# Tilemaps m.m. i Godot - kort intro

Formålet med dette dokument er at introducere hvorledes det er muligt i Godot hurtigt at lave flotte 2D spillebaner med den indbyggede node ”TileMap”. Noden tilføjes blot som en undernode til ens scene og man vælger hvilken ”tile shape” den skal have f.eks. square som er den mest almindelige. Dernæst sætter man størrelsen på disse squares, i billedet til højre er den 40x40.

Ideen med tilemaps er at inddele baggrunden i mindre områder og lave ”tiles” i den valgte størrelse som passer sammen så de kan sættes sammen og meget hurtig danne en rimelig ok spillebane.   
Den tidsmæssige største udfordring er (nok) at lave de tiles man sætter sammen, men heldigvis findes der allerede en del websites som tilbyder gratis tiles.

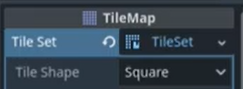
Links til et par websites med tiles:

<https://game-endeavor.itch.io/>

<https://kenney.nl/>

<https://devworm.itch.io/survival-game-godot-4-series-art> (specifik til følgende toturial: <https://www.youtube.com/watch?v=eAEe_9jCV4s> *som billederne i dette dokument er taget fra…*)

xxxx (Assets til mange af de tutorials som der er link til i dette dokument)

Inden man kan tilføje sine tiles til scenen, tilføjes de samlet i et tileset. Det gøres i TileMap nodens inspector faneblad, hvor et billede som f.eks. det der er vist nedenfor (typisk png), trækkes fra FileSystem vinduet og over i tileset feltet her vist til højre 🡪



Den samlede proces kan ses i følgende videoer der gennemgår dels hvordan man sikre at ens spillebane ikke kommer til at se ensformig ud, hvordan man arbejder med Layer, sikre kollision, sætter lyssætningen og hvordan man kan lave autotiling der gør processen hurtigere. Her er dog først et par kommentarer til Layers som kan være rart at vide.

***Layers*** er lavet om i Godot 4.x? (ældre videoer viser det derfor andeledens) så hvor de før lå under TileMap nodens inspector faneblad, ligger de nu som sin egen undernode til TileMaps og hedder TileMapsLayer. Layer (lag) systemet virker som i div. grafik-programmer, hvor et lag har områder med synligt indhold og andre områder der er gennemsigtig. Det højest prioriteret lag ligger over de andre så det altid vises, men de lag der ligger under, vises kun hvis deres indhold ligger på et sted med gennemsigtighed i de overliggende lag. Normalt er det øverste lag prioriteret højest men ikke i Godot. Her er det lag, som er tilføjet sidst og derfor ligger nederst, prioriteret højest (lidt ulogisk for mig 😊).

Denne tutorial viser hvordan TileMapLayer virker: <https://www.youtube.com/watch?v=PuCDAANCBVc>

Mange af de efterfølgende tutorials kan benyttes til at lære hvordan man arbejder med TileMaps, men vi har alligevel skrevet hvilke fokusområder vi tænker de er bedst til.

*Kollision* er beskrevet i de fleste af de tutorials, og vises i følgende tutorial:

<https://www.youtube.com/watch?v=eAEe_9jCV4s>

*Kollision & lyssætning:*

Part1. <https://www.youtube.com/watch?v=PDlwQM1AnYI>

Part2. <https://www.youtube.com/watch?v=6lximAlvBoE>

*Autotiling* vises bl.a. i denne tutorial som er en lang beginners tutorial der viser rigtigt meget og kan være et ok sted at starte: <https://www.youtube.com/watch?v=vV8uKN1VnN4>

***Assets*** til div. projekter er tilgængelig her: [Assets](https://teccph-my.sharepoint.com/:f:/g/personal/kkm_tec_dk/EqeHUMomc45PrCXl0T3T4hIBoTot8cCWmIAWp0FjGM-_uA?e=BtNSdm)

I kan tage en spillefigur fra en tutorial f.eks. <https://www.youtube.com/watch?v=eAEe_9jCV4s> (assets til denne tutorial kan findes under linket ovenfor) og sætte den ind i jeres tilemaps prøve projekt således, at I kan se om kollision virker. Det er muligt at få vist jeres kollisionsbokse mens spillet kører ved at slå det til i Godot under Debug 🡪 Visible collision shapes.

Der er en række personer der har lavet et stort stykke arbejde med at lægge tutorials ud på internettet og de er typisk ret godt til at lave spil. Nedenfor er links til to af de personer, så I kan tage et kig på deres youtube kanaler og se hvad de har lavet.

<https://www.youtube.com/@GameDevelopmentCenter>

<https://www.youtube.com/@dev-worm>