

Proefwerk Javascript Processing vwo versie C

Dit proefwerk bestaat uit een theoretisch deel dat je op papier maakt en een praktisch deel dat je op de computer maakt. In totaal heb je 60 minuten de tijd. Je mag pas beginnen met het praktische deel als je jouw uitwerkingen van het theoretische deel hebt ingeleverd. Het is je eigen verantwoordelijkheid om de tijd goed te verdelen!

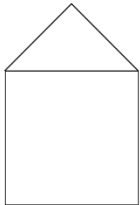
Het praktische deel maak je op de computer met behulp van het programma Notepad++. Hiermee hebben we in de les geoefend. Elke keer als jij vindt dat je een opdracht af hebt, laat je het resultaat zien aan de surveillant. Deze zal dat onderdeel dan beoordelen.

Zit je vast? Je mag vragen om een hint, maar dat kost je wel punten op je beoordeling!

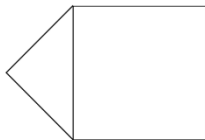
Is het tijd en heb je een opdracht nog niet af? Of wil je verder met een andere opdracht omdat je niet verder komt? Laat je werk dan wel beoordelen! Het kan zijn dat je wel een deel van de punten hebt behaald!

Theoretische deel (TOTAAL 6 • 5 = 30 punten)

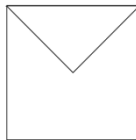
1. Bekijk de code in het blok hiernaast. Wat is het resultaat wanneer deze code wordt uitgevoerd? A, B, C of D?



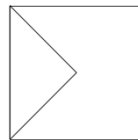
A



B



C



D

```
rect(0,0,200,200);  
triangle(0,0,0,200,100,100);
```

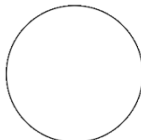
2. Bekijk de code in het blok hiernaast. Wat is het resultaat wanneer deze code wordt uitgevoerd? A, B, C of D?



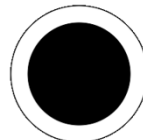
A



B



C



D

```
fill('black');  
ellipse(0,0,150,150);  
fill('white');  
ellipse(0,0,200,200);
```

3. Bekijk de code in het blok hiernaast.
Welke waarde heeft de variabele *sf* na uitvoering van de for-loop?
- A: 2 B: 2.5
C: 4 D: 5
E: 6 F: 11

```
var aantal=4;  
var sf=1;  
for (var a=1;a<=aantal;a++) {  
  sf=sf+a/2;  
}
```

De volgende vragen gaan over de programmeercode die horen bij het spel **Bommenvanger** waar je straks in het praktische deel mee bezig gaat. In dit spel vangt een speciaal vliegtuig bommen met zijn neus.

- In regel 19 (zie de broncode op de bijlage) staat: **yBom=random(10,470);** Leg uit wat deze regel doet. (Dat mag in algemene termen: je hoeft nog niet te snappen hoe het spel precies werkt. Dat ontdek je in het praktische deel.)
- In regel 37 staat: **noLoop();** Leg uit wat een *loop-functie* is en verklaar hiermee wat er gebeurt als deze regel wordt uitgevoerd.
- Stel dat we in regel 13 een variabele met de naam *naamPiloot* willen declareren die als beginwaarde jouw voornaam heeft. Geef de coderegel voor de declaratie van deze variabele.

Proefwerk Javascript Processing vwo versie C

Broncode Bommenvanger

```
8     var hoogte = 250;
9     var xBom = 850;
10    var yBom = 250;
11    var score = 0;
12    var snelheidBom = 2;
13
14    function setup() {
15        var myCanvas = createCanvas(850, 500);
16        myCanvas.parent('mijnDiv');
17        noStroke();
18        textSize(24);
19        yBom=random(10,470);
20    }
21
22    function draw() {
23        background('#ab3030');
24        fill('white');
25        text("SCORE: "+score, 720, 490);
26        if (keyIsDown(UP_ARROW)) {
27            hoogte = hoogte-3;
28        }
29        if (keyIsDown(DOWN_ARROW)) {
30            hoogte = hoogte+3;
31        }
32        tekenVliegtuig(hoogte);
33        tekenBom(xBom,yBom);
34        if (xBom==130 && yBom>hoogte-5 && yBom<hoogte+15) {
35            score++;
36            if (score==5) {
37                noLoop();
38                background('green');
39                textSize(60);
40                text("GEWONNEN", 250, 250);
41            }
42            snelheidBom++;
43            xBom=850;
44            yBom=random(10,470);
45        }
46        else {
47            xBom = xBom-snelheidBom;
48            if (xBom<-20) {
49                noLoop();
50                background('red');
51                textSize(60);
52                text("VERLOREN", 250, 250);
53            }
54        }
55    }
56
57    function tekenBom(xPositie,yPositie) {
58        fill('white');
59        rect(xPositie,yPositie,20,20);
60    }
61
62    function tekenVliegtuig(y) {
63        fill('black');
64        rect(50,y,50,30);
65        triangle(60,y,90,y,50,y-75);
66        triangle(60,y+30,90,y+30,50,y+105);
67        fill('blue');
68        triangle(100,y,100,y+30,130,y+15);
69        fill('yellow');
70        triangle(50,y,50,y+10,40,y+5);
71        triangle(50,y+10,50,y+20,40,y+15);
72        triangle(50,y+20,50,y+30,40,y+25);
73    }
```

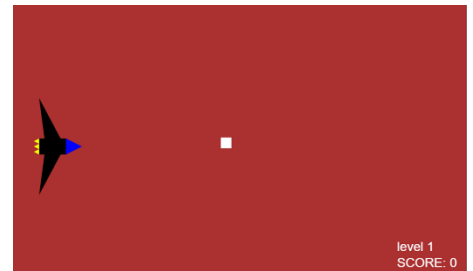
Proefwerk Javascript Processing vwo versie C

Praktisch deel (TOTAAL 60 punten)

Open het bestand *a4vcpd1.html* en bekijk de code. Dit is het spel **Bommenvanger**. Hierin bestuur je een vliegtuig omhoog en omlaag met de pijltjes van je toetsenbord. De bedoeling is om de witte bommen die vanaf de rechterkant verschijnen op te vangen met de blauwe punt van het vliegtuig. Als je een bom opvangt ga je naar het volgende level. Er komt dan een nieuwe bom op je af met een hogere snelheid. Wanneer je een bom mist ben je af. Als je 5 bommen weet te vangen heb je de vijand verslagen en heb je gewonnen. Veel speelplezier! (Maar denk aan je tijd...)

Hieronder staan een aantal opdrachten. Voer deze opdrachten uit (sla eventueel een stap over als je die niet weet):

- A. (10 punten) Rechtsonder in het canvas staat de score. Zorg dat boven deze score ook komt te staan in welk level je bent. Wanneer je begint moet er staan *level 1*, als er één bom gevangen is *level 2*, etc. Zie eventueel figuur 1.
- B. (15 punten) Level 2 is moeilijker dan level 1, maar levert op dit moment evenveel punten op. Pas de code aan, zodat je voor de eerste bom 1 punt aan je score krijgt toegevoegd, voor de 2^e bom twee punten, voor de 3^e bom 3 punten, etc.
LET OP: natuurlijk moet je nog steeds 5 bommen vangen om te winnen!
- C. (15 punten) Om het spel aantrekkelijker te maken voor de spelers, willen we de achtergrondkleur van het spel voor elk level willekeurig door de computer laten kiezen. De achtergrondkleur moet dus pas veranderen, wanneer er een bom is gevangen door onze bommenvanger. Pas de code aan, zodat aan deze eis is voldaan.
- D. (15 punten) We willen het spel aanpassen, zodat het lijkt alsof de piloot van ons vliegtuig *gas geeft* op het moment dat de pijl omhoog wordt ingedrukt. De hoogte moet dan met 2 pixels toenemen. Wanneer er niet op de pijl omhoog wordt gedrukt, moet het vliegtuig automatisch 1 pixel in hoogte dalen.
De pijl omlaag moet worden uitgeschakeld. Het spel wordt dus alleen nog bestuurd met de pijl omhoog.
Verander het spel zodat aan deze eisen wordt voldaan.



FIGUUR 1

EINDE van de toets