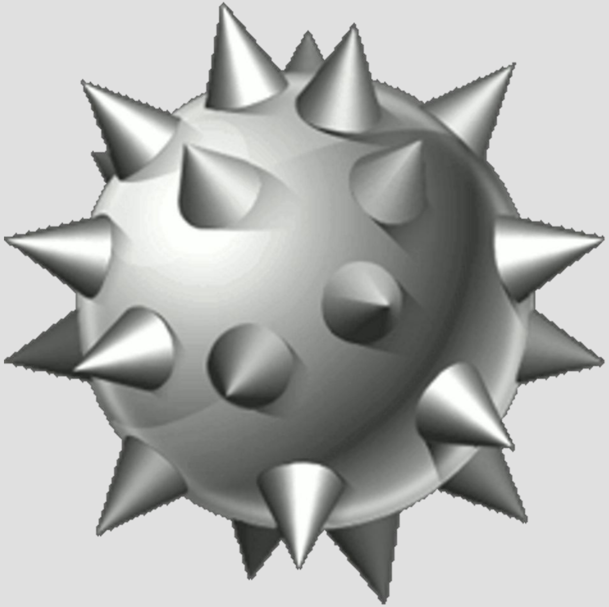


Minesweeper

K. Bém, P. Papiňák, L. Pražáková

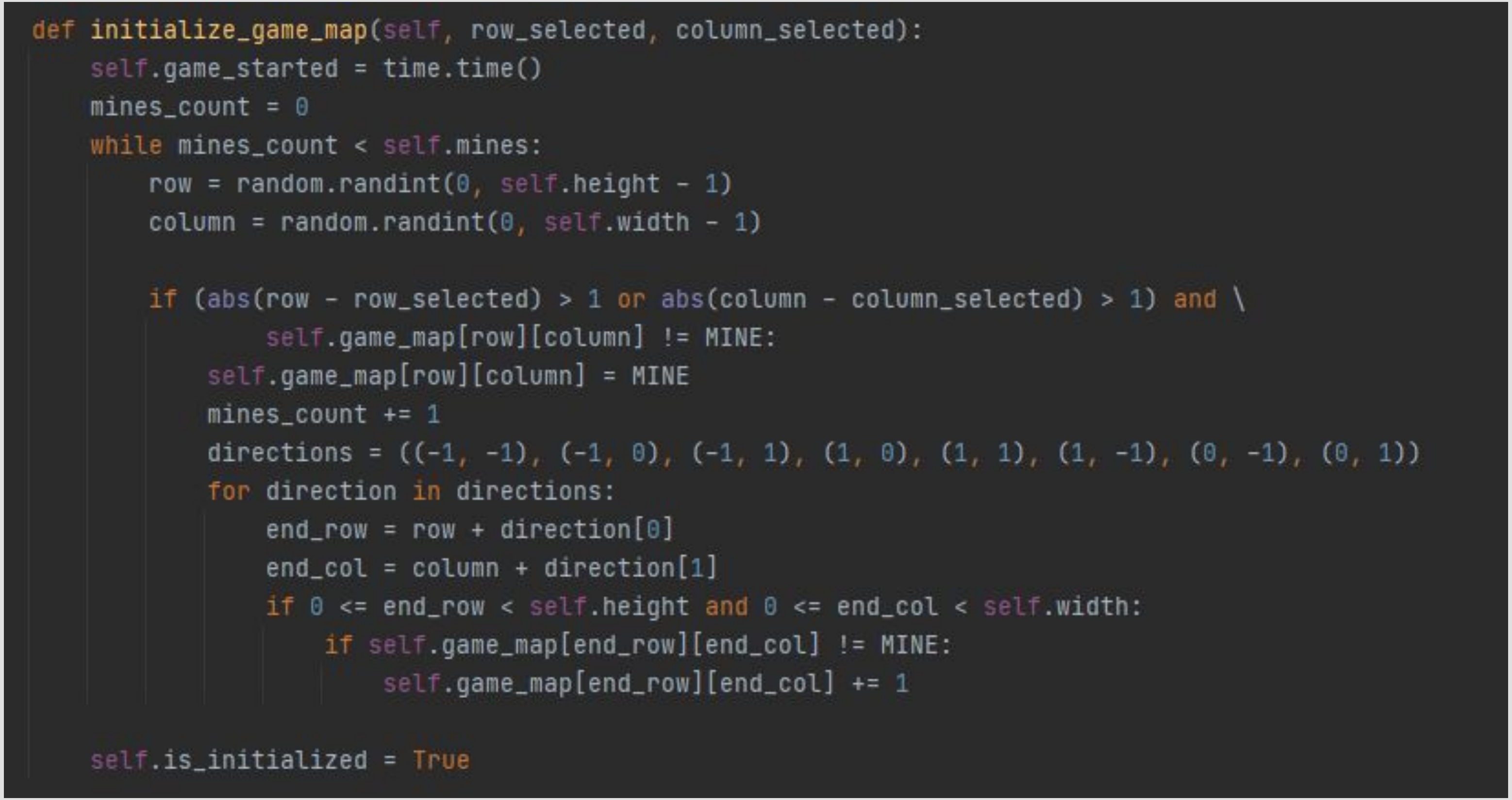
Kompetence

Karel Bém - architektura, game state, game loop

Petr Papiňák - start screen, end screen, testing

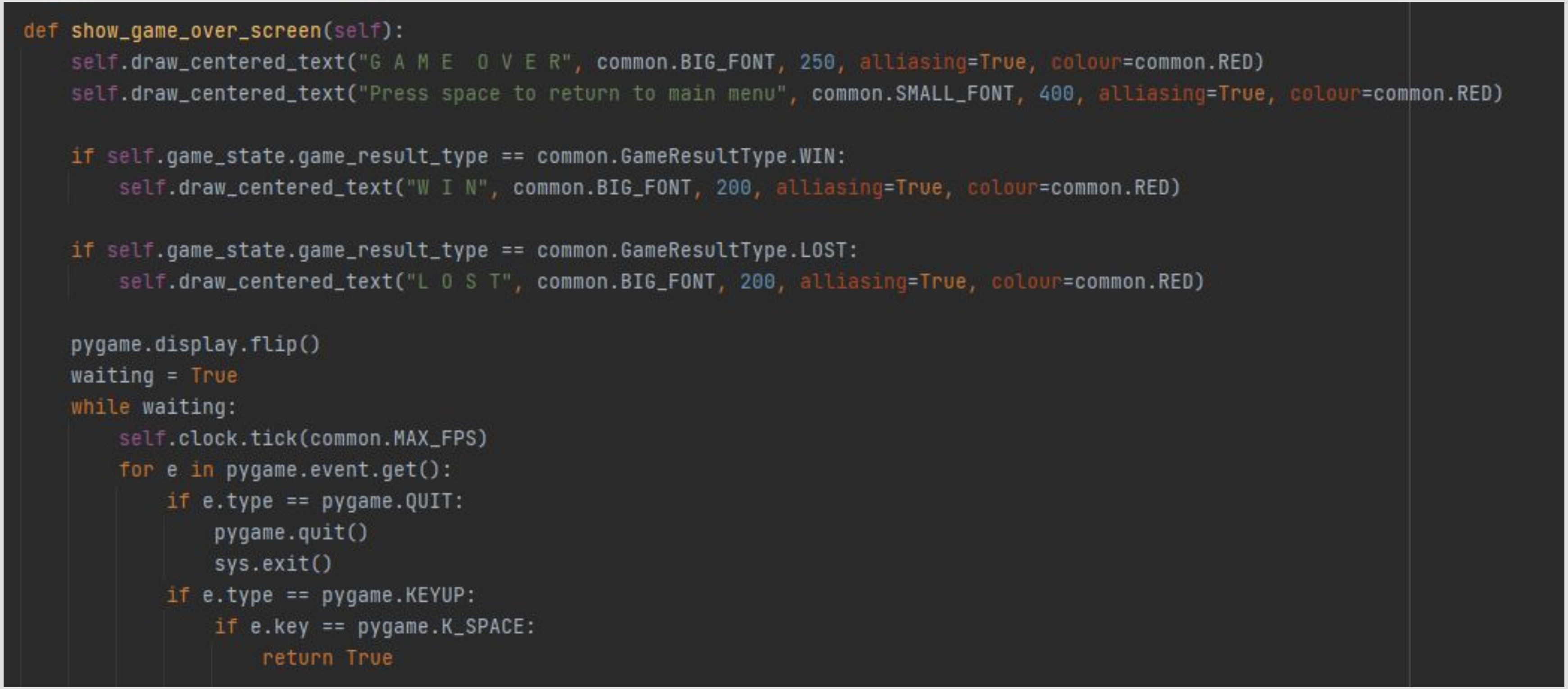
Lucie Pražáková - grafika, hudba, sprites

Karel Bém

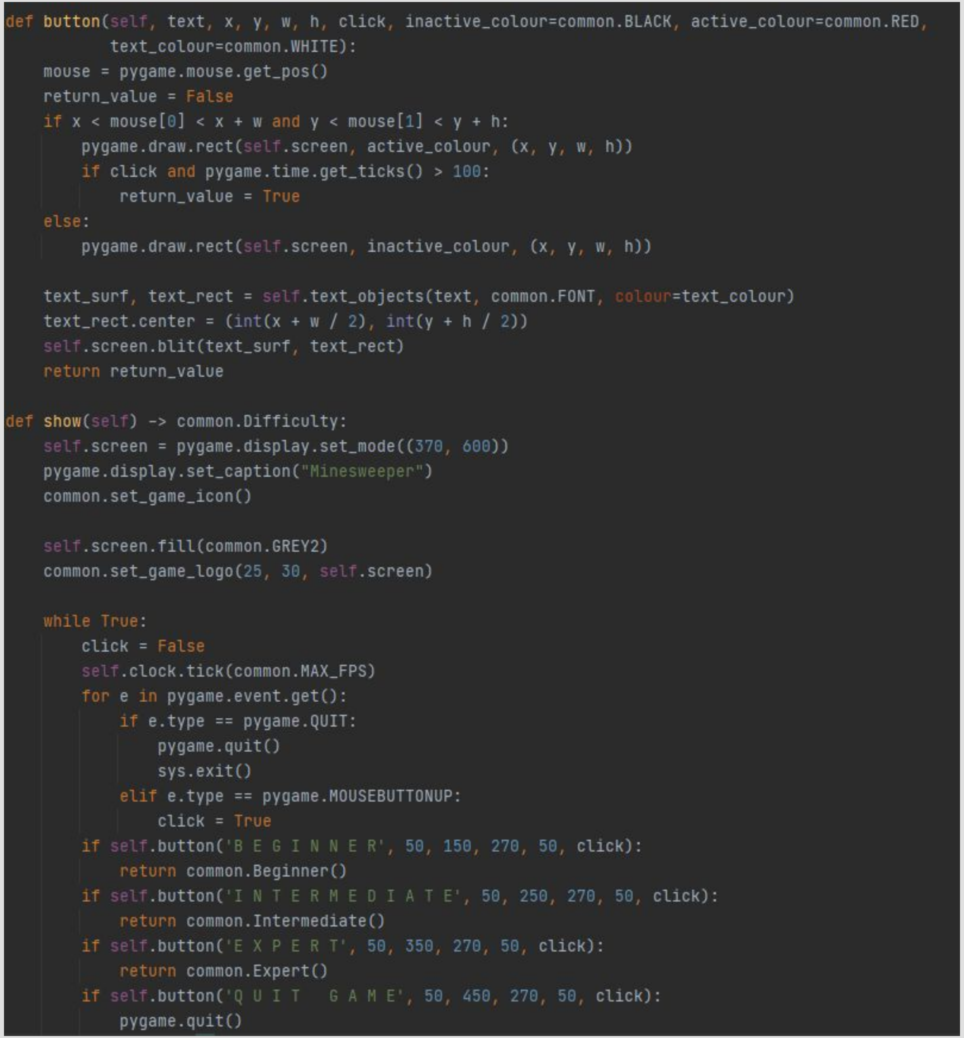


```python
def initialize_game_map(self, row_selected, column_selected):
    self.game_started = time.time()
    mines_count = 0
    while mines_count < self.mines:
        row = random.randint(0, self.height - 1)
        column = random.randint(0, self.width - 1)

        if (abs(row - row_selected) > 1 or abs(column - column_selected) > 1) and \
                self.game_map[row][column] != MINE:
            self.game_map[row][column] = MINE
            mines_count += 1
            directions = ((-1, -1), (-1, 0), (-1, 1), (1, 0), (1, 1), (1, -1), (0, -1), (0, 1))
            for direction in directions:
                end_row = row + direction[0]
                end_col = column + direction[1]
                if 0 <= end_row < self.height and 0 <= end_col < self.width:
                    if self.game_map[end_row][end_col] != MINE:
                        self.game_map[end_row][end_col] += 1

    self.is_initialized = True
```

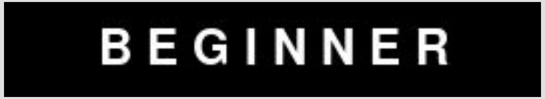
Petr Papiňák

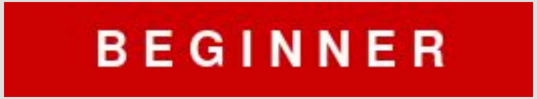


```
def show_game_over_screen(self):
    self.draw_centered_text("G A M E  O V E R", common.BIG_FONT, 250, alliasing=True, colour=common.RED)
    self.draw_centered_text("Press space to return to main menu", common.SMALL_FONT, 400, alliasing=True, colour=common.RED)

    if self.game_state.game_result_type == common.GameResultType.WIN:
        self.draw_centered_text("W I N", common.BIG_FONT, 200, alliasing=True, colour=common.RED)

    if self.game_state.game_result_type == common.GameResultType.LOST:
        self.draw_centered_text("L O S T", common.BIG_FONT, 200, alliasing=True, colour=common.RED)

    pygame.display.flip()
    waiting = True
    while waiting:
        self.clock.tick(common.MAX_FPS)
        for e in pygame.event.get():
            if e.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if e.type == pygame.KEYUP:
                if e.key == pygame.K_SPACE:
                    return True
```

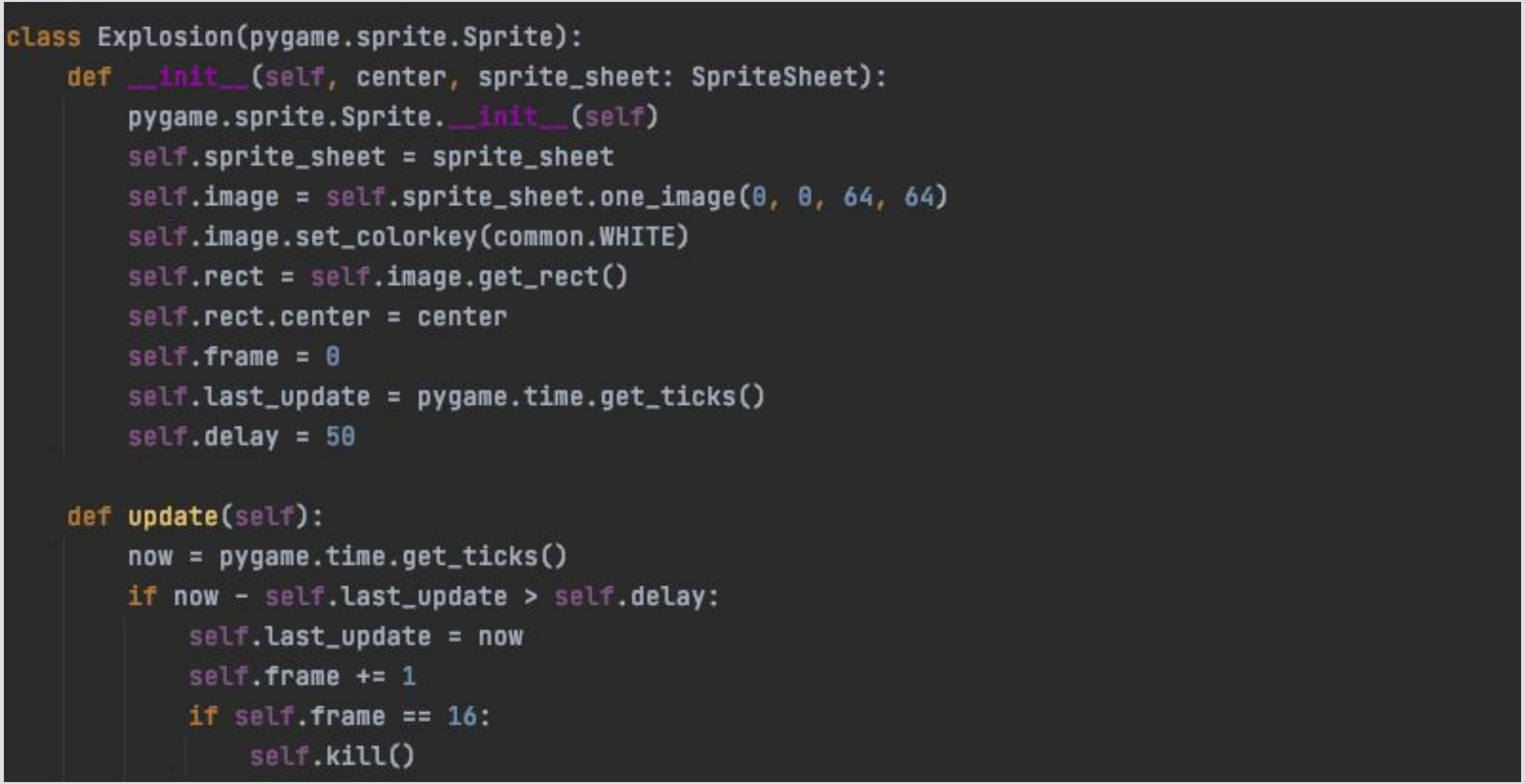


```python
def button(self, text, x, y, w, h, click, inactive_colour=common.BLACK, active_colour=common.RED,
           text_colour=common.WHITE):
    mouse = pygame.mouse.get_pos()
    return_value = False
    if x < mouse[0] < x + w and y < mouse[1] < y + h:
        pygame.draw.rect(self.screen, active_colour, (x, y, w, h))
        if click and pygame.time.get_ticks() > 100:
            return_value = True
    else:
        pygame.draw.rect(self.screen, inactive_colour, (x, y, w, h))

    text_surf, text_rect = self.text_objects(text, common.FONT, colour=text_colour)
    text_rect.center = (int(x + w / 2), int(y + h / 2))
    self.screen.blit(text_surf, text_rect)
    return return_value

def show(self) -> common.Difficulty:
    self.screen = pygame.display.set_mode((370, 600))
    pygame.display.set_caption("Minesweeper")
    common.set_game_icon()

    self.screen.fill(common.GREY2)
    common.set_game_logo(25, 30, self.screen)

    while True:
        click = False
        self.clock.tick(common.MAX_FPS)
        for e in pygame.event.get():
            if e.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            elif e.type == pygame.MOUSEBUTTONUP:
                click = True
        if self.button('B E G I N N E R', 50, 150, 270, 50, click):
            return common.Beginner()
        if self.button('I N T E R M E D I A T E', 50, 250, 270, 50, click):
            return common.Intermediate()
        if self.button('E X P E R T', 50, 350, 270, 50, click):
            return common.Expert()
        if self.button('Q U I T   G A M E', 50, 450, 270, 50, click):
            pygame.quit()
```





Lucie Pražáková



```python
class Explosion(pygame.sprite.Sprite):
    def __init__(self, center, sprite_sheet: SpriteSheet):
        pygame.sprite.Sprite.__init__(self)
        self.sprite_sheet = sprite_sheet
        self.image = self.sprite_sheet.one_image(0, 0, 64, 64)
        self.image.set_colorkey(common.WHITE)
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.frame = 0
        self.last_update = pygame.time.get_ticks()
        self.delay = 50

    def update(self):
        now = pygame.time.get_ticks()
        if now - self.last_update > self.delay:
            self.last_update = now
            self.frame += 1
            if self.frame == 16:
                self.kill()
```

Díky za pozornost

K. Bém, P. Papiňák, L. Pražáková