Програмирање на видео игри

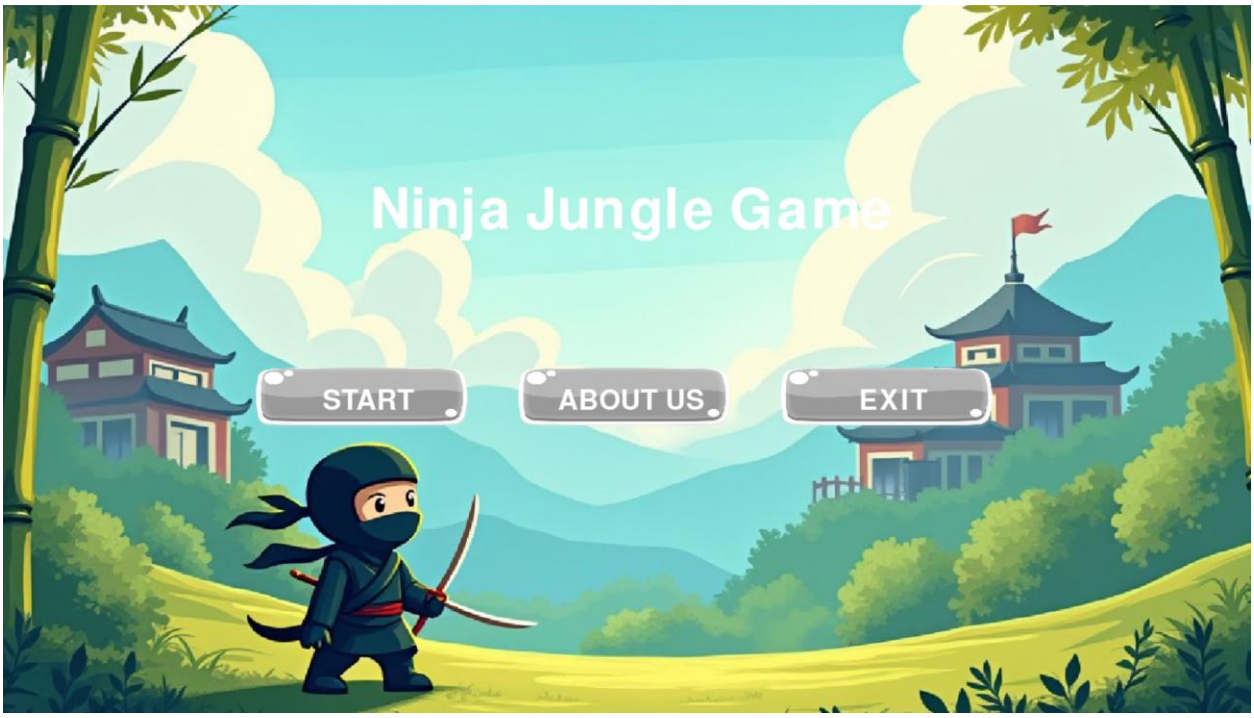
Ninja Jungle Game

Оваа игра е избработена од страна на Џевит Таири 211279 и Енес Сејфовски 211258 како проект по предметот Програмирање на видео игри на Факултетот за информатички науки и компјутерско инженерство во Скопје.

Целта на играта е корисникот со помош на влез од тастатурата да го контролира карактерот – Нинџа до излезот од лавиринтот. Покрај тоа корисникот може да собира парички кои може да ги употреби за купување на нов карактер.

Во прилог се функционалностите на играта со краток опис:

1. start_game_screen(screen, clock) – Ова е првата функционалност која ќе се повика при почеток на играта. Се рендерира позадината заедно со неколку копчиња. Копчињата служат за насочување кон други кориснички интерфејси во играта и за еизлез од играта. Се додаваат 2 аргуемнти screen(подлога на која ги прикажуваме елементите) и clock(часовникот за играта од pygame). Истовремено стартува и музиката во позадина.
   Како изгледа корисничкиот интерфејс за оваа функционалност:



   Дел од кодот на оваа функционланост:

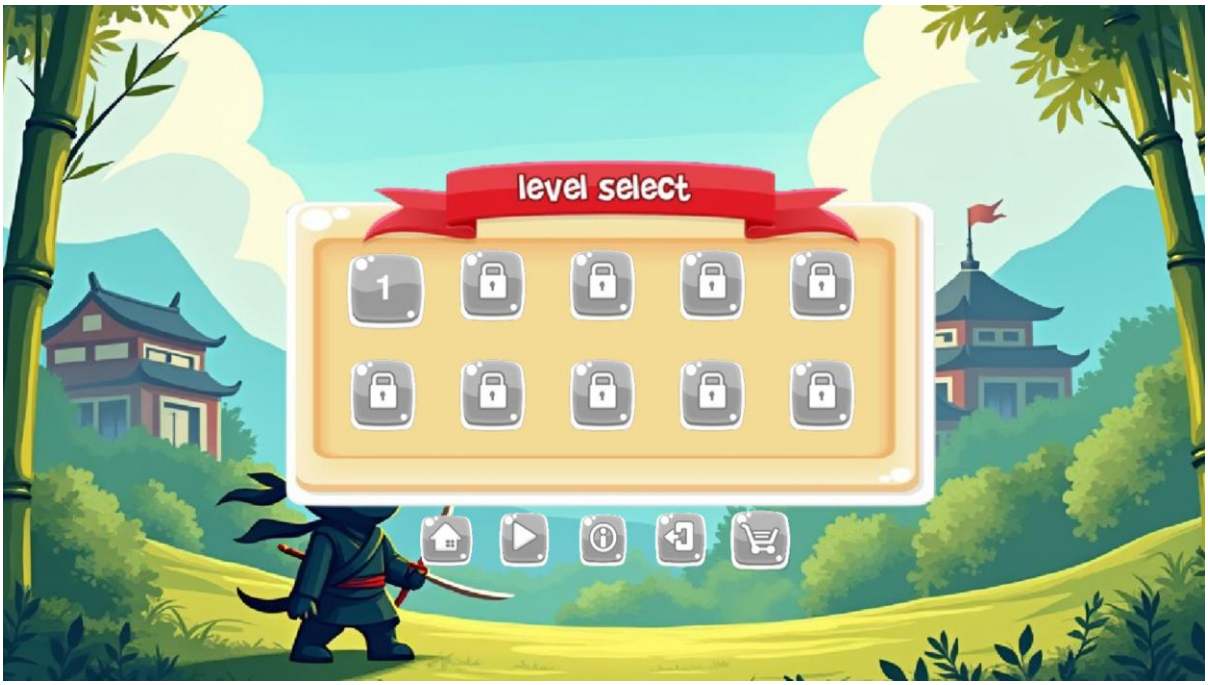
```python
def start_game_screen(screen, clock):  2 usages  Xhevitt +1
    background = load_image( filename: "background.jpg",  size: (WIDTH, HEIGHT))
    button_bg = load_image( filename: "button.png",  size: (300, 100))

    title_font = load_font( filename: 'freesansbold.ttf', size=100)
    button_font = load_font( filename: 'freesans.ttf', size=50)

    start_btn = pygame.Rect(WIDTH // 3 - 150, HEIGHT // 2, 300, 100)
    about_us_btn = pygame.Rect(WIDTH // 2 - 150, HEIGHT // 2, 300, 100)
    exit_btn = pygame.Rect(2 * WIDTH // 3 - 150, HEIGHT // 2, 300, 100)

    running = True
    while running:
        mouse_pos = pygame.mouse.get_pos()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            if event.type == pygame.MOUSEBUTTONDOWN:
                if start_btn.collidepoint(mouse_pos):
                    # Create level status with level 1 unlocked.
                    frames = [pygame.Surface((150, 150)) for _ in range(10)]
                    for frame in frames:
                        frame.fill((128, 128, 128))
                    level_status = {i: "locked" for i in range(1, 11)}
                    level_status[1] = "unlocked"
```

2. about_us_screen(screen, clock) – Функција до која можеме да достигнеме преку функцијата start_game_screen(screen, clock). Тука рендерираме позадина и неколку информации околу нас. Исто така рендерираме и копче преку кое кеорисникот ќе може да се врати на start_game_screen(screen, clock).
   Како изгледа корисничкиот интерфејс за оваа функционалност:

   Дел од кодот за оваа функционалност:

```python
def about_us_screen(screen, clock):  1 usage  sejfovskienes
    pygame.init()
    base_path = os.path.join(os.path.dirname(__file__), 'assets')
    background_image_path = os.path.join(base_path, 'images', 'background.jpg')
    info_box_image_path = os.path.join(base_path, 'images', 'info_box_.png')
    button_image_path = os.path.join(base_path, 'images', 'Button.png')
    background_image = pygame.image.load(background_image_path)
    background_image = pygame.transform.scale(background_image, screen.get_size())
    info_box_image = pygame.image.load(info_box_image_path)
    info_box_image = pygame.transform.scale(info_box_image,  size: (800, 500))
    button_image = pygame.image.load(button_image_path)
    button_image = pygame.transform.scale(button_image,  size: (200, 60))
    black = (0, 0, 0)
    white = (255, 255, 255)
    text_font = pygame.font.Font( name: None,  size: 40)
    button_font = pygame.font.Font( name: None,  size: 60)
    about_text = (
        "This game is made as a project for Programming Video Games course at FCSE. "
        "By students Xhevit Tairi and Enes Sejfovski."
    )
    text_lines = textwrap.wrap(about_text, width=30)
    line_height = text_font.get_height()
    total_text_height = len(text_lines) * line_height
    start_y = (screen.get_height() - total_text_height) // 2
    button_rect = button_image.get_rect(center=(screen.get_width() // 2, screen.get_height() - 200))
```

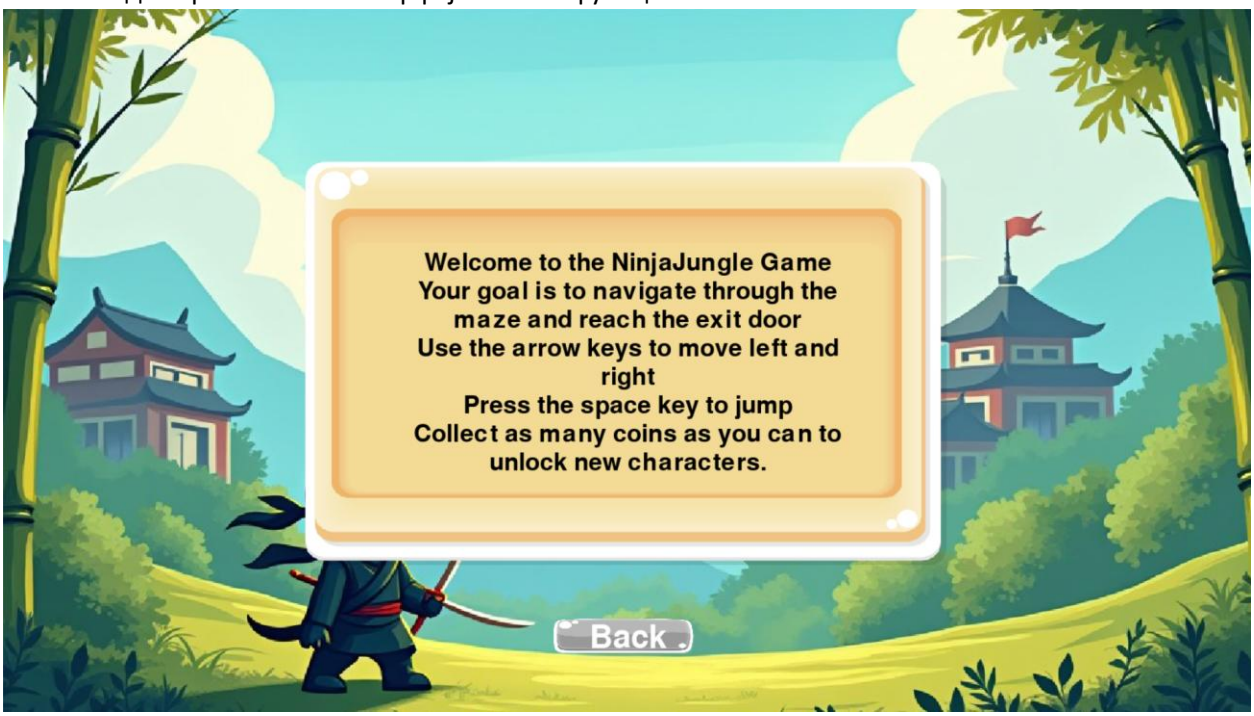
3. level_selection_screen(screen, clock) – До оваа функционалност корисникот може да дојде преку клик на Start копчето од start_game_screen(screen, clock). Во оваа функција рендерираме копчиња за избор на ниво. Иницијално отворено е само првото ниво. Како што корисникот ќе го завршува секое ниво следното ќе му биде достапно. Освен тоа под блокот за избор на ниво имаме 5 копчиња кои имаат функционланост за следното(од лево кон десно): враќање кон home screen, повторно рендерирање на level selection screen, информации како се игра, излез од играта, продавница каде што може да се отвори следниот карактер. Во оваа функција користиме повеќен event-listeners со кои следиме дали има колизија на кликот со некое од копчињата.
   Како изгледа корисничкиот интерфејс за оваа функционалност:




```
                for level, rect in level_buttons:
                    if rect.collidepoint(mouse_pos) and level_status[level] == "unlocked":
                        print(f"Level {level} clicked!")
                        if level == 1:
                            completed = level_one_screen(screen, clock, level_status)
                            if completed:
                                level_status[2] = "unlocked"  # Unlock Level 2
                        elif level == 2:
                            completed = level_two_screen(screen, clock, level_status)
                            if completed:
                                level_status[3] = "unlocked"
                        elif level == 3:
                            completed = level_three_screen(screen, clock, level_status)
                            if completed:
                                level_status[4] = "unlocked"
                        elif level == 4:
                            completed = level_four_screen(screen, clock, level_status)
                            if completed:
                                level_status[5] = "unlocked"
                        elif level == 5:
                            completed = level_five_screen(screen, clock, level_status)
                            if completed:
                                level_status[6] = "unlocked"
```

4. how_to_play_screen(sceen, clock) – Функционалост слична на about_us_screen(screen, clock). Каде што ги рендерираме информациите за тоа како се игра играта.

Како изледа корисничкиот интерфејс за оваа функционалност:



Дел од кодот за оваа функционланост:

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
            if button_rect.collidepoint(event.pos):
                return
    screen.blit(background_image, (0, 0))
    screen.blit(info_box_image, info_box_rect.topleft)
    y_offset = text_start_y
    for line in text_lines:
        text_surface = text_font.render(line, antialias: True, black)
        text_rect = text_surface.get_rect(center=(info_box_rect.centerx, y_offset))
        screen.blit(text_surface, text_rect)
        y_offset += line_height + 5
    screen.blit(button_image, button_rect.topleft)
    button_text = button_font.render( text: "Back", antialias: True, white)
    button_text_rect = button_text.get_rect(center=button_rect.center)
    screen.blit(button_text, button_text_rect)
    pygame.display.flip()
    clock.tick(60)

```

5. store_screen(screen, clock) - Функционланост каде се прикажува моментално избрантиот карактер и карактерот кој може да се купи. Вториот карактер може да се купи со собрани 25 парички. Слика од корисничкиот интерфејс на оваа функционалност:

Дел од кодот на оваа функционалност:

```
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        elif event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
            if back_button_rect.collidepoint(event.pos):
                return
            # Default card: always allow selecting the free/default skin.
            if default_button_rect.collidepoint(event.pos):
                import game_data
                game_data.selected_skin = "ninja"
                print("Default skin selected.")
            # Alternative card:
            if alternative_button_rect.collidepoint(event.pos):
                import game_data
                if not game_data.alternative_skin_bought:
                    if game_data.coins_collected >= 25:
                        game_data.alternative_skin_bought = True
                        game_data.selected_skin = "ninjagirlnew"
                        print("Alternative skin bought and selected.")
                    else:
                        print("Not enough coins to buy the alternative skin.")
                else:
                    game_data.selected_skin = "ninjagirlnew"
```

При клик на секое од копчињата за отворено ниво, податоците за тоа ниво се испраќаат на функцијата run_level(screen, clock, level_background, world_data, level_status, next_level):

Оваа функција ги користи информациите за мапата и ја го рендерира изгледот на секое ниво. Користејќи ги класите: Player, World, Blob. Според карактерите во матрицата за секое ниво го позиционира играчот на почетна позиција и ако нивото е >=7 Blobs кои се појавуваат се појавуваат со рандом позиција преку функцијата place_random_blobs_fair(num_blobs, base_world_data). Изглед од ниво 7: