



## Vragen

1. Schrijf op wat de eerste stap is die je moet doen wanneer je een computerspel gaat ontwikkelen?  
Eerst moet je gaan nadenken over de vragen die te maken hebben met alle aspecten van het spel. Schrijf daarna een voorstel waarin wordt uitgelegd hoe het spel gaat werken.
2. Schrijf op wat een pixel is.  
Computer schermen worden opgebouwd uit individuele puntjes die pixels genoemd worden.
3. Leg uit hoe pixels gebruikt worden op een scherm.  
De scherm coördinaten bestaan uit een paar getallen die de positie van elke pixel op het scherm identificeren. De x-coördinaat geeft aan hoe ver van de linker zijde van het scherm de pixel is. De Y-coördinaat geeft aan hoe ver van de top van het scherm de pixel is.
4. Schrijf op wat de coördinaten zijn van de pixel in de linker boven hoek van het scherm.  
De coördinaten van die pixel zijn (0,0).
5. Schrijf op welk XNA object de eigenschappen bevat die de locatie van een pixel op het scherm beschrijft.  
De **Vector2** object bevat de X en Y waarden van de coördinaten.
6. Schrijf twee redenen op waarom een spel dat het volledige scherm gebruikt beter is dan een spel dat in een venster gespeeld wordt.
  - Door het volledige scherm te gebruiken heb je meer ruimte voor de animaties en de tekeningen.
  - De speler heeft minder kans om per ongeluk de focus van het spel weg te halen naar een ander venster.
  - Het gebruik van het volledige scherm zorgt ervoor dat de speler meer betrokken is in het spel.
7. Schrijf op welke method aangeroepen moet worden wanneer de preferred windows maten ingesteld zijn.  
Dan wordt aangeroepen *graphics.ApplyChanges()* method om de maat van het scherm daadwerkelijk aan te passen.
8. Schrijf op bij welke beeldformaten transparantie gebruikt kan worden.  
.PNG en .GIF
9. Schrijf op welk XNA object gebruikt wordt om een plaatje op te slaan.  
De Texture2D object bevat een verwijzing naar een geladen afbeelding bestand.
10. Schrijf op waar het SpriteBatch object voor gebruikt wordt.  
Het SpriteBatch object wordt gebruikt afbeeldingen "sprites" op het scherm te tekenen.
11. Schrijf op welke XNA klassen gebruikt worden toetsenbord, muis en gamecontroller invoer op te vragen.  
KeyboardState klasse, MouseState klasse en GamePadState klasse.



# XNA opdrachten



12. Schrijf op welke Sprite eigenschap de huidige positie van een Sprite bevat.  
De eigenschap `UpperLeft` bevat de huidige positie van een sprite.
13. Schrijf op hoe je een ingelezen afbeelding toekent aan een Sprite.  
Dit gebeurt door middel van de `Sprite.SetTexture()` methode.
14. Schrijf twee manieren op om de snelheid van een Sprite weer te geven.  
`Velocity` is een vector, welke gerepresenteerd wordt door de snelheid in de X en Y richtingen of door snelheid en richting.
15. Leg uit wat "Collision detection" inhoud.  
Collision detection bestaat uit een controle van alle sprites op het scherm om te bekijken of een sprite tegen een andere is gebotst.
16. Schrijf op welke manieren van collision detection er zijn.
  - **Bounding rectangle collision**
    - Hier worden er virtuele rechthoeken om een afbeelding gebruikt om te bepalen of twee sprites tegen elkaar gaan botsen. Als de rechthoeken overlappen is er sprake van een botsing.
  - **Pixel-perfect collision**
    - Hier wordt er gecontroleerd of niet transparante pixels van twee sprites met elkaar overlappen. Wanneer dat het geval is dan is er sprake van een botsing.
17. Schrijf op welke methode een sprite over het scherm laat bewegen tot het verdwijnt.  
De `MoveAndVanish()` methode zal een sprite in een richting laten bewegen tot het de rand van het venster bereikt. Op dat moment wordt de `Sprite's IsAlive` eigenschap op `false` gezet.
18. Schrijf op hoe je een Sprite tekent op het scherm.  
Met de `Sprite.Draw()` methode. Zorg er wel voor dat eerst `SpriteBatch.Begin()` is aangeroepen voor de `Sprite.Draw()` aanroep.
19. Schrijf op wanneer (vanuit welke methode) het beste de spelers invoer kan worden verwerkt.  
Gebruiker invoer moet onderdeel zijn van de `Update()` methode van de `game logica`.
20. Leg uit waarom de muis cursor standaard verscholen is bij XNA spellen.  
De muis is standaard in XNA verscholen omdat het origineel gemaakt was voor Xbox spellen en deze kennen het concept van een muis niet.



## Opdrachten

### 1. My first game tutorial:

Ga naar de link hieronder en volg de instructies daar:

<https://msdn.microsoft.com/en-us/library/bb203893%28v=xnagamestudio.31%29.aspx>

In de volgende opdrachten wordt een Breakout spel gemaakt zoals in de afbeelding hier onder te zien is.

De .PDF bestanden leggen uit wat er precies in de code gebeurt.



### 2. 2D XNA Tutorials

Bekijk de tutorials bij de onderstaande link:

<http://rbwhitaker.wikidot.com/2d-tutorials>

### 3. Breaking Out – Deel 1:

Neem de .pdf BreakingOut1 door en bestudeer de bron code van BreakingOut1.

### 4. Breaking Out – Deel 2:

Neem de .pdf BreakingOut1 door en bestudeer de bron code van BreakingOut2.

### 5. Breaking Out – Deel 3:

Neem de .pdf BreakingOut1 door en bestudeer de bron code van BreakingOut3.