

1. Phaser:

De enige kennis die ik al had over Phaser alvorens te gaan opzoeken hierover was dat het een HTML5 en JavaScript game framework is voor browser games mee te maken. Dit wist ik dankzij mijn stagebedrijf. Zij werken hier namelijk af en toe zelf mee in projecten maar dan meestal enkel voor animaties te implementeren in het project en niet om games te maken.

Dit is dan ook de reden waarom ik dit onderwerp wil onderzoeken. Op deze manier heb ik al een beetje kennis en ervaring met Phaser voor moest ik het ooit moeten toepassen in een project of als ik moet inspringen op een project waar Phaser al in gebruikt wordt.

Na wat onderzoekwerk kan ik zeggen dat Phaser dus een 2D game framework is dat gebruikt wordt om HTML5 games te maken voor desktop en mobile. Het gebruik van Phaser wordt wel afgeraden indien je een crossplatform game zou willen maken maar voor een 2D game die gebruik maakt van HTML is Phaser de beste optie. De documentatie van Phaser, die je kan vinden op de website phaser.io, is zeer uitgebreid en de website is voorzien van enorm veel voorbeelden met de bijhorende code wat het gebruik en leren van Phaser zoveel makkelijker en leuker maakt.

Phaser gebruikt zowel een Canvas als een WebGL renderer en kan ook automatisch wisselen tussen beide gebaseerd op browser support. Dit laat ook toe om de snelste laadtijden te bereiken. Daarnaast voorziet Phaser de mogelijkheid om verschillende types games te maken zoals *platformers*, *RPGs* and *Arcades*. Phaser komt met 3 physics systems namelijk *Arcade Physics*, *Ninja Physics* en *P2.JS* die allen een ander aspect op zich nemen. In Phaser kan je ook animaties gebruiken en voor geluiden af te spelen in jouw games kan je Web audio en HTML5 audio gebruiken.

Om af te sluiten heeft Phaser ook een plugin systeem ingebouwd zodat je plugins van de community kan gebruiken of je kan zelf plugins maken en deze dan delen met de community.

```
// Now let's create some ledges
platforms.create(600, 400, 'ground');
platforms.create(50, 250, 'ground');
platforms.create(750, 220, 'ground');

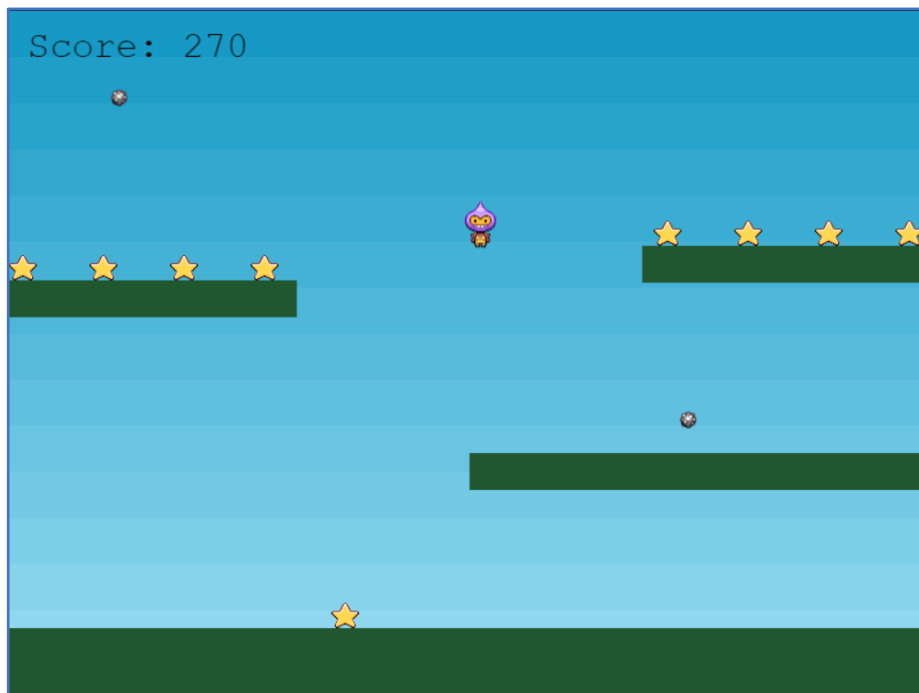
// The player and its settings
player = this.physics.add.sprite(100, 450, 'dude');

// Player physics properties. Give the little guy a slight bounce.
player.setBounce(0.2);
player.setCollideWorldBounds(true);

// Our player animations, turning, walking left and walking right.
this.anims.create({
  key: 'left',
  frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 }),
  frameRate: 10,
  repeat: -1
});

this.anims.create({
  key: 'turn',
  frames: [ { key: 'dude', frame: 4 } ],
  frameRate: 20
});
```

Code snippet waarin platformen en de speler gemaakt worden. Onderaan de code worden ook de animaties voor de speler toegevoegd.



Een voorbeeld van een game die je kan maken.