

DDU-opstart – august 2023

Dette dokument indeholder en oversigt over de forskellige færdigheder I skal arbejde med i starten af skoleåret samt links til hvor I finder relevant materiale.

En vigtig del af DDU er programmering og da der ofte er elever der ikke har kodet før, er der et særligt delafsnit der omhandler hvordan man helt fra bunden kan lære at kode. Hvis man allerede har en grundlæggende forståelse for programmering, kan man skimme dette delafsnit eller springe det over.

VIGTIG INFO OM DDU PÅ H.C.Ø gym. Lyngby

Information omkring DDU kan se på fagets website: <https://digitaltdesignlyngby.github.io/>

BLIV SPILUDVIKLER- INTRO

Grundlæggende information om det at blive spiludvikler generelt og særligt i Godot, samt hvad det vil sige at bruge en "engine" til at lave sine spil.

<https://www.gdquest.com/tutorial/getting-started/learn-to/game-engine-basics/>

Der er masser af links i disse tekster som vil lede dig til andre websider, undgå til at starte med at benyt disse links men læse teksterne færdige. De vigtigste links vil du alligevel finde nedenfor i dette dokument under den relevante overskrift.

Become a developer

Learn to Code

text

Game Engine Basics

text

Choosing the right game engine

text

30-minute practice

text

NY TIL PROGRAMMERING

Der er flere veje du kan vælge at lære at programmere. Det optimale er at lære GDScript med det samme da det er et specielt udviklet sprog til Godot for at gøre udviklingsprocessen lettere, men med den ulempe at sproget kun benyttes i Godot. Alternativt er det betydeligt lettere at finde materiale omkring Python som er et meget udbredt sprog med masser af tutorials og "beginners guide". Det er et sprog der minder om GDScript men de er ikke ens og man skal derfor lære to sprog hvis man tager udgangspunkt i Python.

Erfaringsmæssigt har det vist sig at det at kaste sig ud i Godot og lære GDScript uden noget kendskab til programmering er (for) svært. At starte med Python og dernæst skifte til GDScript er langsomt og demotiverer derfor visse personer som ønsker at komme hurtigere i gang med udviklingen af spil.



På baggrund af ovenstående problematik har folkene bage websitet GDQuest udviklet en App som gør indlæringen af GDScript lettere og vi anbefaler at I starter med den selv om I ikke kender til programmering på forhånd. Her kan du se en appens introfilm: <https://www.youtube.com/watch?v=GmWZ19Uw77o>

Her finder du appen: <https://gdquest.github.io/learn-gdscript/>

Som supplerende materiale kan henvises til GDScript websitet som har tutorials over de grundlæggende begreber inden for programmering, ligeledes med udgangspunkt i GDScript sproget: <https://gdscript.com/tutorials/>



Eller gamedevacadamy (kort beskrivelse af div. begreber inden for kode): <https://gamedevacademy.org/complete-gdscript-godot-tutorial/>



Endvidere kan Python bruges som beskrevet tidligere og bogen "Automate the boring stuff" findes online her (kapitel 1): <https://automatetheboringstuff.com/2e/chapter1/>

En let måde at få en Python editor på, er ved at benytte en der er online som f.eks. på w3schools der også har en masse Python eksempler og starter fra bunden: https://www.w3schools.com/python/trypython.asp?filename=demo_default

GDSCRIPT

Hvis du har valgt at springe forrige afsnit over, er det fordi du har et vist kendskab til det at programmere. Du begynder dog det samme sted som dem der ikke kan programmere, så vi ved at alle har gennemgået de samme opgaver til at starte med. Start derfor med at se introfilmen til GDQuest Appen: <https://www.youtube.com/watch?v=GmWZ19Uw77o> og gennemfør så opgaverne: <https://gdquest.github.io/learn-gdscript/>



Herefter er der masser af læsestof på Godots egen webside omkring GDScript:

<https://docs.godotengine.org/en/3.5/tutorials/scripting/gdscript/index.html>

Der er eksempler på kode i andre sprog hvor de sammenligner med GDScript kode, det kan du evt. springe over

For at være sikker på at du har lært og kan huske de vigtigste ting, kan du løbe nedenstående spørgsmål igennem.

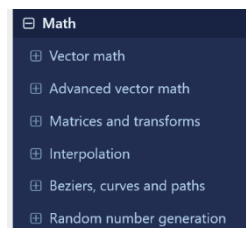
1. Hvad er GDScript og hvilket andet sprog minder det om?
2. Hvad betyder extends?
3. Hvordan definerer man en variabel i GDScript?
4. Hvad er en funktion og hvordan afsluttes den?
5. Hvorfor er den anden "Note" under "GDScript reference" vigtig?
6. Nævn tre vigtige fordele og tre ulemper ved "dynamic languages"
7. Hvilke type løkker kan man lave i GDScript?
8. Hvad bruges en iterator til?
9. Hvad betyder ## i GDScript?
10. Hvorfor findes der style guides til GDScript?
11. Brug af "explicit types" giver hvilke fordele?
12. Hvornår kan brug af placeholders i en string være en fordel?

VIGTIG MATEMATIK TIL SPIL

Når vi skal lave vores spil, benytter vi os altid af algoritmer der er små sammensatte stykker kode der kan styre logikken i spillet. Det kræver ofte noget tid at få disse algoritmer til at fungere optimalt og for at vi kan skrive dem optimalt skal vi benytte en række matematiske metoder.

På Godots webside er der en gennemgang af en række af disse vigtige metoder som du med fordel kan læse om her:

<https://docs.godotengine.org/en/3.5/tutorials/math/index.html>



Fra tidligere år hvor vi benyttede Java og editoren Processing, tog vi udgangspunkt i de fire første kapitler i bogen "The nature of code" ift. grundlæggende matematik i spil. Bogen findes online her:



<https://natureofcode.com/book/>

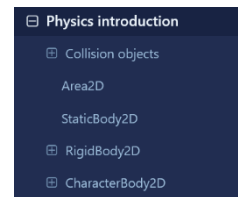
Bogen har selvfølgelig den ulempe nu at den forklarer tingene med udgangspunkt i et andet sprog (Java) end GDScript, hvilket gør at eksemplerne og opgaverne kan være svære at forstå at benytte. Principperne bag koden altså matematikken fungerer dog stadigvæk fint og bogen kan derfor godt være en kilde til forståelse af matematik i spil.

1. **VECTORS**
2. **FORCES**
3. **OSCILLATION**
4. **PARTICLE SYSTEMS**

KOLLISION

En vigtig del af mange spil er kontrol af hvornår forskellige spilelementer kolliderer. For at få en forståelse af hvordan det foregår i Godot kan du læse følgende afsnit på Godots egen website:

https://docs.godotengine.org/en/3.5/tutorials/physics/physics_introduction.html



GODOT (generelt)

På følgende link kan du finde og downloade **Godot version 3.5.2 (vi bruger denne version)** eller 4.1.1:

<https://godotengine.org/>

På Godots officielle website finder du masser af information omkring Godot og hvordan man bruger programmet. Hvis du vil have en **offline version** af deres website, kan det downloades her:

https://nightly.link/godotengine/godot-docs/workflows/build_offline_docs/master/godot-docs-html-master.zip

Der er desuden en masse links til **div. tutorials** der kan være interessante her:

<https://docs.godotengine.org/en/3.5/community/tutorials.html>

I DDU benytter vi **undervisningsbogen** "Godot-Game-Development-for-Beginners" der ligger på DDU websitet som pdf. Den første del af bogen kan også findes online her: <https://gamedevacademy.org/how-to-make-games-godot/>