**Ja:** Implement a `Kamera` (camera) class in `triedy.kamera`

**AI:** I'll help you implement a `Kamera` class that will handle camera movement and following the player in the game. Based on the codebase, we'll need to integrate it with the existing sprite and scene system.

Here's the implementation for `triedy/kamera.py`:

"""

Modul pre kameru.

"""

import typing as t

import pygame

from triedy.sprite.entity import Hrac

class Kamera:

"""

Kamera, ktorá sleduje hráča a posúva všetky objekty na obrazovke.

Používa sa staticky (nie je potrebné vytvárať viac ako jednu inštanciu).

"""

OFFSET = pygame.Vector2(0, 0)

"""Aktuálny posun kamery."""

@staticmethod

def sleduj\_hraca(hrac: Hrac, velkost\_okna: t.Tuple[int, int]):

"""

Nastaví pozíciu kamery tak, aby sledovala hráča.

"""

Kamera.OFFSET.x = hrac.rect.centerx - velkost\_okna[0] // 2

Kamera.OFFSET.y = hrac.rect.centery - velkost\_okna[1] // 2

@staticmethod

def aplikuj\_na\_sprite(sprite: pygame.sprite.Sprite) -> pygame.Rect:

"""

Aplikuje posun kamery na daný sprite a vráti jeho nový rect.

"""

return pygame.Rect(

sprite.rect.x - Kamera.OFFSET.x,

sprite.rect.y - Kamera.OFFSET.y,

sprite.rect.width,

sprite.rect.height,

)

Now we need to modify the `Level` class to use the camera. Here's how to update `triedy/scena/levely/level.py`:

import nastavenia as n

from triedy.kamera import Kamera

class Level(Scena):

def \_\_init\_\_(self, mapa\_id: str):

super().\_\_init\_\_()

self.nacitat\_level(mapa\_id)

self.hrac = None

def nacitat\_level(self, mapa\_id: str):

self.mapa = pytmx.load\_pygame(

str(self.LEVELY\_ROOT / f"{mapa\_id}.tmx"), pixelalpha=True

)

for layer in self.mapa.visible\_layers:

if isinstance(layer, pytmx.TiledTileLayer):

for x, y, gid, \*\_ in layer.iter\_data():

tile = self.mapa.get\_tile\_image\_by\_gid(gid)

if tile:

sprite = Sprite(

(x \* self.mapa.tilewidth, y \* self.mapa.tileheight)

)

sprite.image = tile

self.add(sprite)

elif isinstance(layer, pytmx.TiledObjectGroup):

for obj in layer:

if obj.name == "hrac":

self.hrac = Hrac((obj.x, obj.y))

self.add(self.hrac)

def update(self):

super().update()

if self.hrac:

Kamera.sleduj\_hraca(self.hrac, n.VELKOST\_OKNA)

def draw(self, surface: pygame.Surface):

for sprite in self.sprites():

surface.blit(sprite.image, Kamera.aplikuj\_na\_sprite(sprite))

The camera will keep the player centered on the screen as they move around the level. The camera offset is applied to all sprites during rendering, creating the illusion of camera movement.

To use this, no changes are needed in other files since we're using the existing sprite and scene system. The camera will automatically work when levels are rendered.