**Praktische opdracht programmeren Digger**

**Globale beschrijving:**

Je ontwerpt en bouwt samen het spelletje Digger.

**Doel van de opdracht:**

* Leren plannen en uitvoeren van een praktische opdracht.
* Leren samenwerken in het plannen en uitvoeren van een praktische opdracht.
* Inzicht verwerven in het proces van ontwerpen en bouwen van een programma.
* Ervaring opdoen met het werken met een moderne hogere programmeertaal.
* Leren “zelfstandig” methoden te leren kennen en gebruiken.

**Eindproducten:**

1. Logboeken van het gehele ontwerp- en bouwproces.
2. Alle solution-bestanden.

**Werkvorm:**

Deze opdracht moet worden gedaan in groepjes van 2 personen, niet meer, niet minder.

**Tijdplanning:**

De totale tijd die je voor dit project nodig hebt schat ik op 8 a 10 klokuren per persoon. Ik ga er dan wel van uit dat je het lesboek programmeren hebt doorgewerkt en redelijk beheerst.

**Omschrijving - probleemstelling**

Het spel toont een zijaanzicht van een stuk grond. In de grond bevinden zich diamanten. De held van het spel is de avatar Digger die door de grond graaft. Digger moet alle diamanten verzamelen door er over heen te bewegen. Digger wordt achtervolgd door zijn vijand. Deze begint te lopen 2 seconden nadat het spel is gestart. De vijand loopt iets sneller dan Digger. De vijand kan zich alleen verplaatsen door ruimtes die vrij zijn gegraven. Zodra de vijand Digger aanraakt verliest Digger een leven. Ook wanneer Digger wordt geraakt door een vallend blok verliest hij een leven. Een blok valt 1 seconde nadat de grond eronder is weggegraven. Tijdens het spel is zichtbaar hoe lang Digger al loopt. Als Digger alle diamanten heeft verzameld is hij klaar en wordt zijn eindtijd getoond. Als Digger al zijn 3 levens heeft verloren is het spel afgelopen. Er is een lijst van 5 toptijden zichtbaar, en wanneer de speler een toptijd behaalt, kunnen de naam van de speler en de tijd aan de lijst worden toegevoegd, als de speler dit wil. De gegevens van de slechtste speler in de toptijdenlijst verdwijnen dan. Deze toptijdenlijst wordt telkens opgeslagen wanneer het spel wordt afgesloten en weer ingelezen bij het starten van het spel.

**TIPS**

* Maak voor de verschillende event-handlers eerst PSD’s. Dit helpt. ECHT!!!
* Een werkend voorbeeld van wat ik in gedachten heb is beschikbaar op iedere willekeurige spelletjes site.
* Gebruik timers voor het laten rijden van de andere auto’s.

## Logboeken

Jullie houden individueel en als groep logboeken bij van je werkzaamheden. Gebruik hiervoor de (eventueel iets aangepaste) standaard-logboekformulieren. Deze logboeken moeten een goed inzicht geven van het gehele proces. De verzameling logboeken vormt onderdeel van het eindproduct en wordt betrokken bij de beoordeling van de opdracht.

**Nabespreking**

De opdracht wordt **NIET** afgesloten met een nabespreking. Je wordt beoordeeld op de ingeleverde producten.

**Beoordelingsmodel PO Straatrace**

* Logboeken geven een goed beeld van het proces 0 1 2
* Interface is “vanzelfsprekend” en aantrekkelijk. 0 1
* Besturing auto werkt goed. 0 1
* Botsing wordt goed verwerkt. 0 1
* Geluiden werken goed. 0 1
* Bijhouden toptijdenlijst werkt goed, ook opslaan. 0 1
* Voldoende commentaar zodat duidelijk is wat er gebeurt. 0 1
* Efficiënte, mooie en/of slimme implementatie 0 1 2

In principe krijgen leerlingen van dezelfde groep hetzelfde cijfer, tenzij er redenen zijn hiervan af te wijken.

Naam:

Totaal aantal punten:

Ingeleverd op:

Maximaal haalbare cijfer: 10 6 3

Eindcijfer: