

```

import pygame
from pygame.sprite import Sprite # Importerer Sprite-klassen fra
pygame for at
                                # håndtere spilobjekter som
sprites.
                                # Læs her: https://www.pygame.org/docs/ref/
sprite.html

class Ship(Sprite):
    """
    Sprite-klassen, der styrer skibet.
    """

    def __init__(self, ai_game):
        """
        Initialiserer skibet og dets startposition
        """
        # Kører Sprite-klassen med __init__-metode
        # for at arve dens funktionalitet.
        super().__init__()
        # Reference til spillet skærm som defineret i ai-objectet.
        self.screen = ai_game.screen
        # Reference til spillet indstillinger som defineret i
        Settings-objectet.
        self.settings = ai_game.settings
        # Henter rektanglen (position samt dimensioner) for skib
        objectet.
        self.screen_rect = ai_game.screen.get_rect()

        # Indlæs billede af skip
        self.image = pygame.image.load('images/ship.bmp')
        # Definer rektangel (position og dimensioner)
        self.rect = self.image.get_rect()

        # Placerer rumskibet nederst i midten
        self.rect.midbottom = self.screen_rect.midbottom

        # Gemmer den vandrette position af skibet som decimal
(float)
        # for præcis bevægelse.
        self.x = float(self.rect.x)

        # Bevægelser (flags): inaktive ved spillets start indtil
TRYK.
        self.moving_right = False
        self.moving_left = False

    def center_ship(self):
        """
        Centrerer skibet i bunden og midten ved tab af liv.
        """
        # Placer i midten
        self.rect.midbottom = self.screen_rect.midbottom

```

```

        # Placeringen er et præcist decimaltal.
        self.x = float(self.rect.x)

    def update(self):
        """
        Opdaterer skibets placering baseret på inputs.
        Denne metode kaldes for at opdatere skibets position,
        hver gang skærmen opdateres.
        """
        # Flytter rumskibet til højre hvis tryk-input er aktiv
        # OG rumskibet ikke rører kanten.
        if self.moving_right and self.rect.right <
self.screen_rect.right:
            # Ændrer (øger) den vandrette position med den
indstillede hastighed.
            self.x += self.settings.ship_speed
        # Flytter rumskibet til venstre hvis tryk-input er aktiv
        # OG rumskibet ikke rører kanten.
        if self.moving_left and self.rect.left > 0:
            # Ændrer (reducerer) den vandrette position
            # med den indstillede hastighed.
            self.x -= self.settings.ship_speed

        # Opdaterer skibets (rektangelens) position
        # baseret på den præcise x-værdi.
        self.rect.x = self.x

    def blitme(self):
        """
        Metoden, der indtegner skibet på dets aktuelle position
        """
        self.screen.blit(self.image, self.rect)

```