Technická dokumentace - Záchrana včel (Bee Saver)

1. Základní třídy

1.1 GameObject (game_object.py)

Základní třída pro všechny herní objekty. Tato třída poskytuje základní funkcionalitu pro pozicování, pohyb a vykreslování herních objektů.

Atributy

- `x` (int): X-ová souřadnice objektu
- `y` (int): Y-ová souřadnice objektu
- `width` (int): Šířka objektu
- `height` (int): Výška objektu
- `color` (tuple): RGB barva objektu (výchozí: černá)
- `speed_y` (int): Rychlost pohybu ve směru Y (výchozí: 0)
- `rect` (pygame.Rect): Obdélník pro kolizní detekci
- `image` (pygame.Surface): Obrázek objektu (volitelný)

Metody

`__init__(self, x, y, width, height, color=(0, 0, 0), speed_y=0, image=None)`

Inicializuje nový herní objekt.

Parametry:

- `x` (int): Počáteční X-ová souřadnice
- `y` (int): Počáteční Y-ová souřadnice
- `width` (int): Šířka objektu
- `height` (int): Výška objektu
- `color` (tuple): RGB barva objektu
- `speed_y` (int): Rychlost pohybu ve směru Y
- `image` (pygame.Surface): Obrázek objektu

`update(self)`

Aktualizuje pozici objektu.

Chování:

- Přidává `speed_y` k Y-ové souřadnici
- Aktualizuje pozici kolizního obdélníku

`draw(self, screen)`

Vykreslí objekt na obrazovku.

Parametry:

- `screen` (pygame.Surface): Plocha pro vykreslování

Chování:

- Pokud je nastaven `image`, vykreslí obrázek
- Jinak vykreslí obdélník s nastavenou barvou

`is_off_screen(self, screen_height)`

Kontroluje, zda je objekt mimo spodní okraj obrazovky.

Parametry:

- `screen_height` (int): Výška obrazovky

Návratová hodnota:

- `bool`: True, pokud je objekt mimo obrazovku

Příklad použití

```
# Vytvoření nového objektu
objekt = GameObject(x=100, y=100, width=50, height=50, color=(255, 0, 0))

# Aktualizace pozice
objekt.update()

# Vykreslení na obrazovku
objekt.draw(screen)

# Kontrola, zda je objekt mimo obrazovku
if objekt.is_off_screen(screen_height=600):
    # Objekt je mimo obrazovku
    pass
```

Poznámky

- Tato třída slouží jako základ pro všechny herní objekty
- Ostatní třídy jako `Player`, `Bee`, `Wasp`, `Hive` a `Honey` dědí z této třídy
- Poskytuje základní funkcionalitu pro:
    - Pozicování objektů
    - Pohyb ve směru Y
    - Vykreslování (obrázek nebo obdélník)
    - Detekci kolizí pomocí obdélníku
    - Kontrolu, zda je objekt mimo obrazovku

Dědičnost

Tato třída slouží jako základ pro všechny herní objekty. Ostatní třídy jako `Player`, `Bee`, `Wasp`, `Hive` a `Honey` dědí z této třídy a rozšiřují její funkcionalitu.

1.2 Player (player.py)

Třída reprezentující hráče (včelaře). Hráč se pohybuje pouze horizontálně po spodní hraně obrazovky a sbírá padající včely.

Atributy

- `x` (int): X-ová souřadnice včelaře
- `y` (int): Y-ová souřadnice včelaře (vždy na spodní hraně obrazovky)
- `width` (int): Šířka včelaře (výchozí: 70)
- `height` (int): Výška včelaře (výchozí: 90)
- `speed` (int): Rychlost pohybu (výchozí: 5)
- `lives` (int): Počet životů (výchozí: 3)
- `score` (int): Skóre hráče

- `rect` (pygame.Rect): Kolizní obdélník
- `color` (tuple): Barva včelaře (žlutá)
- `image` (pygame.Surface): Obrázek včelaře
- `screen_height` (int): Výška obrazovky
- `stunned_until` (float): Čas do kdy je hráč omráčen
- `bee_buffer` (int): Aktuální počet včel v zásobníku
- `bee_buffer_max` (int): Maximální kapacita zásobníku včel (výchozí: 5)
- `sprite_frame_width` (int): Šířka jednoho snímku spritu
- `sprite_frame_height` (int): Výška jednoho snímku spritu
- `animation_frames_walk` (list): Seznam snímků pro animaci chůze
- `animation_frame_idle` (pygame.Surface): Snímek pro animaci stání
- `current_frame_index` (int): Index aktuálního snímku animace
- `last_animation_update` (int): Čas poslední aktualizace animace
- `animation_speed` (int): Rychlost animace v milisekundách
- `is_moving` (bool): Zda se včelař pohybuje
- `facing_right` (bool): Směr, kterým včelař hledí

Metody

`__init__(self, x, y, width=70, height=90, speed=5, screen_height=800)`

Inicializuje nového včelaře.

Parametry:

- `x` (int): Počáteční X-ová souřadnice
- `y` (int): Počáteční Y-ová souřadnice
- `width` (int): Šířka včelaře
- `height` (int): Výška včelaře
- `speed` (int): Rychlost pohybu
- `screen_height` (int): Výška obrazovky

Chování:

- Načte sprite včelaře z assets/beekeeper-scaled.png
- Inicializuje animační snímky pro chůzi a stání
- Nastaví výchozí hodnoty pro animaci a pohyb

`update_animation(self)`

Aktualizuje aktuální snímek animace včelaře.

Chování:

- Přepíná mezi snímky chůze při pohybu

- Zobrazuje snímek stání při nehybnosti
- Převrací sprite podle směru pohybu
- Škáluje sprite na požadovanou velikost

`is_stunned(self)`

Kontroluje, zda je včelař momentálně omráčen.

Návratová hodnota:

- `bool`: True, pokud je včelař omráčen

`move(self, dx, screen_width, max_x=None)`

Pohybuje včelařem horizontálně po spodní hraně obrazovky.

Parametry:

- `dx` (int): Směr pohybu (-1 pro vlevo, 1 pro vpravo)
- `screen_width` (int): Šířka obrazovky
- `max_x` (int, optional): Maximální X-ová souřadnice

Chování:

- Včelař se nemůže pohybovat, pokud je omráčen
- Pohyb je omezen na hranice obrazovky
- Y-ová pozice zůstává konstantní na spodní hraně
- Aktualizuje stav pohybu a směr pro animaci

`draw(self, screen)`

Vykreslí včelaře na obrazovku.

Parametry:

- `screen` (pygame.Surface): Plocha pro vykreslování

Chování:

- Aktualizuje animaci před vykreslením
- Vykreslí aktuální snímek včelaře na správnou pozici

Příklad použití

```
# Vytvoření nového včelaře
vcelar = Player(x=400, y=0, screen_height=800)

# Pohyb včelaře doprava
vcelar.move(1, screen_width=800)

# Kontrola omráčení
if vcelar.is_stunned():
    print("Včelař je omráčen!")

# Vykreslení včelaře
vcelar.draw(screen)
```

Poznámky

- Včelař se pohybuje pouze horizontálně
- Y-ová pozice je vždy fixována na spodní hraně obrazovky
- Včelař má omezený počet životů
- Systém zásobníku včel umožňuje sbírat více včel najednou
- Omráčení dočasně znemožňuje pohyb
- Implementuje animaci chůze a stání
- Používá sprite z assets/beekeeper-scaled.png
- Sprite se automaticky převrací podle směru pohybu

1.3 Bee (bee.py)

Třída reprezentující včelu v hře. Včela padá shora dolů a může být zachráněna včelařem. Dědí z třídy `GameObject`.

Atributy

Dědí všechny atributy z `GameObject`:

- `x` (int): X-ová souřadnice včely
- `y` (int): Y-ová souřadnice včely
- `width` (int): Šířka včely (40)
- `height` (int): Výška včely (40)
- `color` (tuple): Barva včely (světle žlutá)
- `speed_y` (int): Rychlost pádu (výchozí: 4)
- `rect` (pygame.Rect): Kolizní obdélník
- `image` (pygame.Surface): Obrázek včely

Další atributy:

- `frames` (list): Seznam snímků pro animaci včely

- `current_frame` (int): Index aktuálního snímku animace
- `animation_speed` (float): Rychlost animace v sekundách na snímek
- `last_update` (int): Čas poslední aktualizace animace

Metody

`__init__(self, screen_width, max_x=None, speed_y=4)`

Inicializuje novou včelu. Včela se náhodně objeví na horní hraně obrazovky a začne padat dolů.

Parametry:

- `screen_width` (int): Šířka obrazovky
- `max_x` (int, optional): Maximální X-ová souřadnice pro spawn včely
- `speed_y` (int): Rychlost pádu včely

Chování:

- Načte sprite včely z assets/bee-sprite.png
- Nastaví náhodnou X-ovou pozici
- Y-ová pozice začíná nad obrazovkou
- Inicializuje animační snímky (používá snímky 2 a 3 ze spritesheetu)
- Sprite je načten ze spritesheetu a škálován na požadovanou velikost

`update(self)`

Aktualizuje pozici včely a animaci.

Chování:

- Aktualizuje pozici včely (dědí z GameObject)
- Přepíná mezi animačními snímky podle nastavené rychlosti
- Aktualizuje aktuální snímek včely

Příklad použití

```
# Vytvoření nové včely
vcelicka = Bee(screen_width=800)

# Vytvoření včely s omezeným rozsahem spawnu
vcelicka_omezena = Bee(screen_width=800, max_x=400)

# Vytvoření včely s vlastní rychlostí pádu
vcelicka_rychla = Bee(screen_width=800, speed_y=6)
```

Poznámky

- Včela se spawnuje náhodně na horní hraně obrazovky
- Používá sprite z assets/bee-sprite.png
- Sprite je načten ze spritesheetu (4 včely v jednom obrázku)
- Používá pouze snímky 2 a 3 ze spritesheetu pro animaci
- Včela padá konstantní rychlostí dolů
- Dědí základní funkcionalitu z GameObject (update, draw, is_off_screen)

1.4 Wasp (wasp.py)

Třída reprezentující vosu v hře. Vosa padá shora dolů a představuje nebezpečí pro včelaře. Dědí z třídy `GameObject`.

Atributy

Dědí všechny atributy z `GameObject`:

- `x` (int): X-ová souřadnice vosy
- `y` (int): Y-ová souřadnice vosy
- `width` (int): Šířka vosy (40)
- `height` (int): Výška vosy (40)
- `color` (tuple): Barva vosy (tmavá)
- `speed_y` (int): Rychlost pádu (výchozí: 5)
- `rect` (pygame.Rect): Kolizní obdélník
- `image` (pygame.Surface): Obrázek vosy

Další atributy:

- `frames` (list): Seznam snímků pro animaci vosy
- `current_frame` (int): Index aktuálního snímku animace
- `animation_speed` (float): Rychlost animace v sekundách na snímek
- `last_update` (int): Čas poslední aktualizace animace

Metody

`__init__(self, screen_width, max_x=None, speed_y=5)`

Inicializuje novou vosu. Vosa se náhodně objeví na horní hraně obrazovky a začne padat dolů.

Parametry:

- `screen_width` (int): Šířka obrazovky
- `max_x` (int, optional): Maximální X-ová souřadnice pro spawn vosy
- `speed_y` (int): Rychlost pádu vosy

Chování:

- Načte sprite vosy z assets/zla-vosa.png
- Nastaví náhodnou X-ovou pozici
- Y-ová pozice začíná nad obrazovkou
- Inicializuje všechny 4 snímky ze spritesheetu pro animaci
- Sprite je načten ze spritesheetu a škálován na požadovanou velikost

`update(self)`

Aktualizuje pozici vosy a animaci.

Chování:

- Aktualizuje pozici vosy (dědí z GameObject)
- Přepíná mezi animačními snímky podle nastavené rychlosti
- Aktualizuje aktuální snímek vosy

Příklad použití

```
# Vytvoření nové vosy
vosa = Wasp(screen_width=800)

# Vytvoření vosy s omezeným rozsahem spawnu
vosa_omezena = Wasp(screen_width=800, max_x=400)

# Vytvoření vosy s vlastní rychlostí pádu
vosa_rychla = Wasp(screen_width=800, speed_y=7)
```

Poznámky

- Vosa se spawnuje náhodně na horní hraně obrazovky
- Používá sprite z assets/zla-vosa.png
- Sprite je načten ze spritesheetu (4 snímky v jednom obrázku)
- Používá všechny 4 snímky ze spritesheetu pro animaci
- Padá rychleji než včela (výchozí rychlost: 5)
- Při kolizi s včelařem způsobuje ztrátu životů
- Dědí základní funkcionalitu z GameObject (update, draw, is_off_screen)

1.5 Hive (hive.py)

Třída reprezentující úl v hře. Úl je umístěn v pravém dolním rohu obrazovky a slouží jako cíl pro zachráněné včely. Dědí z třídy `GameObject`.

Atributy

Dědí všechny atributy z `GameObject`:

- `x` (int): X-ová souřadnice úlu (pravý okraj obrazovky)
- `y` (int): Y-ová souřadnice úlu (spodní okraj obrazovky)
- `width` (int): Šířka úlu (výchozí: 60)
- `height` (int): Výška úlu (výchozí: 80)
- `color` (tuple): Barva úlu (hnědá)
- `rect` (pygame.Rect): Kolizní obdélník
- `image` (pygame.Surface): Obrázek úlu

Další atributy:

- `original_image` (pygame.Surface): Původní obrázek úlu před škálováním
- `bee_buffer` (int): Aktuální počet včel v úlu
- `bee_buffer_max` (int): Maximální kapacita úlu (výchozí: 15)

Metody

`__init__(self, screen_width, screen_height, width=60, height=80)`

Inicializuje nový úl. Úl je umístěn v pravém dolním rohu obrazovky.

Parametry:

- `screen_width` (int): Šířka obrazovky
- `screen_height` (int): Výška obrazovky
- `width` (int): Šířka úlu
- `height` (int): Výška úlu

Chování:

- Načte sprite úlu z assets/hive-scaled2.png
- Nastaví pozici úlu do pravého dolního rohu
- Škáluje sprite na požadovanou velikost
- Inicializuje zásobník včel na 0
- Nastaví maximální kapacitu úlu na 15 včel

Příklad použití

```
# Vytvoření nového úlu
ul = Hive(screen_width=800, screen_height=600)

# Vytvoření většího úlu
velky_ul = Hive(screen_width=800, screen_height=600, width=80, height=100)
```

Poznámky

- Úl je statický objekt (nemá rychlost pohybu)
- Je umístěn v pravém dolním rohu obrazovky
- Slouží jako cíl pro zachráněné včely
- Používá sprite z assets/hive-scaled2.png
- Sprite je škálován na požadovanou velikost
- Má omezenou kapacitu (15 včel)
- Dědí základní funkcionalitu z GameObject (draw)
- Kolize s úlem znamená úspěšnou záchranu včely

1.6 Honey (honey.py)

Třída reprezentující med v hře. Med je bonusový předmět, který padá shora dolů a může být sebrán včelařem. Dědí z třídy `GameObject`.

Atributy

Dědí všechny atributy z `GameObject`:

- `x` (int): X-ová souřadnice medu
- `y` (int): Y-ová souřadnice medu
- `width` (int): Šířka medu (20)
- `height` (int): Výška medu (20)
- `speed_y` (int): Rychlost pádu (výchozí: 4)
- `rect` (pygame.Rect): Kolizní obdélník
- `image` (pygame.Surface): Obrázek medu
- `original_image` (pygame.Surface): Původní obrázek medu před škálováním

Metody

`__init__(self, screen_width, max_x=None, speed_y=4)`

Inicializuje nový med. Med se náhodně objeví na horní hraně obrazovky a začne padat dolů.

Parametry:

- `screen_width` (int): Šířka obrazovky
- `max_x` (int, optional): Maximální X-ová souřadnice pro spawn medu
- `speed_y` (int): Rychlost pádu medu

Chování:

- Nastaví náhodnou X-ovou pozici v rozsahu 0 až max_x
- Y-ová pozice začíná nad obrazovkou (-height)
- Načte a škáluje obrázek medu z assets/honeycomb.png
- Med padá stejnou rychlostí jako včela (výchozí: 4)

Příklad použití

```
# Vytvoření nového medu
med = Honey(screen_width=800)

# Vytvoření medu s omezeným rozsahem spawnu
med_omezeny = Honey(screen_width=800, max_x=400)

# Vytvoření medu s vlastní rychlostí pádu
med_rychly = Honey(screen_width=800, speed_y=6)
```

Poznámky

- Med se spawnuje náhodně na horní hraně obrazovky
- Je menší než včela (20x20 pixelů)
- Padá stejnou rychlostí jako včela (výchozí: 4)
- Při kolizi s včelařem přidává bonusové body
- Dědí základní funkcionalitu z GameObject (update, draw, is_off_screen)
- Používá vlastní obrázek z assets/honeycomb.png

1.7 Game (game.py)

Hlavní třída hry, která řídí celý herní proces. Spravuje herní stav, zpracovává vstupy, aktualizuje pozice objektů a vykresluje herní scénu.

Atributy

Základní nastavení

- `width` (int): Šířka herního okna (výchozí: 500)
- `height` (int): Výška herního okna (výchozí: 700)
- `screen` (pygame.Surface): Herní plocha
- `clock` (pygame.time.Clock): Herní hodiny pro FPS
- `running` (bool): Stav běhu hry
- `fps` (int): Cílové FPS (60)

Herní objekty

- `player` (Player): Instance včelaře
- `hive` (Hive): Instance úlu
- `bees` (list): Seznam aktivních včel
- `wasps` (list): Seznam aktivních vos
- `honey` (Honey): Instance medu (pokud existuje)

- `hive_forbidden_x` (tuple): Zakázaná oblast pro spawn (x-ové souřadnice úlu)
- `max_bee_x` (int): Maximální x-ová souřadnice pro spawn včel a vos

Čítače a časovače

- `spawn_timer` (int): Čítač pro spawn včel
- `spawn_interval` (int): Interval spawnu včel (80 ticků)
- `wasp_spawn_timer` (int): Čítač pro spawn vos
- `wasp_spawn_interval` (int): Interval spawnu vos (160 ticků)

Stav hry

- `score` (int): Aktuální skóre
- `game_over` (bool): Stav konce hry
- `keys` (dict): Slovník stisknutých kláves
- `score_effect` (tuple): Efekt získání bodů (text, x, y, time_end)
- `life_effect_end` (float): Čas konce efektu ztráty života
- `last_life_lost_time` (float): Čas poslední ztráty života
- `last_life_lost_index` (int): Index posledního ztraceného života
- `prev_lives` (int): Předchozí počet životů

Grafika

- `background` (pygame.Surface): Herní pozadí
- `font` (pygame.font.Font): Malý font pro HUD
- `large_font` (pygame.font.Font): Větší font pro Game Over info
- `go_font` (pygame.font.Font): Velký font pro GAME OVER
- `heart_icon` (pygame.Surface): Ikona srdce pro životy
- `heart_off_icon` (pygame.Surface): Ikona prázdného srdce
- `HEART_SIZE` (int): Velikost ikony srdce (48)
- `HEART_SPACING` (int): Mezera mezi srdci (2)

Metody

`__init__(self, width=500, height=700, title="Bee Saver")`

Inicializuje hru a vytváří herní okno.

Parametry:

- `width` (int): Šířka herního okna
- `height` (int): Výška herního okna
- `title` (str): Titulek herního okna

Chování:

- Inicializuje Pygame a vytváří herní okno
- Vytváří včelaře uprostřed spodní části obrazovky
- Vytváří úl v pravém dolním rohu
- Inicializuje všechny herní objekty a stavové proměnné
- Načítá všechny potřebné obrázky a fonty

`reset(self)`

Resetuje herní stav do výchozího nastavení.

Chování:

- Resetuje pozici včelaře a úlu
- Vyčistí seznamy včel a vos
- Resetuje skóre a časovače
- Resetuje stav hry a kláves

`handle_events(self)`

Zpracovává uživatelské vstupy a události.

Chování:

- Zpracovává události zavření okna
- Zpracovává stisky kláves:
    - ESC: Ukončení hry
    - R: Restart při game over
    - ENTER: Ukončení při game over
    - Šipky/A/D: Pohyb včelaře

`update(self)`

Aktualizuje herní stav.

Chování:

- Aktualizuje pozici včelaře podle stisknutých kláves
- Spawnuje nové včely a vosy podle časovačů
- Aktualizuje pozice všech herních objektů
- Kontroluje kolize mezi objekty
- Aktualizuje skóre a životy
- Kontroluje podmínky pro game over

`draw(self)`

Vykresluje herní scénu.

Chování:

- Vykresluje herní pozadí
- Vykresluje všechny herní objekty
- Vykresluje HUD (životy, skóre, zásobníky včel)
- Vykresluje efekty získání bodů a ztráty životů
- Vykresluje game over obrazovku

`run(self)`

Spouští hlavní herní smyčku.

Návratová hodnota:

- `bool`: False při ukončení hry

Chování:

- Spouští hlavní herní smyčku
- Zpracovává události, aktualizuje stav a vykresluje scénu
- Udržuje konstantní FPS

Herní mechaniky

Pohyb a ovládání

- Včelař se pohybuje horizontálně pomocí šipek nebo A/D
- Pohyb je omezen na levou stranu obrazovky a pravou stranu až k úlu

Bodový systém

- +1 bod za každou včelu doručenou do úlu
- Med přidává extra život
- Vosa způsobuje dočasné omráčení

Kolize

- Včelař + Včela: Sběr včely do zásobníku
- Včelař + Úl: Předání včel do úlu
- Včelař + Vosa: Omráčení
- Včelař + Med: Bonusový život

Game Over

- Hra končí při ztrátě všech životů
- Možnost restartu (R) nebo ukončení (ESC)

Poznámky

- Hra používá Pygame pro grafiku a vstupy
- Všechny herní objekty dědí z GameObject
- Herní logika je rozdělena do samostatných tříd
- Implementuje systém zásobníků pro včely
- Obsahuje vizuální efekty pro herní události
- Používá vlastní fonty a obrázky z assets složky

1.8 Menu (menu.py)

Třída reprezentující menu hry. Zajišťuje vykreslování a interakci s hlavním menu, úvodní obrazovkou, instrukcemi a kredity.

Konstanty

- `BUTTON_WIDTH` (int): Šířka tlačítek (200)
- `BUTTON_HEIGHT` (int): Výška tlačítek (50)
- `BUTTON_SPACING` (int): Mezera mezi tlačítky (0)

Atributy

Základní nastavení

- `screen` (pygame.Surface): Herní plocha
- `width` (int): Šířka okna
- `height` (int): Výška okna
- `buttons` (list): Seznam tlačítek menu
- `cloud_offset` (float): Offset pro animaci pozadí
- `selected_index` (int): Index vybraného tlačítka
- `keyboard_active` (bool): Zda je aktivní klávesové ovládání

Grafika

- `small_font` (pygame.font.Font): Malý font pro text
- `hover_font` (pygame.font.Font): Font pro hover efekt
- `logo_img` (pygame.Surface): Obrázek loga
- `night_menu_bg` (pygame.Surface): Pozadí menu
- `loading_bg` (pygame.Surface): Pozadí načítací obrazovky

Barvy

- `WHITE` (tuple): Bílá barva
- `YELLOW` (tuple): Žlutá barva

Animace

- `logo_anim_time` (float): Časovač pro animaci loga
- `loading_dots_time` (int): Časovač pro animaci teček
- `loading_screen_time` (int): Časovač načítací obrazovky
- `transition_alpha` (int): Průhlednost přechodu
- `transition_speed` (int): Rychlost přechodu
- `transition_surface` (pygame.Surface): Plocha pro přechod
- `transition_direction` (int): Směr přechodu

Metody

`__init__(self, screen, width, height)`

Inicializuje menu.

Parametry:

- `screen` (pygame.Surface): Herní plocha
- `width` (int): Šířka okna
- `height` (int): Výška okna

Chování:

- Inicializuje všechny potřebné proměnné
- Načítá fonty a obrázky
- Vytváří tlačítka menu

`init_buttons(self)`

Inicializuje tlačítka menu.

Chování:

- Vytváří tlačítka: START GAME, INSTRUCTIONS, CREDITS, EXIT GAME
- Nastavuje jejich pozice a rozměry

`handle_click(self, pos)`

Zpracovává kliknutí myší.

Parametry:

- `pos` (tuple): Pozice kliknutí

Návratová hodnota:

- `str`: Text tlačítka nebo None

`draw(self)`

Vykresluje hlavní menu.

Chování:

- Vykresluje animované pozadí
- Vykresluje logo s animací
- Vykresluje tlačítka s hover efektem

`draw_intro(self)`

Vykresluje úvodní obrazovku.

Chování:

- Vykresluje animované pozadí
- Vykresluje logo s animací skákání
- Zobrazuje text "LOADING GAME..." s animací teček

`draw_transition(self)`

Vykresluje přechodovou animaci.

Návratová hodnota:

- `str`: Nový stav ("menu" nebo "transition")

`draw_instructions(self)`

Vykresluje obrazovku s instrukcemi.

Návratová hodnota:

- `pygame.Rect`: Obdélník tlačítka zpět

Chování:

- Vykresluje nadpis "INSTRUCTIONS"

- Zobrazuje sekce GOAL a MOVEMENT/CONTROLS
- Vykresluje tlačítko zpět

`draw_credits(self)`

Vykresluje obrazovku s kredity.

Návratová hodnota:

- `pygame.Rect`: Obdélník tlačítka zpět

Chování:

- Vykresluje logo a nadpis "CREDITS"
- Zobrazuje sekce Game Design a Made with
- Vykresluje tlačítko zpět

Poznámky

- Menu podporuje ovládání myší i klávesnicí (W/S pro navigaci)
- Obsahuje animované pozadí a efekty
- Používá vlastní fonty z assets složky
- Implementuje přechodové animace mezi stavy
- Obsahuje interaktivní tlačítka s hover efektem
- Vykresluje načítací obrazovku s animací
- Používá vlastní obrázky pro logo a pozadí

2. Vstupní bod aplikace

2.1 main.py

Vstupní bod aplikace, který inicializuje a spravuje herní stavy. Implementuje hlavní herní smyčku a přepínání mezi různými stavy hry (intro, menu, hra, instrukce, kredity).

Konstanty a proměnné

Herní stavy

- `STATE_INTRO` (str): Úvodní obrazovka
- `STATE_MENU` (str): Hlavní menu
- `STATE_GAME` (str): Herní obrazovka
- `STATE_INSTRUCTIONS` (str): Obrazovka s instrukcemi
- `STATE_TRANSITION` (str): Přechodová obrazovka
- `STATE_CREDITS` (str): Obrazovka s kredity

Základní nastavení

- `WIDTH` (int): Šířka okna (800)
- `HEIGHT` (int): Výška okna (600)
- `screen` (pygame.Surface): Herní plocha
- `font` (pygame.font.Font): Základní font
- `clock` (pygame.time.Clock): Herní hodiny
- `state` (str): Aktuální herní stav

Funkce

`set_state(new_state)`

Přepíná herní stav a provádí potřebné inicializace.

Parametry:

- `new_state` (str): Nový herní stav

Chování:

- Nastaví globální proměnnou state
- Při přepnutí do herního stavu resetuje hru

`exit_game()`

Ukončí hru a zavře Pygame.

Herní smyčka

Hlavní herní smyčka zpracovává události a aktualizuje stav podle aktuálního herního stavu:

Zpracování událostí

- **Menu:**
  - Kliknutí na tlačítka (START GAME, INSTRUCTIONS, CREDITS, EXIT GAME)
  - Klávesnice (W/S pro navigaci, ENTER pro výběr)
  - ESC pro návrat do menu
- **Instrukce:**
  - ESC pro návrat do menu
  - Kliknutí na tlačítko zpět
- **Hra:**

    - ESC pro návrat do menu

- **Kredity:**
    - ESC pro návrat do menu
    - Kliknutí na tlačítko zpět

- **Intro:**
    - Jakákoliv klávesa pro přechod do menu
    - Automatický přechod po 3 sekundách

Aktualizace stavu

- **Intro:**
    - Zobrazuje úvodní obrazovku
    - Automatický přechod po 3 sekundách

- **Transition:**
    - Zobrazuje přechodovou animaci
    - Přepíná do dalšího stavu

- **Menu:**
    - Zobrazuje hlavní menu
    - Zpracovává interakce s tlačítky

- **Game:**
    - Spouští herní smyčku
    - Při ukončení hry se vrací do menu

- **Instructions:**
    - Zobrazuje obrazovku s instrukcemi

- **Credits:**
    - Zobrazuje obrazovku s kredity

Poznámky

- Hra používá Pygame pro grafiku a vstupy
- Implementuje systém stavů pro různé obrazovky
- Podporuje ovládání myší i klávesnicí
- Obsahuje přechodové animace mezi stavy
- Automaticky ukončuje hru při zavření okna
- Používá vlastní fonty a obrázky

- FPS je nastaveno na 60