

Jan Maczuga

Oleksandr Hrekov

Tomasz Kielar

Bartosz Kopyś

[Link GitHub:](https://github.com/yotmand18/Projekt-Podstawy-Informatyki.git) <https://github.com/yotmand18/Projekt-Podstawy-Informatyki.git>

Gra platformowa Wiedźmin: Gra platformowa w Stylu Mario, gracz ma za zadania przejście przez poziom pokonując przeciwników

Wykorzystane biblioteki: SFML, vector, fstream, sstream, algorithm, cmath, filesystem, ctime, unordered_map, random

Wybrane funkcjonalności z kodem:

Mechaniki walki:

```
void Player::takeDamage(int amount){
    if(this->damageCooldownClock.getElapsedTime().asSeconds() < this->damageCooldown)
        return;

    std::cout << "Player health: " << this->health << "\n";
    this->lostHealth = true;

    if(this->health - amount <= 0)
        this->health = 0;
    else
        this->health -= amount;

    this->damageCooldownClock.restart();
    this->sprite.setColor(sf::Color::Red);
}
```

```
void Player::gainHealth(int amount){
    if(this->health + amount > this->maxHealth)
        this->health = this->maxHealth;
    else
        this->health += amount;
    this->gainedHealth = true;
    this->sprite.setColor(sf::Color::Green);
}
```

Mechanika progresji gracza:

```

inline void Player::LevelUp() {
    this->level++;
    this->attack = 2 * this->getLevel();
    this->health = 20 * this->getLevel(); //Heal on levelup?
    this->maxHealth = 20 * this->getLevel();
    this->modHealthPotion(1); this->modSpeedPotion(1); this-
>modAttackPotion(1);

}

```

Poruszanie się:

```

void Player::updateMovement() {
    sf::Vector2f velocity = this->physics->getVelocity();

    // Determine facing direction based on horizontal velocity
    if (velocity.x > 0.f)
        this->facingRight = true;
    else if (velocity.x < 0.f)
        this->facingRight = false;

    // We only change the state to movement/idle if we aren't currently
    attacking
    // If we are attacking, we let the animation finish first
    if (this->animState != ATTACKING) {
        if (std::abs(velocity.x) > 0.1f) {
            // Both running and walking now trigger the same animation logic
            this->animState = (velocity.x > 0.f) ? MOVING_RIGHT : MOVING_LEFT;
        } else {
            this->animState = IDLE;
        }
    }
}

```

Pobieranie Inputu, na przykładzie eliksiru:

```

if (this->input->isAction("HEALTH_POTION") && !this->player->isAttacking()
&& this->player->timeHealthPotion == 0) {
    if (this->player->getHealthPotions() > 0) {
        this->player->timeHealthPotion = clock();
        this->player->gainHealth(this->player->getMaxHealth() / 2);
        this->player->modHealthPotion(-1);
    }
}

```