



Hvilke tekniske komplikationer er forbundet med  
skærmopløsning i forhold til 2d/3d spil og  
hvordan kan de forebygges?



ERHVERVSAKADEMI DANIA

Poul Munk, Michael Carnø Ranegaard, Niclas Houtved Andersen

# Indhold

1	Introduktion . . . . .	1
2	Teksturer der skalerer til alle skærmopløsninger . . . . .	1
3	Mulige løsninger til 2d teksturer . . . . .	1
4	Teksture . . . . .	2
4.1	Hukommelse i forhold til teksture . . . . .	3
5	Vector vs. Raster Grafik . . . . .	3
5.1	Raster over Vector i Spil Udvikling . . . . .	3
5.2	Test: Fil Størrelser . . . . .	4
6	Konklusion . . . . .	4
<b>A Referencer</b>		<b>7</b>

# 1 Introduktion

I denne artikel vil vi eksaminere problemer og løsninger i forhold til udfordringerne man måtte have med at få sit spil til at passe til den enorme mængde skærmstørrelser, aspekt ratioer og pixel tætheder som de forskellige enheder har. Noget som skal tænkes efter når man sidder og er i gang med sine design dokumenter er hvordan man takler de potentielle problemer der kan opstå i forhold til at få ens spil til at passe til de mange forskellige enheder.

## 2 Teksturer der skalerer til alle skærmopløsninger

Et problem der kan have væsentlig betydning for om en bruger vil spille ens spil, er håndteringen af pixels/teksturer over de forskellige enheder, og et problem som man absolut bør undgå er, at få sit spil til at skalere grafikken op til alle opløsninger og aspekter. Problemet gør sig i særdeleshed gældende når man skifter aspekt, da grafikken skal justeres i både højde og bredde, og det kan så have den konsekvens at billedet bliver fordrejet så det kan komme til at se dårlig og udtværet ud og i sidste ende være årsagen til at man holder op med at spille spillet.

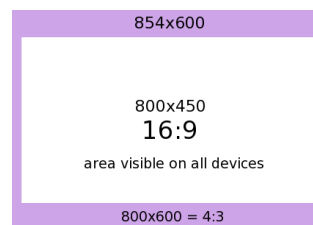
## 3 Mulige løsninger til 2d teksturer

De forskellige indstillinger har en markant betydning for 2d spil, og en mulig løsning for at fixe dette er at lave alt ens grafik så det passer til de forskellige opløsninger til de forskellige enheder. Hvis vi tager figur 1[1] som eksempel ser vi et udvalg af nogle af de lavere skærmopløsninger blandt spilbare enheder.

Her har vi den laveste opløsning som er 800x450

og den højeste på 854x600. Man bør derfor lave alt

baggrundsgrafik til den størst mulige opløsning og gameplay grafikken til den mindst mulige. Dette betyder, at man kan have forskellige opløsninger og aspekter uden at det har nogen markant påvirkning for grafikken, da det blot er synsfeltet der skal justeres. Et større synsfelt kan dog have sine konsekvenser for gameplayet, så det er vigtigt at få det testet og justeret så gameplayet passer til det man havde forstillet sig. Et oplagt eksempel ville være et tower defense spil, da et større synsfelt betyder at der er mere plads at bygge på, og spilleren med den største vil på den måde få have størst fordel. Skal man op i de højere opløsninger bør man tage den samme fremgangsmåde som førnævnte eksempel.



Figur 1:

Typical resolution	Play area	Background	Scale
800x480	800x450	854x600	100%
1024x600	960x540	1024x800	120%
1280x800	1280x720	1280x1024	160%
1536x1152	1536x864	1536x1152	192%
1920x1200	1920x1080	2048x1536	240%
2560x1600	2560x1440	2560x1600	320%

Ud fra overstående tabel[1] kan man se, at man bør have flere sæt grafik for at dække alle de anvendte skærmopløsninger og aspekter for 2d spil hvis man følger denne fremgangsmåde.

En anden mulighed er, at lade vær med at bruge skalering og bare holde sig til den størrelse som pixel-sene er lavet i og have grænser i siderne, toppen eller, og, bunden af skærmen, også kaldt *boxing*<sup>1</sup>. Denne metode kan være meget tidsbesparende og have den effekt, at produktions tiden bliver reduceret markant eller at den besparede tid kan sættes af til at lave mere indhold, da grafikerne blot skal fokusere på at lave alt materiale til en opløsning i stedet for at skal lave det til adskillige [2].



Figur 2:

## 4 Teksture

Teksture kan være et stort problem, i forhold til skærmopløsning, når man laver spil, da teksturene ikke er lavet til alle skærme. Hvis man vælger at lave sine teksture i den mest normale/brugte grafiktype som er raster grafik, kan man i hvert fald få problemer med opløsningen. En måde som man kan komme det problem til livs er, ved at man kan lave teksturene højere opløsning end de egentligt skal være, og så nedskalere dem i spillet til den valgte opløsning. Ofte er det en god idé at vælge dobbelt så høj opløsning som den opløsning man faktisk skal bruge.[3] Grunden til det er en god idé at lave teksturene i højere opløsning er, at det er lettere at nedskalere en tekstur end at opskalere den, da detaljerne så stadig er der, og at man også samtidig fremtidssikrer spillet på den måde, ved at det passer til næste generation af skærmopløsning. Alt efter hvilket objekt man laver, er det ikke nødvendigt at man bruger en fast høj opløsning, hvis man laver en lille ting til et spil som for eksempel en kugle til en pistol, så kan man sagtens bare tage en opløsning

<sup>1</sup><https://support.apple.com/en-us/HT204429>

der passer til at objektet ikke vil blive set særlig meget.[4]

## 4.1 Hukommelse i forhold til teksture

Som sagt er det en god idé at have højere opløsning end man faktisk skal bruge. Men til f.eks. den mobile platform er hukommelsen meget mere begrænset, så der er ikke plads til kæmpe store filer. Så derfor for at spare plads, kan man tage sin tekstur og så lave flere versioner i lidt forskellige størrelser, ved at nedskalere tekturen selv, så spillet sparer hukommelse og arbejdskraft ved ikke at skulle nedskalere fra en kæmpe størrelse.[1] Det er ikke kun til mobiler at dette kan svare sig, da der er mange forskellige skærme, har folk også mange forskellige opløsninger. Så for folk med en skærm der har en lidt lavere opløsning, vil deres spil indlæse hurtigere ved at have en tekstur der ligger tættere på deres opløsning, så de ikke skal nedskalere den største version.

## 5 Vector vs. Raster Grafik

I forhold til raster grafik som vi kender fra .png, .jpg filer, så er .svg en type der er baseret ud fra vector grafik. Dette er smart grundet at raster grafik bliver pixileret og grumset når det forstørres for meget, **dette gør vector grafik tilgængelig ikke.** Dette vil blot tilpasse sig selv til størrelsen grundet det er baseret på matematiske formler.[5]

### 5.1 Raster over Vector i Spil Udvikling

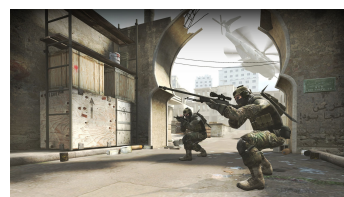
**Årsagen er** at raster grafik har flere grafiske muligheder i form af farver og effekter, dette har vector ikke på samme måde, det skal laves manuelt og små detaljer tager derfor lang tid. Men det udelukker ikke vector for at blive brugt til spil, men det bliver sandsynligvis simple former og mere cartoony grundet farve mulighederne. Spil såsom Castle Crashers og South Park: Stick of Truth, er begge gode eksempler på spil der med stor sandsynlighed har gjort brug af vector grafik i storstil, men



(a) Castle Crashers



(b) South Park: SoT



(c) Counter-Strike: GO





**Figur 3: Billeder af forskellige spil**

det kan også ses på farverne i spillene, de er ikke så detaljerede, i hvert fald ikke i samme stil som f.eks. Counter-Strike spillene, hvor man også kan se pixilering på jorden. Derfor er det nok sagen at mange spil vil have detaljer, realistisk grafik eller har ikke tiden til at lave det i vector og laver det derfor i raster i stedet for. [6]

## 5.2 Test: Fil Størrelse

Vi har foretaget en test af filstørrelser i forhold til Vector og Raster grafik. Vi har taget udgangspunkt i et billede på størrelsen **2048x2048**, dette fik vi programmet **InkScape** til at tegne op for os i Vector og gemte dette som vector fil. Denne fyldte **135MB** hvor at original raster filen fylder **2,6MB**, herefter skalerede vi ned til **1024x1024** og **512x512** som henholdsvis er **763KB** og **225KB** i fil størrelse, dette viser at vector fylder væsentlig mere. Vi valgte at tegne op for at få et billede

der har lige så stor detalje der hvor det er muligt, hvilket viser at filerne i vector er har en større fylde men de kan til gengæld skaleres op og ned, dog ikke uden at få de fejl og mangler med som raster billedet har haft, dette ville kræve at det blev tegnet op manuelt. Samtidig viser det at hvis denne teori holder, vil man i forhold til størrelse hellere skulle lave meget store raster billeder pga. Tiden og fil størrelse da man blot kan skalerer ned, tog man i stedet vector filer med i spillet og bad spillet renderer billeder der passer til skærm størrelsen, vil det at have vector filerne gøre spillet væsentlig tungere, end at lave rigtig mange raster billeder tilpasset til skærm størrelse. Dog skal der laves simple former, logoer og lignende vil vector grafik nok være idéen da de så ikke fylder nær så meget pga. De manglende detaljer, effekter kunne evt. Ligges ovenpå efter de er renderet til raster grafik, eller en raster tekstur i spillet ligges henover et logo på en væg for eksempel.

Name	Type	Size	Height	Width
 newbrickwallVector....	SVG File	139,006 KB		
 newbrickwall.png	PNG File	2,660 KB	2048 pixels	2048 pixels
 newbrickwall1024.p...	PNG File	764 KB	1024 pixels	1024 pixels
 newbrickwall512.png	PNG File	226 KB	512 pixels	512 pixels

Figur 4: Sammenligning af fil størrelser

## 6 Konklusion

Ud fra denne opgave kan vi konkludere at der er vælg af metoder til at bearbejde eventuelle problemer der måtte opstå i forhold til at få grafik til at passe til de forskellige enheder. Hvad 3d angår er det ret lige til da man blot kan skalere sin model, men det er mere problematisk hvis man arbejder med 2d grafik. Hvis man er på et stramt budget er det vær at overveje og lave sine materialer i en opløsning og skjule toppen og bunden af skærmen ved hjælp af boxing. Har man derimod et større budget og mere tid kan man lave sine materialer i flere opløsninger så man har et udvalg der passer til de forskellige opløsninger og aspekter. Vi kan også



konkluderer at raster grafik er bedst til detaljerede teksturer, hvor at vector er godt til mindre detaljerede f.eks. simple logo'er og figure, her vil vector være bedst da den kan skalereres op og ned, hvor at raster kun kan skales ned, hvis det stadig skal se godt ud.

# Bibliografi

- [1] Bigosaur. *Screen resolutions*. 2016. URL: <http://bigosaur.com/blog/31-android-resolutions-definite-answer>.
- [2] VPlay. *How to create mobile games*. 2016. URL: <http://v-play.net/doc/vplay-different-screen-sizes/>.
- [3] Todd Gantzler. *Game Development Essentials: Video Game Art*. 2004. Kap. 2 “Textures s. 43”.
- [4] qJake. *Making a Texture*. 2016. URL: <http://answers.unity3d.com/questions/20461/making-a-texture.html>.
- [5] Vector Conversions.com. *Raster vs. Vector*. 2016. URL: [http://vector-conversions.com/vectorizing/raster\\_vs\\_vector.html](http://vector-conversions.com/vectorizing/raster_vs_vector.html).
- [6] Stackoverflow API-Beast og Mansuro. *Why don't more games use vector art?* 2016. URL: <http://gamedev.stackexchange.com/questions/30111/why-dont-more-games-use-vector-art>.



Bilag  A

Referencer