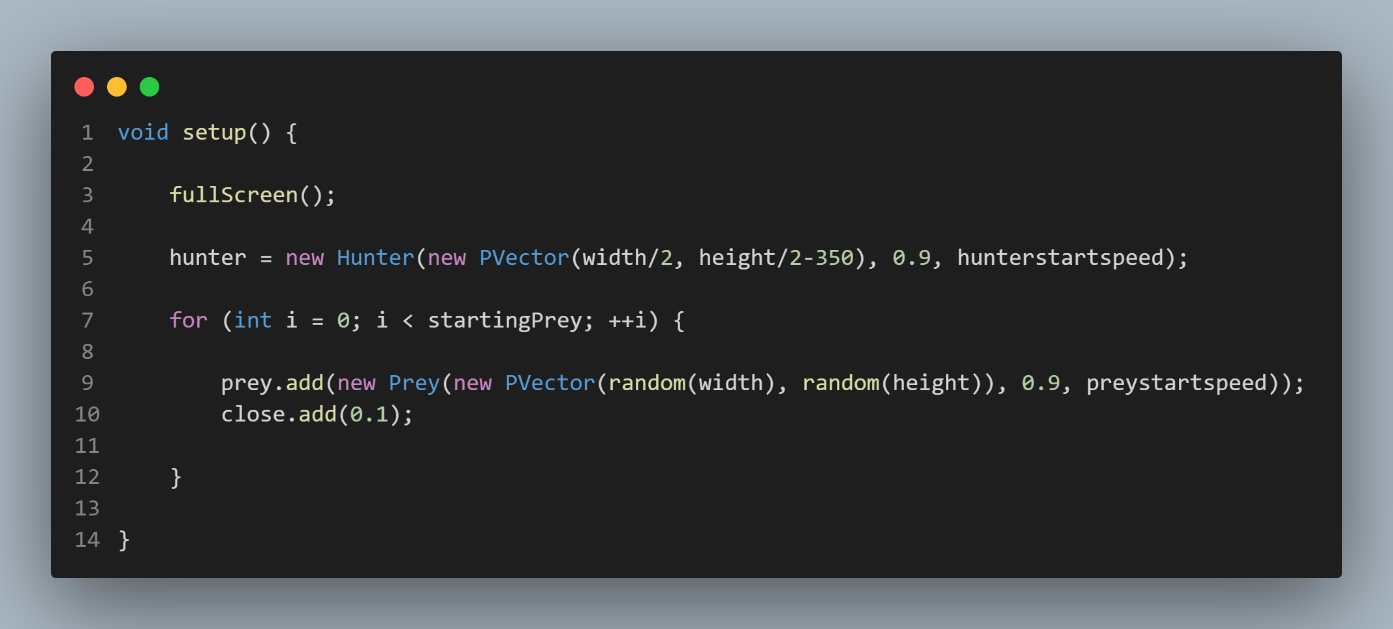
Kodeforklaring

Forneden ses funktionen setup. Denne funktion bliver kørt én gang lige i starten af programmet. Spillet startes med at blive sat i fullscreen. Derefter laves en hunter med tre forskellige parametre: location, maxforce og maxspeed. Til sidst er der et for-loop hvor der fra variablen startingPrey er fastsat en mængde af preys, og derudfra spawnes det fastsatte antal preys.



Her ses *wander* funktionen. Den begynder med at definere tre variabler - i dette tilfælde *floats*. Der bliver defineret en *wradius* som er *wander radius*. Derefter vælger den en tilfældig vinkel, og derefter bevæger den sig i den retning.



Forneden ses *move* funktion af *prey’et*. De første fire *if-statements* tjekker for om *location* er 25 pixels fra kanten. Hvis én af disse er sandt får hunter/prey ”lyst” til at bevæge sig den anden vej ved brug af *desired* variablen.

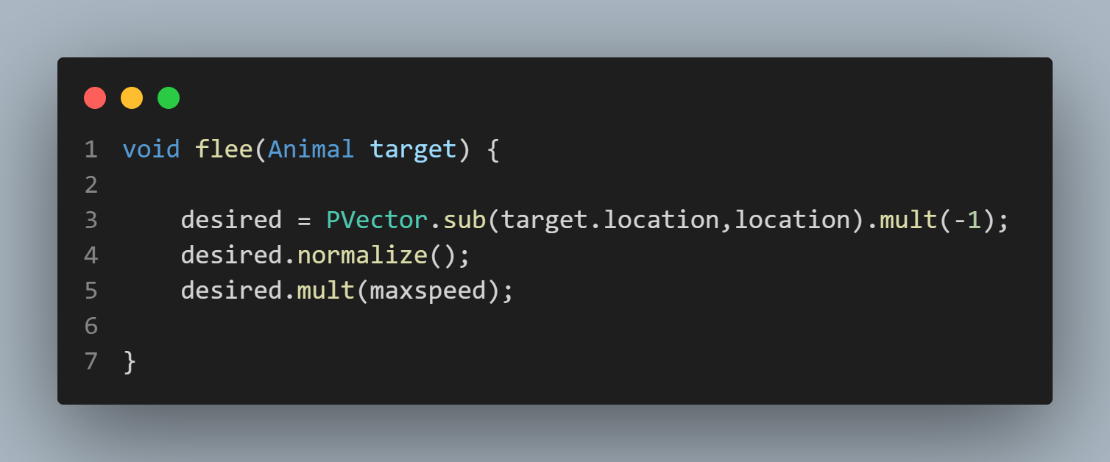
Det femte *if*-*statement* tjekker for om *hunters* lokation og *preys* lokation er mindre end *hunters range*. Hvis dette er sandt, sætter den i dette tilfælde gang i *seek* funktionen og multiplicerer dens *maxspeed* med 1,01. Hvis det havde været hunterens *move* funktion, havde det været *seek* funktionen der var blevet kaldt.

Sjette og sidste *if*-*statement* tjekker for når timeren løber ud, og derefter sætter den gang i *wander* funktionen, og nulstiller derefter timeren igen.



Forneden ses to forskellige metoder. Den første bruger *prey’et*, og den anden bruger *hunter’en*.

Første funktion hedder *flee*. Denne funktion bruger *prey’et* til at flygte fra *hunter’eren*. I *preys* move funktion bliver denne flee funktion kaldt for at få *prey’et* til at bevæge sig i den modsatte retning.



Som nævn foroven, bliver seek funktionen kaldt når et *prey* er inden for *hunters* *range*. Derefter får hunteren en ”lyst” til at bevæge sig mod det udvalgte prey.

