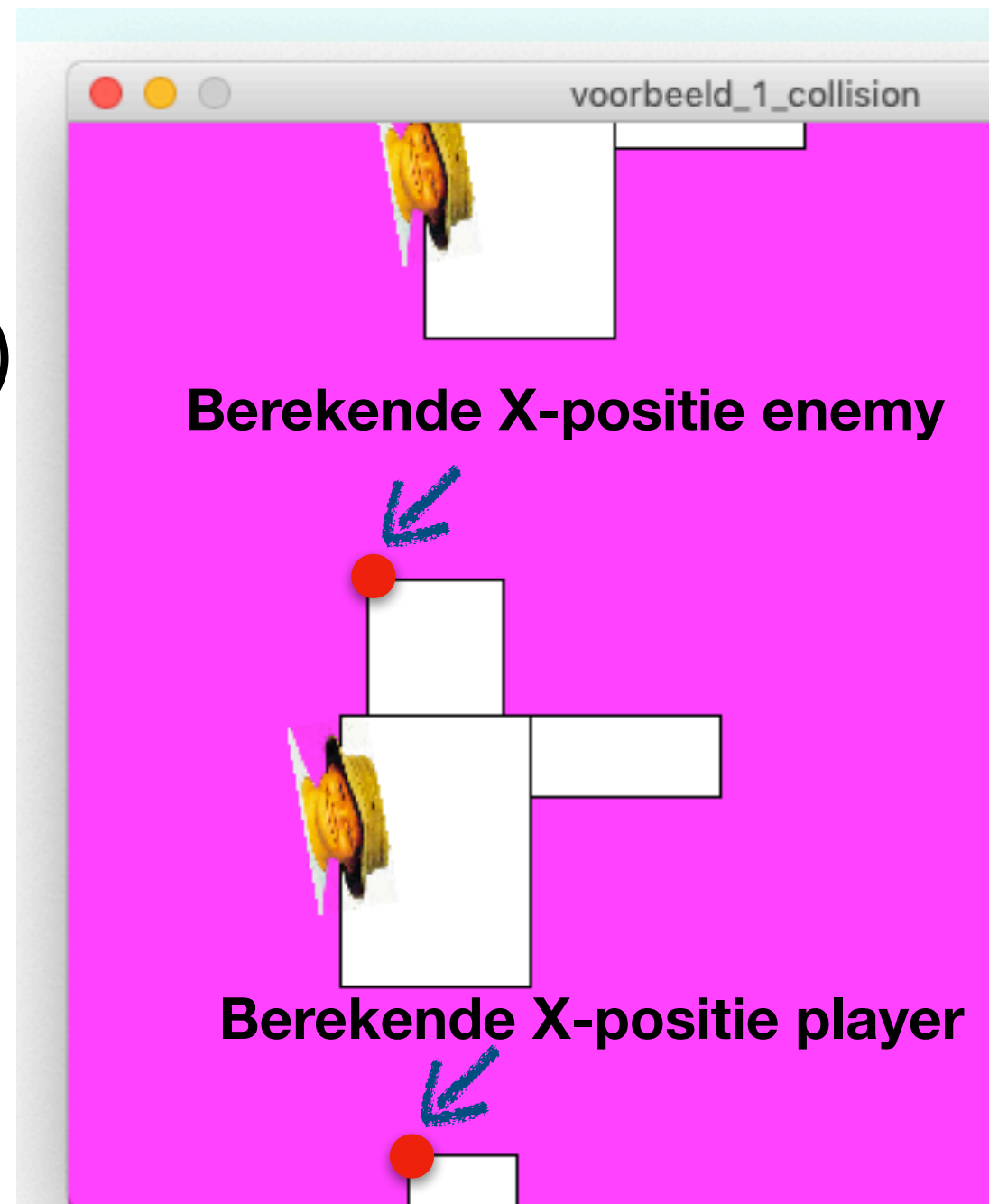
Berekend vs Visueel

Rekening houden met verschil tussen berekende x-posities en visuele x-posities van enemy en player.

Dus een simpele

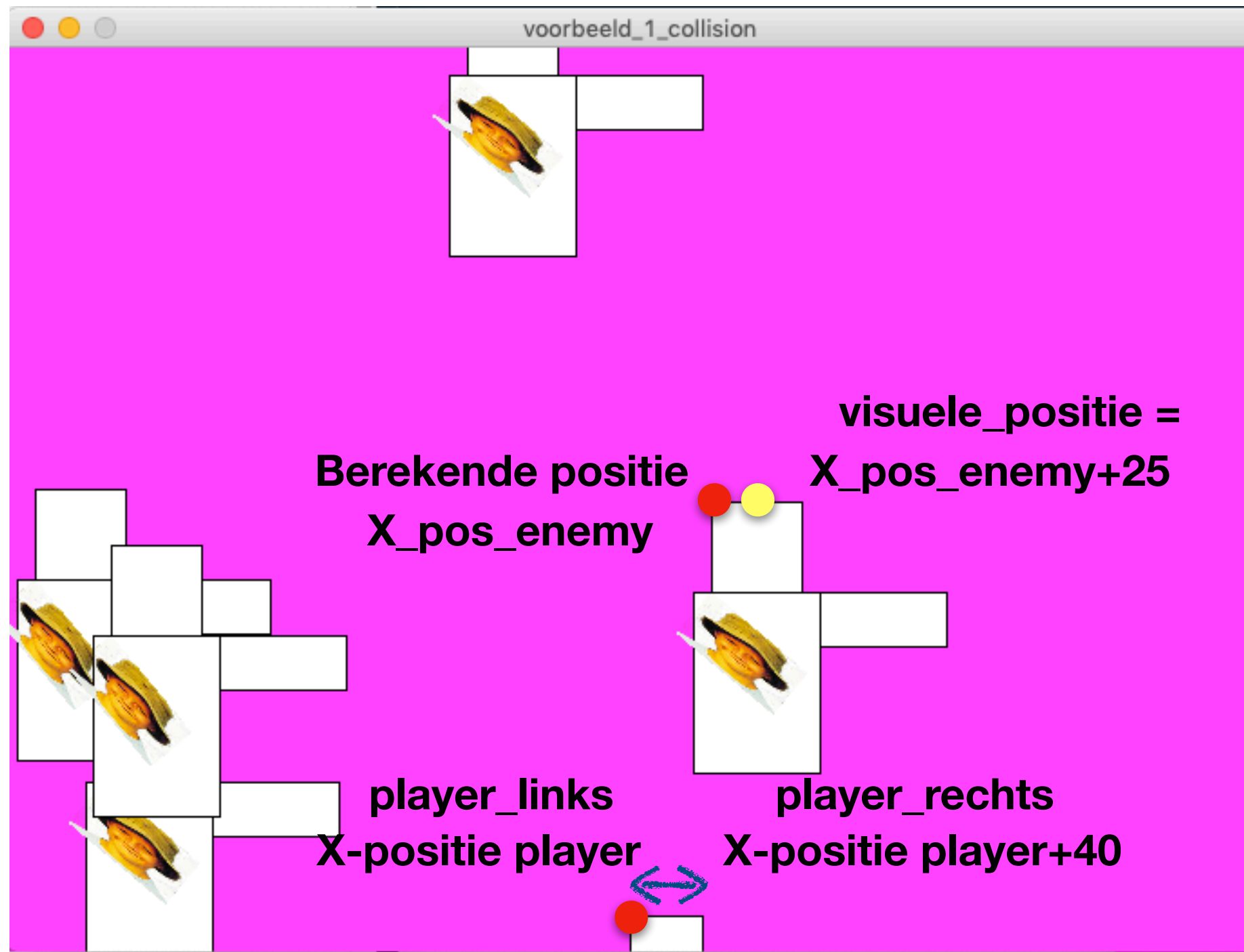
If ($x_pos_enemy - x_pos_player < 10$)

gaat hier niet werken



Omrekenen

Visuele midden enemy vergelijken met linker- en rechterkant player

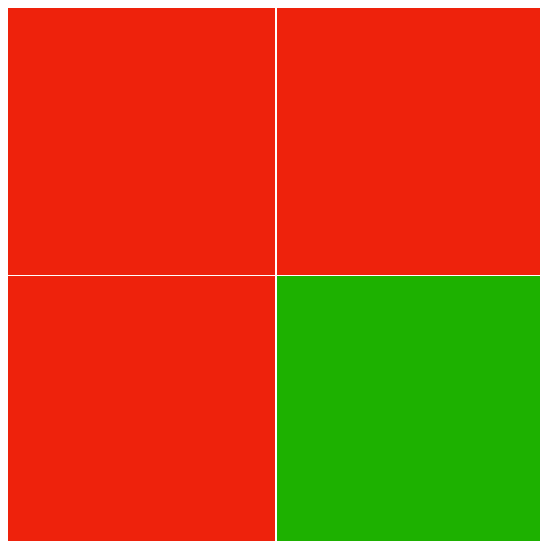


Combi-if statement

- Voorwaarden if statement combineren
- && (and) voor allebei de voorwaarden
- || (or) voor 1 van de voorwaarden

If (x_pos>100 && y_pos>100)

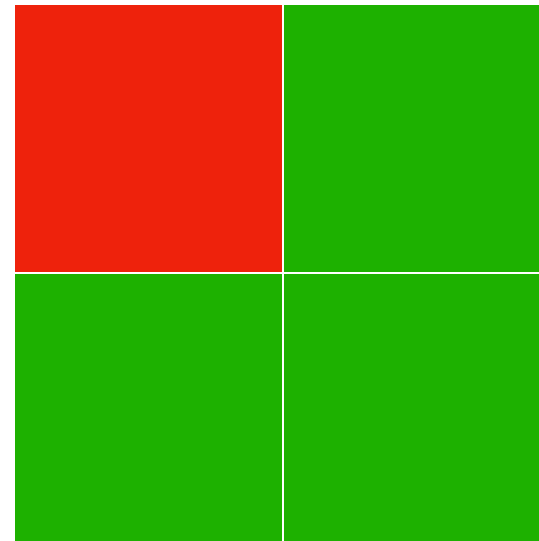
0,0



200,200

If (x_pos>100 || y_pos>100)

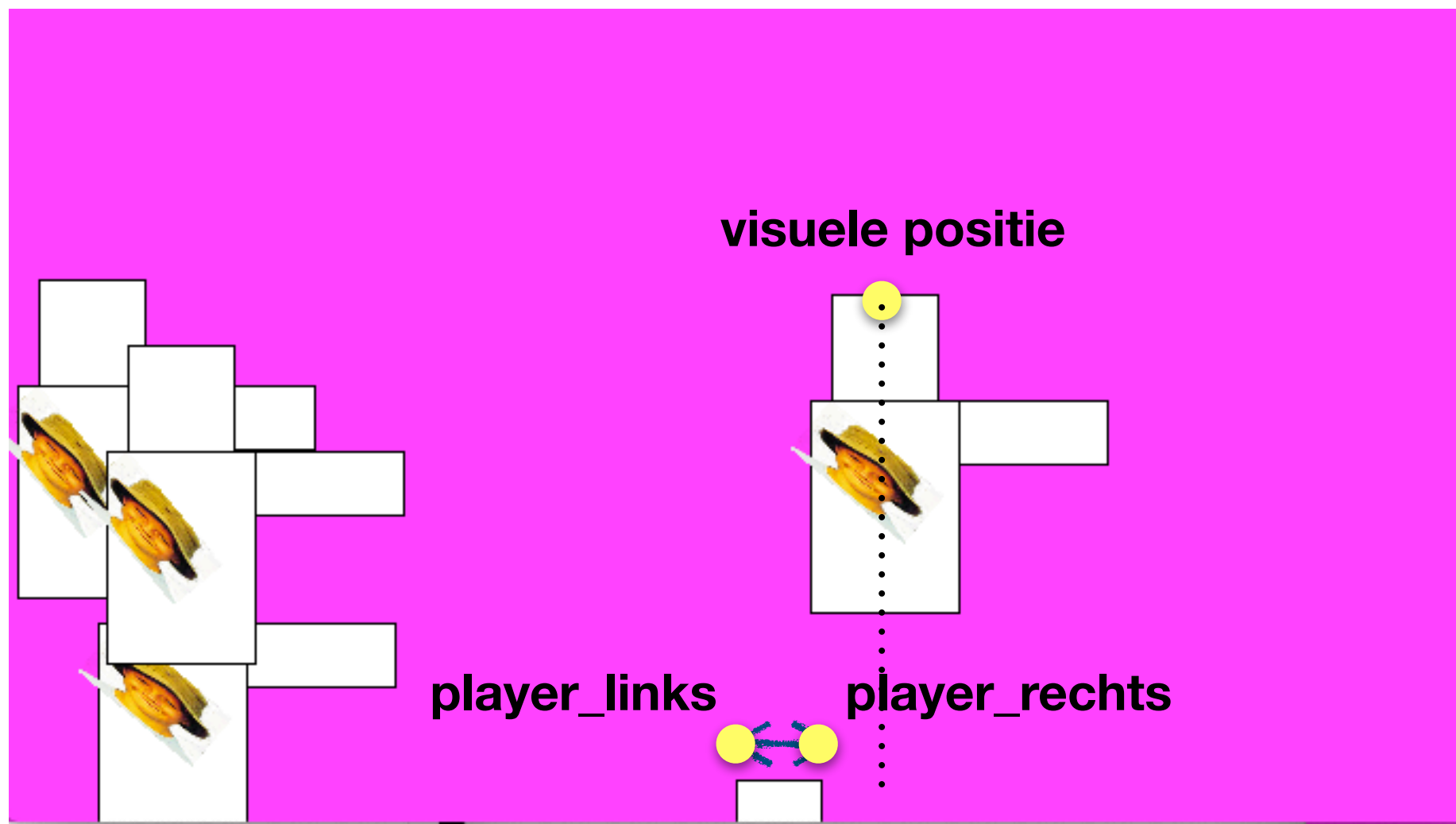
0,0



200,200

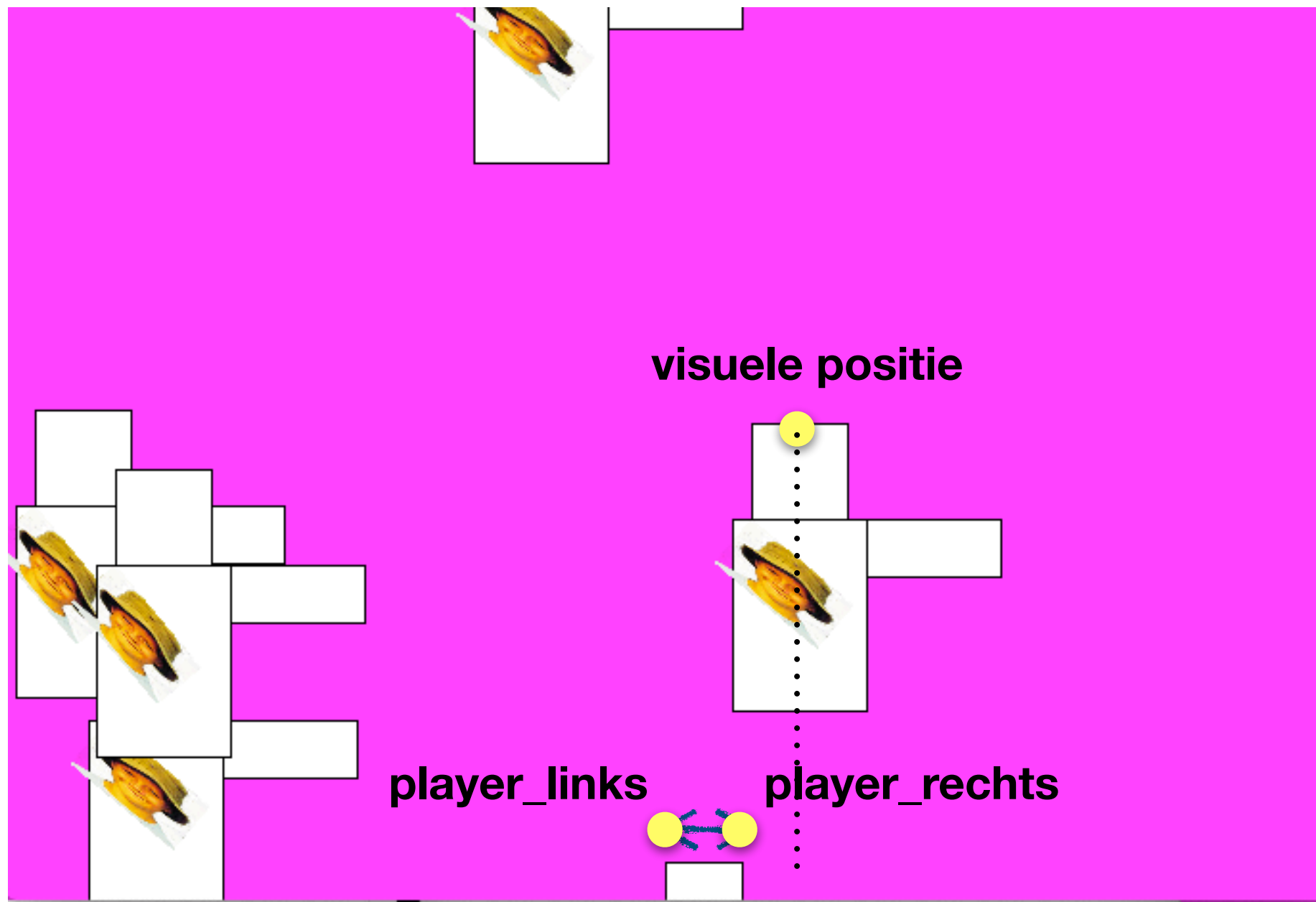
Raak...

- Als de visuele positie van de enemy tussen de linker en de rechterkant van de player valt



Raak...

if (visuele_positie > player_links
&&
visuele_positie < player_rechts)



Raak...

if (visuele_positie > player_links
&&
visuele_positie < player_rechts)

```
13 background(255, 0, 255); //Geef het venster een achtergrondkleur
14 for (int i=0; i<enemies.size(); i=i+1) {
15     Stickman tijdelijkeStickman = enemies.get(i);
16     tijdelijkeStickman.beweegStickman();
17     int alBeneden = tijdelijkeStickman.benIkAlBeneden();
18     if (alBeneden!=0) { // collision moment
19         int visuelle_positie = alBeneden+25; // omzetten berekend naar visueel
20         int player_links = mouseX; // linkerkant player
21         int player_rechts = mouseX+40; // rechterkant player
22         enemies.remove(i); // verwijder enemy (moet altijd, raak of mis
23         if (visuele_positie > player_links && visuelle_positie < player_rechts) {
24             println("raak"); // verhoog de score met 1 punt
25         } else {
26             println("mis"); // verhoog de score met 1 punt
27         }
28     }
29 }
30
31
```

Animatie (images)

- Animatie is deels te programmeren (move, rotation)
- maar als het te complex wordt is het handiger een frame-by-frame animatie te laten zien.
 - Stappen:
 - lijst met images laden (begin van programma)
 - images opslaan in array[]
 - images 1 voor 1 laten zien door door de array heen te lopen

(deze stappen komen je vast wel ergens bekend van voor)

Array[]

- Meer smaken dan alleen arrayList()
 - arrayList voor vaak dingen toevoegen/verwijderen
 - Deze versie:
 - 1 keer images toevoegen
 - voor de rest alleen uitlezen
 - 1 image laden met PImage
 - lijst images laden met PImage[]
- (de vierkante haken geven aan dat het een array is)

Stappen array maken en vullen

voorbeeld_1_animatie

```
1
2 PImage[] explosie_plaatjes; // stap 1: ruimte aanmaken en naam geven
3 int hoeveel_plaatjes = 90; // aantal plaatjes
4
5 void setup() {
6     explosie_plaatjes = new PImage[hoeveel_plaatjes]; // stap 2: ruimte vullen met array en grootte array
7     for (int i=0; i<hoeveel_plaatjes; i++) { // stap 3: array vullen met plaatjes
8         explosie_plaatjes[i] = loadImage("explosie_plaatjes/ontploffing" + (i+1) + ".png");
9     }
10 }
11
12 void draw() {
```

Stappen images gebruiken

- 1 image per frame (of aantal frames) laten zien.
- globale variabele nodig die bijhoudt welke image zichtbaar is
 - juiste image uit array kiezen
 - image uit array laten zien

Stappen images gebruiken

.

```
1
2 PImage[] explosie_plaatjes; // stap 1: ruimte aanmaken en naam geven
3 int hoeveel_plaatjes = 90; // aantal plaatjes
4 int welkPlaatjeIsZichtbaar = 0;
5
6 void setup() {
7     size (500,500);
8     explosie_plaatjes = new PImage[hoeveel_plaatjes]; // stap 2: ruimte vullen m
9     for (int i=0; i<hoeveel_plaatjes; i++) { // stap 3: array vullen met plaatje
10         explosie_plaatjes[i] = loadImage("explosie_plaatjes/ontploffing" + (i+1)
11     }
12 }
13
14 void draw() {
15     background(255);
16     image(explosie_plaatjes[welkPlaatjeIsZichtbaar], mouseX, mouseY);
17     if (welkPlaatjeIsZichtbaar < hoeveel_plaatjes - 1) {
18         welkPlaatjeIsZichtbaar++;
19     } else {
20         welkPlaatjeIsZichtbaar = 0;
21     }
```


Lesopdracht

Maak een venster aan van 500 pixels breed.

Gebruik de explosie_plaatjes in de Github slides/les 11 map

Laad de images in de explosie_plaatjes folder om een explosie animatie in je sketch te laten zien

Beeld de explosie af op het moment dat de muispositie groter wordt dan 300

Beeld de explosie af op de y-lokatie die de muis heeft op het moment dat de x-lokatie van de muis groter wordt dan 300