

EGENREFLEKSJON OG RAPPORT FOR YFF

FYLLES UT VED OPPSTART

Problemstilling

Hva er målet? Hva ønsker jeg å lære?

Lære hvordan pygame fungerer og lage et enkelt spill i python med Pygame.

Kompetansemål?

Hvilke kompetansemål har du lyst til å trene mot i denne oppgaven. Skriv ned minst to mål.

Trenger du hjelp, spør en lærer om råd 😊

Kompetansemål som er relevant til dette prosjektet. Jeg har tatt bort deler av et kompetansemål fordi deler av det ikke samsvarer med prosjektet.

- Fagbrevet jobbet mot her, er i hovedsak IT-utvikler
- Av kompetansemålene, vil jeg i hovedsak jobbe med:
- Feilsøke kode og rette feil i algoritmer og kode
- bruke utviklingsstrategier og samarbeide med andre utviklere
- utforske prosesser knyttet til testing og feilsøking i utvikling av løsninger
- Men jeg kommer til å være innom deler av kompetansemålene for:
 - Planlegge, utvikle og dokumentere løsninger med innebygd personvern og sikkerhet
 - Utvikle og tilpasse brukergrensesnitt som ivaretar krav til universell utforming
 - Utforske og vurdere eksisterende kodespråk

Om prosjektet

Hva handler prosjektet om? (kort forklart)

Prosjektet handler om å prøve ut og lære om kommandoer i Pygame og lage et spill med det vi har lært underveis.

Hvilket utstyr trenger jeg? Programvare, Hardware osv.
VS code med pygame innstallert.

Mitt utgangspunkt

Hva slags erfaring har jeg om dette fra før. Kort fortalt

Jeg har jobbet med python i billedredigerings oppgaven, men aldri pygame. Etter å ha jobbet med pygame, ser jeg at man bruker også veldig få vanlige python koder.

Plan for gjennomføring

Beskriv hva du har tenkt til å gjøre i hver økt?

Første økt:

Planlegging av spill ider

Se gjennom kompetansemålene

Finne mer ut av hvordan bruke pygame

Få en karakter til å kunne styres opp, ned, venstre og høyre.

Andre økt:

Lage fiender

Lage en bakgrunn og byen

Tredje økt:

Skrive rapport

Gjøre spillet mer avansert hvis tid

SKRIVES HVER ØKT

Logg / fremgangsmåte

Skriv logg etter hver økt hvor du beskriver hva du har gjort.

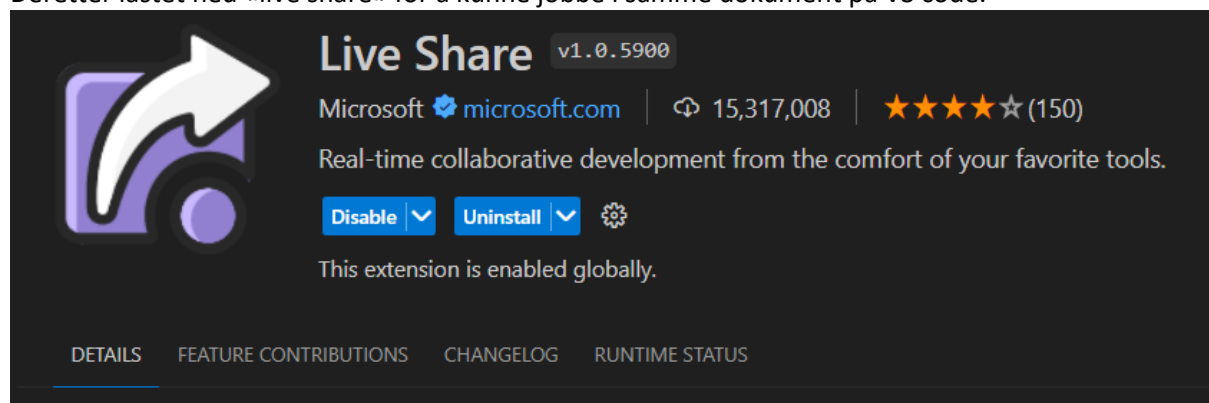
Hvilke steg har du tatt for å komme der du nå er i kodingen/prosjektet?

Legg inn bilder/skjermdump som viser prosessen.

Logg uke 1

Vi lastet ned og importerte pygame ved å: gå inn i comand prompt, skrive inn «python3 -m pip install -U pygame –user». Hvis dette ikke fungerer, kan du [trykke her](#) for å finne de andre mulige kommandoene for å installere pygame. → Til slutt skal du skrive «import pygame» øverst i en Python-fil. (man må gjøre dette for å kunne kjøre spillet)

Deretter lastet ned «live share» for å kunne jobbe i samme dokument på VS code.



Dette fungerte ikke så bra, på grunn av at programmet ikke alltid fungerte som det skulle. For eksempel: Når jeg skrålet opp og ned på skjermen skjedde det samme på Jørgen din skjerm og den kan være treg til å oppdatere seg selv.

Vi tokk derfor å opplastet koden vår på Discord i en gruppe vi lagde for å dele endringene vi gjør til hverandre.

Vi brukte mesteparten av tiden på å forstå mer om Pygame ved å se videoer og søke på nett. Ideen vår for spillet er at du er en Nordkoreansk flyktning som skal over grensa til Kina. På grensen er det vakter som beskytter området og du må snike deg rundt dem. (denne ideen endret seg etter hvert)

Det vi la inn i spillet etter økt en er:

legge inn muligheten for å opplaste et bilde av karakteren, karakteren kan bevege seg opp, ned, til høyre og venstre og kan sprinte ved å trykke skift (vi fjernet sprinte funksjonen senere i spillet fordi det ikke ga mening å kunne løpe så kjapt i senere versjoner) og vi la til bakgrunnen som vi brukte [Al nettsiden Pixlr](#) for å lage.

Logg uke 2

For denne uken fikk vi ikke gjort så mye relatert til koden, men vi fikk lagd spriten til karakteren du spiller som i [pixelartmaker.com](#).

Deler av grunnen for at vi ikke kom oss så langt denne uken er på grunn av de annerledes timene og å at det var vanskelig for oss å koordinere hvem som skulle gjøre hva, nå som vi forsto at Live share ikke var en mulighet.

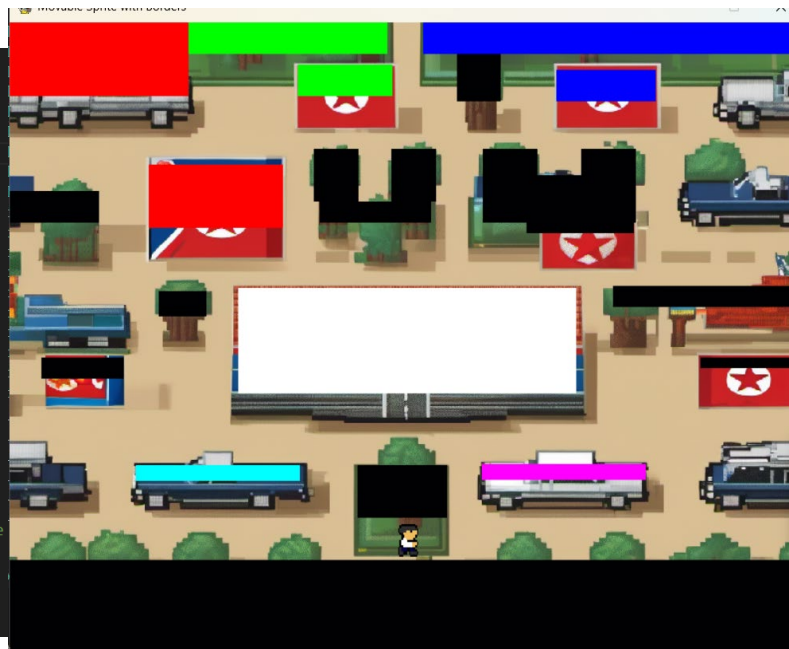
Vi hadde ikke nok tid til å lage 3 sprites til for å gå forover, bakover og til venstre, så karakteren ser alltid mot høyre.

Logg uke 3

Jørgen valgte å lage sitt eget spill istedenfor å fortsette å jobbe med meg og Eldar.

Denne dagen ble brukt for det meste for å forstå hvordan man lager hittboxer og deretter plassere dem ut, så spilleren ikke kan gå hvor den vil.

```
rect1 = CollidableRectangle(140, 135, 135, 60, (255, 0, 0))
rect2 = CollidableRectangle(290, 40, 95, 30, (0, 255, 0))
rect3 = CollidableRectangle(550, 45, 100, 30, (0, 0, 255))
rect4 = CollidableRectangle(32, 318, 83, 20, (0, 0, 2))
rect5 = CollidableRectangle(695, 318, 90, 10, (0, 0, 2))
rect6 = CollidableRectangle(535, 180, 93, 20, (0, 0, 2))
rect7 = CollidableRectangle(312, 170, 112, 20, (0, 0, 2))
rect8 = CollidableRectangle(306, 120, 45, 50, (0, 0, 2))
rect9 = CollidableRectangle(384, 120, 45, 50, (0, 0, 2))
rect10 = CollidableRectangle(350, 420, 90, 50, (0, 0, 2))
rect11 = CollidableRectangle(150, 255, 48, 24, (0, 0, 2))
rect12 = CollidableRectangle(0, 160, 90, 30, (0, 0, 2))
rect13 = CollidableRectangle(450, 30, 44, 45, (0, 0, 2))
rect14 = CollidableRectangle(575, 120, 55, 70, (0, 0, 2))
rect15 = CollidableRectangle(476, 120, 55, 70, (0, 0, 2))
rect16 = CollidableRectangle(520, 145, 55, 55, (0, 0, 2))
rect17 = CollidableRectangle(607, 250, 450, 20, (0, 0, 2))
rect18 = CollidableRectangle(678, 130, 450, 20, (0, 0, 2))
rect19 = CollidableRectangle(0, 510, 800, 110, (0, 0, 2))
collidable_rect1 = CollidableRectangle(0, 0, 180, 70, (255, 0, 0)) # Red rectangle
collidable_rect2 = CollidableRectangle(180, 0, 200, 30, (0, 255, 0))
collidable_rect3 = CollidableRectangle(416, 0, 400, 30, (0, 0, 255)) # Blue rectangle
collidable_rect4 = CollidableRectangle(127, 420, 165, 15, (0, 255, 255))
collidable_rect5 = CollidableRectangle(475, 419, 165, 15, (255, 0, 255))
collidable_rect6 = CollidableRectangle(230, 252, 340, 100, (255, 255, 255))
```



Grunnen for at hitt boxene ikke dekker hele området de er ment for å stoppe deg fra å gå inn i er for å la det være mer dybde, ved at hodet ditt går foran objektet og kun beina står inntil boksen.

Vi fant ut at å lage fiender blir for vanskelig så vi valgte å endre dem til å være miner isteden. Når du trækker på en mine vil den eksplodere og spille lukkes for så at det printes en melding i terminalen som sier: «Du tapte!».

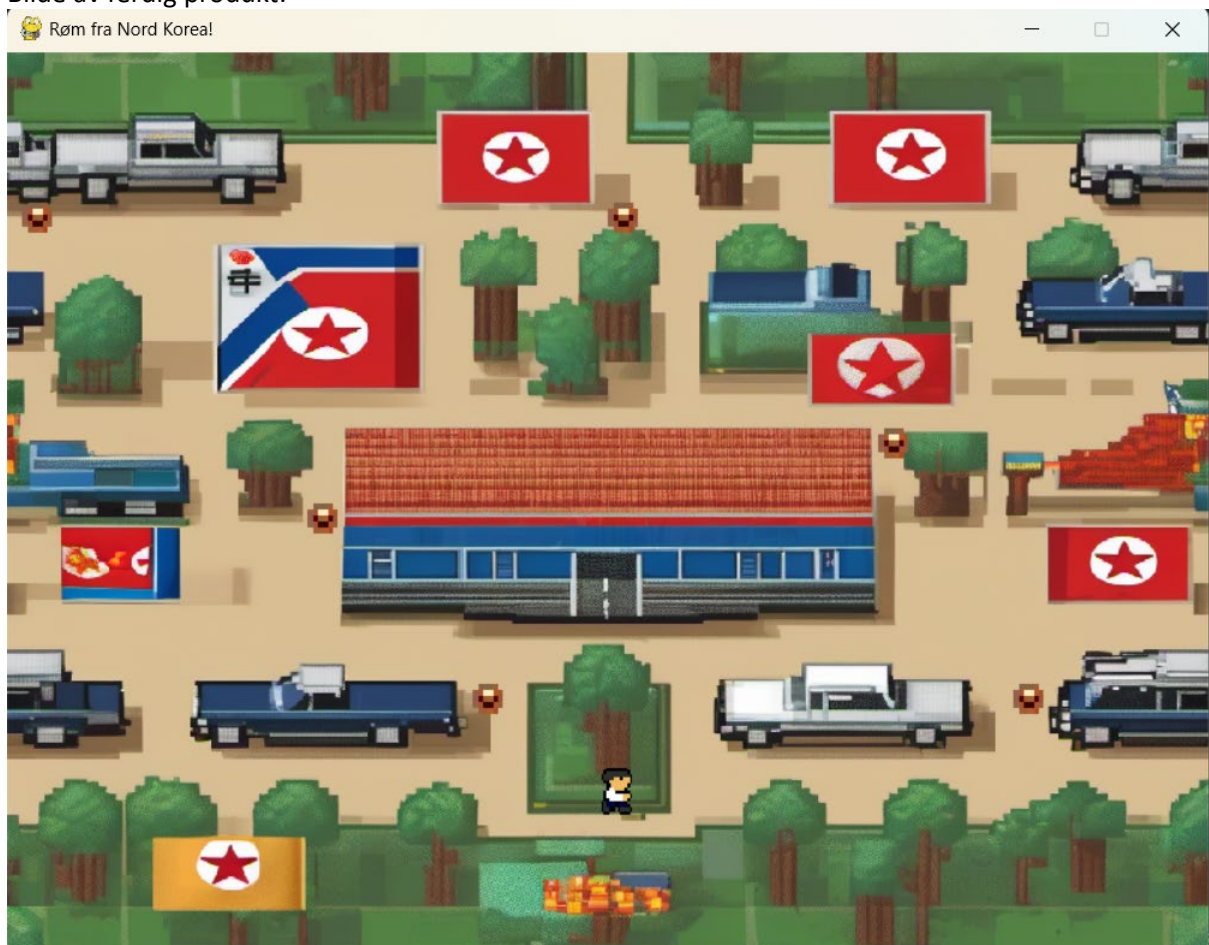
Vi valgte å ikke lage vakter fordi vi forsto at å lage fiender ville ta mer tid enn det vi hadde vi tenkte derfor å lage miner som hvis du går over ville slutte spillet fra å kjøre.

Når jeg begynte å jobbe med koden på lørdag, ville ikke koden på linje 138 fungere. Når jeg ble tilsendt koden til Eldar for å teste, var det akkurat samme problemet som skjedde. Jeg vet ikke hvorfor dette skjedde, men problemet ble fikset etter at jeg resatte PC-en.

Før hadde vi «pygame.display.update()» under samme «if» erklæring som «running = false, men jeg så at det ikke er noen vits i å oppdatere spillet rett før det skal lukkes, så jeg fjernet «pygame.display.update()»

```
if event.type == pygame.QUIT:  
    running = False  
    pygame.display.update()
```

Bilde av ferdig produkt:



FYLLES UT VED PROSJEKTETS SLUTT

Nådde jeg målet?

Hjelpespørsmål: *Hva fikk jeg til og hva gjenstår? Var det noen mål du ikke nådde? Hva var grunnen til at det gikk slik? Har jeg svart på problemstillingen?*

Vi måtte endre litt på spillet for å få det ferdig som skrevet over, men vi rakk å gjøre ferdig et enkelt spill i Python med pygame og jeg forstår mye mer av hvordan pygame fungerer. For eksempel bruker man nesten ingen vanlige Python kommandoer og koder under laging av et pygame spill. Noe av det eneste vi brukte fra Python biblioteket er løkker.

Hva har jeg lært

Hjelpespørsmål: *Hvilke nye begreper kan jeg nå noe om? Har jeg lært noen nye metoder? Har jeg lært noen nye verktøy?*

Koordinater i Pygame

Først vill jeg forklare hvordan koordinatsystemet i pygame fungerer, fordi alt som vises på skjermen trenger koordinater for å være plassert riktig og kunne bli vist. I et vanlig koordinatsystem er verdien 0 for x og y akse i midten, men i pygame er 0 øverst i venstre hjørne. I dette spillet vill derfor nederst i venstre hjørne ikke ha motsatt verdi av øverste, venstre hjørne, men heller en bredde(x) på 800 og en høyde(y) på 600.

Lage vinduet spillet vises fra

For å lage vinduet som Python skal vise spillet i, definerer vi bredden og høyden til vinduet, og for at spillet skal kjøre likt på alle maskiner, låser vi det til å ikke kunne kjøre på mer enn 60 bilder per sekund. Ved å låse bilde til 60 bilder per sekund, vil spille sjekke for inputer som for eksempel om «A» på tastaturet er trykket ned eller hva som er de nåværende koordinatene til karakteren 60 ganger per sekund.

```
WIDTH, HEIGHT = 800, 600
FPS = 60
```

Lagre bildene i variabler

Her blir alle bildene lagret i hver sin variabel som vi senere kaller for å gi en visualisering av karakteren, bakgrunnen, minene og karakteren som eksploderer etter å ha tråkket på en mine.

```
bg = r"C:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\IMG\bakgrunn.png"
img = r"C:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\IMG\karakter.png"
img2 = r"C:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\IMG\eksplosjon.png"
landmine = r"C:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\IMG\mine.png"
```

Legge til bakgrunnen

«pygame.image.load()» laster inn bilde av grensen på bakgrunnen når vi plasserer «bg»(variabelen bakgrunnen er lagret i) inn i parentesene. For å forbedre ytelsen til spillet, kan vi sette «.convert» etter «pygame.image.load()» for å konvertere bilde til å ha samme piksel format som vinduet (linje132).

I linje 133 gir vi «bg_image bredden og høyden til vinduet, ved å bruke «pygame.image.load()» og sia at det er bg_image som skal få den nye bredden og høyden og gi den nye høyden og bredden innenfor de to rosa parentesene. I dette tilfelle tar vi bredden og høyden som er lagret i «WIDTH og HEIGHT».

```
132 bg_image = pygame.image.load(bg).convert()
133 bg_image = pygame.transform.scale(bg_image, (WIDTH, HEIGHT))
```

Lage spill karakteren

Med denne koden blir karakteren lagd. Karakteren er definert via pygame.sprite.Sprite på linje 16 og er gitt en størrelse på 50 ganger 50 piksler, og ser for nå ut som en firkant. Dette er gjort på linje 19. for å få programmet til å plassere spilleren, brukte vi «self.rect.center = (WIDTH // 2, 490)» innenfor parentesene valgte vi bruke vanlige koordinater på den høyre siden av slash tegnene og for det venstre siden valgte vi å bruke «WIDTH». WIDTH definerte vi til å ha en bredde på 800 piksler for å lage bredden av vinduet, og vi brukte derfor denne variabelen på nytt for å gi starts posisjonen til karakteren.

```

16 class Player(pygame.sprite.Sprite):
17     def __init__(self):
18         super().__init__()
19         self.image = pygame.Surface((50, 50))
20         ogimg = pygame.image.load(img).convert_alpha()
21         self.image = pygame.transform.scale(ogimg, (25, 40))
22         self.rect = self.image.get_rect()
23         self.rect.center = (WIDTH // 2, 490)
24         self.move=True
25     def update(self):
```


Gi karakteren en sprite og få den til å kunne bevege seg

Karakteren er gitt variabelen som inneholder bilde av karakteren i linje 32.

I linje 34 bruker vi «key» som inneholder tilstanden til alle knappene på tastaturet. «key» kaller vi ved å bruke funksjonen «pygame.key.get_pressed()». Nå som programmet kan oppdage knapper som trykkes ned, kan vi si at: hvis «a» er trykket ned og koordinat posisjonen til venstre side av karakteren er større enn 0, vill karakteren sin x koordinat bli satt ned med 5.

det er det samme som skjer når du trykker på w, s og d, men forskjellen er at programmet sjekker for forskjellige posisjoner av karakteren sine sider for å ikke la karakteren gå på utsiden av vinduet, og den gir større eller mindre verdi til enten x eller y aksen.

```

32     global img
33     if self.move == True:
34         keys = pygame.key.get_pressed()
35         previous_rect = self.rect.copy()
36         if keys[pygame.K_a] and self.rect.left > 0:
37             self.rect.x -= 5
38         if keys[pygame.K_d] and self.rect.right < WIDTH:
39             self.rect.x += 5
40         if keys[pygame.K_w] and self.rect.top > 0:
41             self.rect.y -= 5
42         if keys[pygame.K_s] and self.rect.bottom < HEIGHT:
43             self.rect.y += 5

```

Hvordan vinne spillet

Når karakteren treffer det øverste punktet på y aksen (0) som bare er mulig å komme seg til via den lille passasjen helt øverst i vinduet, vil Python printe en melding som sier «Du greide å komme deg ut!» og vinduet vill deretter lukkes og programmet vil stoppes.

```

26     def update(self):
27         if self.rect.top == 0:
28             print("Du greide å komme deg ut!")
29             pygame.quit()
30             sys.exit()

```

Lage rektangler med koordinater, størrelse, farge og synlighet, og vise dem i vinduet

«CollidableRectangle» blir definert av koden og arver (å basere et objekt eller klasse på et annen objekt eller klasse) fra class.sprite.Sprite klassen i linje 66.

I linje 67 gir koden muligheten for å endre posisjonen, bredden og høyden, fargen og gjennomsiktighetn(alpha=0) til rektanglene, som vi kan gi verdier til ved å lage en variabel (f.eks. rect1) og sette inn verdiene til hver av dem i samme rekkefølge vi skrev dem opp i linje 67 (f.eks. i linje 74 ser vi at bredden(width) er 135, mens høyden(height) er 60. hvis vi hadde byttet plassen på høyden og bredden i linje 67, ville disse to også byttet verdier).

I linje 70, blir «fill» kalt for å fylle rektangler som rect1 med RGB koden som er spesifisert innenfor de rosa parentesene.

```

66 class CollidableRectangle(pygame.sprite.Sprite):
    CodiumAI: Options | Test this method
    def __init__(self, x, y, width, height, color, alpha=0):
67         super().__init__()
68         self.image = pygame.Surface((width, height), pygame.SRCALPHA)
69         self.image.fill((color[0], color[1], color[2], alpha))
70         self.rect = self.image.get_rect()
71         self.rect.topleft = (x, y)
72
73
74 rect1 = CollidableRectangle(140, 135, 135, 60, (255, 0, 0))
75 rect2 = CollidableRectangle(290, 40, 94, 30, (0, 255, 0))
76 rect3 = CollidableRectangle(550, 45, 100, 30, (0, 0, 255))
77 rect4 = CollidableRectangle(32, 318, 83, 20, (0, 0, 2))
78 rect5 = CollidableRectangle(695, 318, 90, 10, (0, 0, 2))
79 rect6 = CollidableRectangle(535, 180, 93, 20, (0, 0, 2))
80 rect7 = CollidableRectangle(312, 170, 112, 20, (0, 0, 2))
81 rect8 = CollidableRectangle(306, 120, 45, 50, (0, 0, 2))
82 rect9 = CollidableRectangle(384, 120, 45, 50, (0, 0, 2))
83 rect10 = CollidableRectangle(350, 420, 90, 50, (0, 0, 2))
84 rect11 = CollidableRectangle(150, 255, 48, 24, (0, 0, 2))
85 rect12 = CollidableRectangle(0, 160, 90, 30, (0, 0, 2))
86 rect13 = CollidableRectangle(450, 30, 44, 45, (0, 0, 2))
87 rect14 = CollidableRectangle(575, 120, 55, 70, (0, 0, 2))
88 rect15 = CollidableRectangle(476, 120, 55, 70, (0, 0, 2))
89 rect16 = CollidableRectangle(520, 145, 55, 55, (0, 0, 2))
90 rect17 = CollidableRectangle(607, 250, 450, 20, (0, 0, 2))
91 rect18 = CollidableRectangle(678, 130, 450, 20, (0, 0, 2))
92 rect19 = CollidableRectangle(0, 510, 800, 110, (0, 0, 2))
93 collidable_rect1 = CollidableRectangle(0, 0, 180, 70, (255, 0, 0))
94 collidable_rect2 = CollidableRectangle(180, 0, 200, 30, (0, 255, 0))
95 collidable_rect3 = CollidableRectangle(416, 0, 400, 30, (0, 0, 255))

```

Vise rektanglene

For å vise rektanglene i koden definerer vi variabelen collidable_sprites med «Pygame.sprite.Group()». Denne klassen holder styr på valgte sprite objekter.

Nå kaller vi på «collidable_sprites.add» i linje 125 og setter variabelen til rektanglene innenfor parantesene. «add» er der for å forklare til Python at vi vil legge inn og vise rektanglene i spillet.

Vi har gjort akkurat det samme for minene, men spesiviserte ikke farge eller om figurene skal være transparente, siden vi uannset ikke trengte de verdiene(linje 100 – 113, 121, 129 og 130).

```

108 mine1 = mines(10, 100, 20, 20, landmine)
109 mine2 = mines(400, 100, 20, 20, landmine)
110 mine3 = mines(670, 420, 20, 20, landmine)
111 mine4 = mines(580, 250, 20, 20, landmine)
112 mine5 = mines(310, 420, 20, 20, landmine)
113 mine6 = mines(200, 300, 20, 20, landmine)
114
115 screen = pygame.display.set_mode((WIDTH, HEIGHT))
116 pygame.display.set_caption("Røm fra Nord Korea!")
117 clock = pygame.time.Clock()
118
119 all_sprites = pygame.sprite.Group()
120 collidable_sprites = pygame.sprite.Group()
121 land_mines = pygame.sprite.Group()
122 player = Player()
123 all_sprites.add(player)
124
125 collidable_sprites.add(rect1, rect2, rect3, rect4, rect5, rect6, rect7, rect8, rect9, rect10, rect11, rect12, rect13, rect14, rect15, rect16, rect17, rect18, rect19)
126 all_sprites.add(collidable_sprites)
127
128
129 land_mines.add(mine1, mine2, mine3, mine4, mine5, mine6)
130 all_sprites.add(land_mines)

```


Stoppe karakteren fra å gå inn i figurene vi nettopp lagde

Men vi kan fortsatt ikke samhandle med både rektanglene og minene. For å få rektanglene til å være umulige å gå gjennom, kan vi gjøre som i linje 45 til 47. her definerer vi «collisions» med `pygame.sprite.spritecollide`. «`spritecollide`» er brukt for å sjekke om «self» (karakteren) og «`collidable_sprites`» (rektanglene) rører hverandre. Hvis «self» og «`collidable_sprites`» rører hverandre vil koden stoppe deg fra å bevege deg inn i figuren ved å sende deg tilbake til den siste posisjonen du var på, før du traff veggen.

```
45     collisions = pygame.sprite.spritecollide(self, collidable_sprites, False)
46     if collisions:
47         self.rect = previous_rect
```

Eksplodere karakteren hvis du treffer en mine

For å eksplodere spilleren og lukke spillet når du treffer en mine, definerer vi en variabel, og forteller den at en handling skal skje om karakteren og minene treffer hverandre. Akkurat som «`collidable_sprites`» fra forrige avsnitt.

Når du treffer en mine vil spriten til karakteren bli byttet ut med et bilde av en eksplosjon(`img2`). «`collision_time`» og «`player_transformed`» blir aktivert. «`Collision_time`» blir definert med «`pygame.time.get_ticks()`» som noterer tiden karakteren treffer minen, og hvis «`player_transformed`» er lik «`TRUE`»(sann) og det har blitt notert ned et tidspunkt, vil «`player_transformed`» telle ned fra 1000 før den først printer «Du tapte!» og deretter lukker og stopper spillet.

```
49     explode = pygame.sprite.spritecollide(self, land_mines, False)
50     global img2
51     global collision_time
52     global player_transformed
53
54     if explode and not player_transformed:
55         player_transformed = True
56         self.move = False
57         self.image = pygame.image.load(img2).convert_alpha()
58         self.image = pygame.transform.scale(self.image, (150, 200))
59         collision_time = pygame.time.get_ticks()
60
61     if player_transformed and pygame.time.get_ticks() - collision_time > 1000:
62         print("Du tapte!")
63         pygame.quit()
64         sys.exit()
```

Spillmotoren

Når «running» er satt til «True», vil koden under håndtere hendelser, oppdatere bildene og vise dem på skjermen. Altså er dette hvor hoved oppgavene programmet tar for seg blir gjort. Dette er spillmotoren til programmet.

I linje 138, hvis du lukker spill applikasjonen vil «pygame.quit()» bli kjørt og spillet avsluttes.

```
135 while True:
136     for event in pygame.event.get():
137         if event.type == pygame.QUIT:
138             pygame.quit()
```

Ved å bruke denne metoden vil spillet støte på en del feilmeldinger rett før spillet avsluttes.

```
Traceback (most recent call last):
  File "c:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\spill.py", line 140, in <module>
    all_sprites.update()
  File "C:\Users\fredr\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalC
prite.py", line 555, in update
    sprite.update(*args, **kwargs)
  File "c:\Users\fredr\OneDrive - Viken fylkeskommune\1IMA\YFF\Pygame\spill.py", line 34, in update
    keys = pygame.key.get_pressed()
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
pygame.error: video system not initialized
```

Men de har egnetlig ingen ting å si, siden spillet uansett lukkes et millisekund etter.

Om man ikke vil ha disse feilmeldingene, kan du bruke denne koden isteden:

```
135 running = True
136 while running:
137     for event in pygame.event.get():
138         if event.type == pygame.QUIT:
139             running = False
140
141     all_sprites.update()
142     screen.blit(bg_image, (0, 0))
143     all_sprites.draw(screen)
144     pygame.display.flip()
145     clock.tick(FPS)
146
147 pygame.quit()
```

Når denne koden lukker spill applikasjonen, vil «running» bli satt til «FALSE». Når «running» er satt til «FALSE» vil «pygame.quit()» bli aktivert og spillet avsluttes.

- «all_sprites.update()» sier at «.update()» blir kalt på «all_sprites» variabelen. Dette oppdaterer bildene «all_sprites» refererer til.
- «screen.blit(bg_image, (0, 0))» sier at bakgrunnsbildet skal blegges til på koordinatene (0, 0) ved å bruke blit()-metoden.
- «draw()» metoden er kalt etter for å tegne alle bildene fra «all_sprites».
- «flip()» er brukt for å for å oppdatere vinduet.
- Klokken får verdien til «FPS» til å bestemme hvor ofte spillet skal oppdateres.

```
141 all_sprites.update()
142 screen.blit(bg_image, (0, 0))
143 all_sprites.draw(screen)
144 pygame.display.flip()
145 clock.tick(FPS)
```

Oppsummering:

Jeg forstår mye mer om pygame nå, enn det jeg gjorde før.

- Jeg forstår hvordan koordinater fungerer
- Hvordan vise bilder i et vindu og hvordan lage det vinduet
- Hvordan få en karakter til å kunne bevege seg og gjøre enkle interaksjoner med spill verdenen
 - Lage vegger spilleren ikke kan gå gjennom
 - Lage objekter som slutter spill loopen(minene)
- Hvordan forenkle og feilsøke i pygame
- Vet hva mange koder og kommandoer fra pygame biblioteket gjør og hvordan man bruker dem.
- Forstår bedre hvor mye man kan få gjort på 3 uker
- Hvordan jobbe som et lag i programmering

Hva ville vært neste steg for deg om du skal lære mer om akkurat dette?

Hjelpespørsmål: Om du skulle utviklet konseptet videre, hva tenker du mangler?

Lære hvordan man lager enkel KI for fiender og legge til flere baner som blir vanskeligere og vanskeligere pluss animasjon til karakteren.

Hvilke ønsker har du for neste YFF-prosjekt?

NOE MINDRE STRESSENDE Å FÅ GJORT FERDIG