

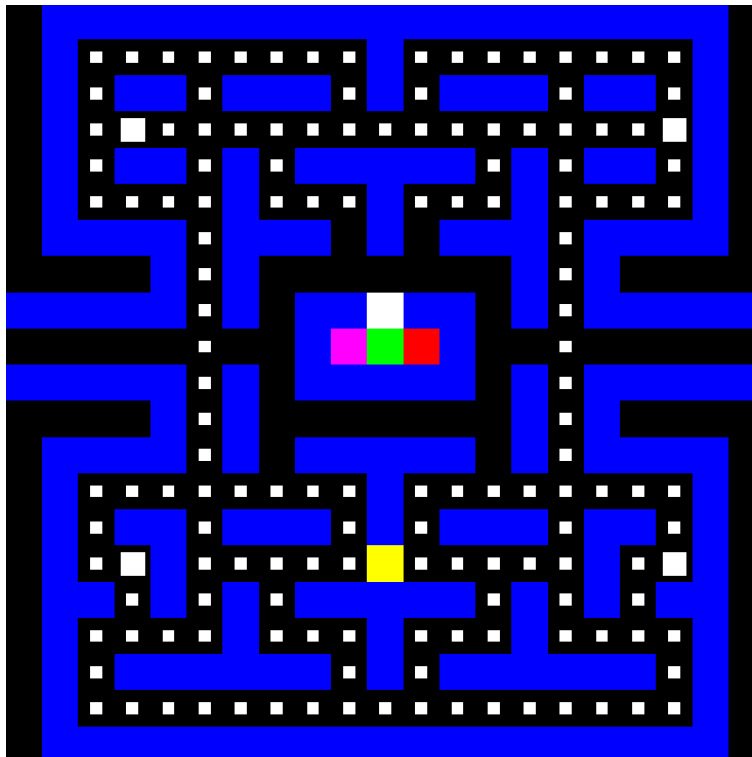
PacMan

Obsah

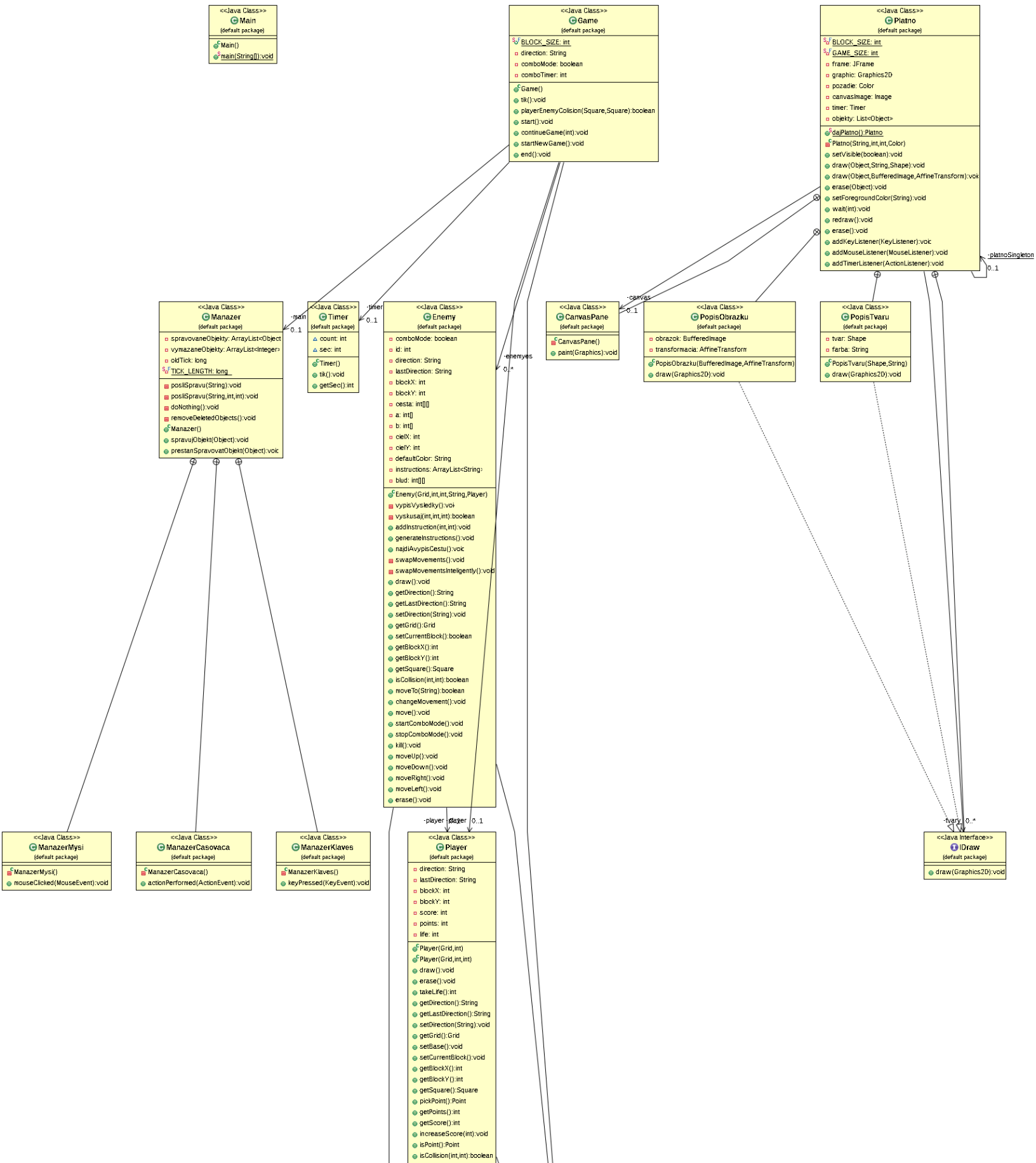
PacMan	1
Ovládanie	2
UML.....	3
Zaujímavé metódy	5
Pohyb nepriateľov.....	6

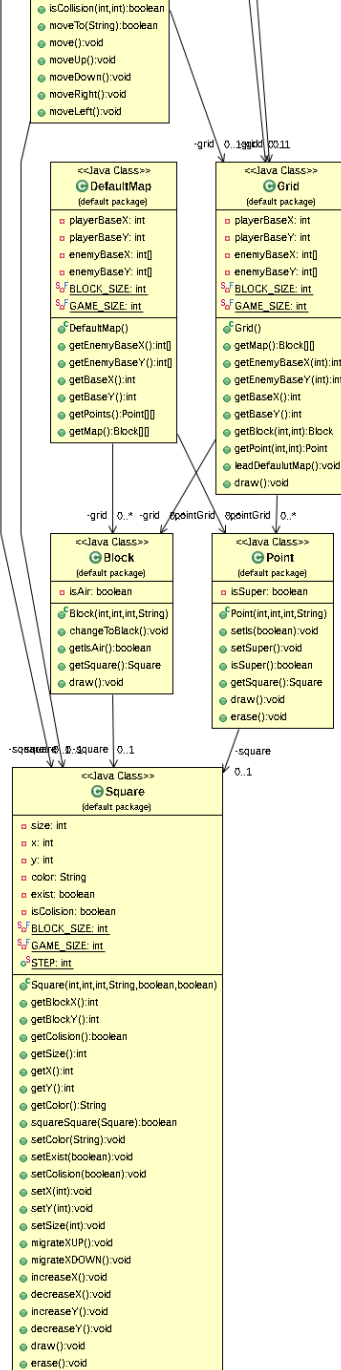
Ovládanie

- Pohyb – šípky na klávesnici
- Hra začne stlačením jednej zo šípok
- Hra sa ukonči stlačením klávesy esc
- Hráč ma 3 životy
- Na hracej ploche sa nachádza 151 bodov, za každý pozbieraný bod hráč dostane 200 skóre
- Keď hráč pozbiera všetky body načíta sa nových 151 blokov a pokračuje sa v hre
- 4 z bodov sú špeciálne, od obvyčajných väčšie, keď ich hráč pozbiera dostane na pár sekúnd schopnosť zabiť nepriateľa
- Zabitie je kolízia hráča s nepriateľom
- Po zabití nepriateľa sa nepriateľ presúva na štartovnú pozíciu a pokračuje v hre
- Po zabití hráča sa hráč aj všetci nepriatelia presunu na štartovnú pozíciu



UML





Zaujímavé metódy

```
/**
 * kontroluje kolíziu s tromi blokmi pred hracom
 */
public boolean moveTo(String direction){
    if(direction.equals("up")){
        this.square.increaseY();
        if(this.isCollision(0, 1) || this.isCollision(1, 1) || this.isCollision(-1, 1)){
            this.square.decreaseY();
            return false;
        }
        return true;
    }
    if(direction.equals("do")){
```

```
/**
 * kolízia stvorec stvorec
 */
public boolean squareSquare(Square square){
    //ak najde blok s ktorým sa nerobi kolízia
    if(square.isCollision == false)
        return false;
    //vrati true ak naslo kolíziu
    return (square.x < this.x + this.size
        && square.y < this.y + this.size
        && square.x + square.size > this.x
        && square.y + square.size > this.y);
}
```

```
/**
 * vykona pohyb a ak nastala kolízia tak pokračuje
 * v predoslom pohybe
 */
public void move(){
    this.setCurrentBlock();
    if(this.moveTo(this.getDirection()) == false)
        this.moveTo(this.getLastDirection());

    draw();
}
```

Pohyb nepriateľov

1. Pohyb nepriateľov sa vykonáva pomocou rekurzívneho vyhľadávania v poli
2. Vyhľadáva sa od pozície nepriateľa po pozíciu hráča ale vyhľadávanie sa predčasne ukončí ak nájdená cesta prekročí dĺžku 20
3. z cesty sa vygenerujú inštrukcie (hore, dole, doprava, doľava) a zapíšu do arraylistu
4. vždy keď sa zmení blok na ktorom sa nepriateľ nachádza načíta si novú inštrukciu
5. keď sa vyprázdni zásobník inštrukcií vygeneruje si nové...
6. pre lepšiu náhodnosť pohybov sa pri každom generovaní inštrukcií pomieša poradie v akom nepriateľ skúša bloky na ktoré sa môže pohnúť pri rekurzívnom vyhľadávaní