

Projekt z objektovo orientovaného programovania LS2025

Zámer projektu

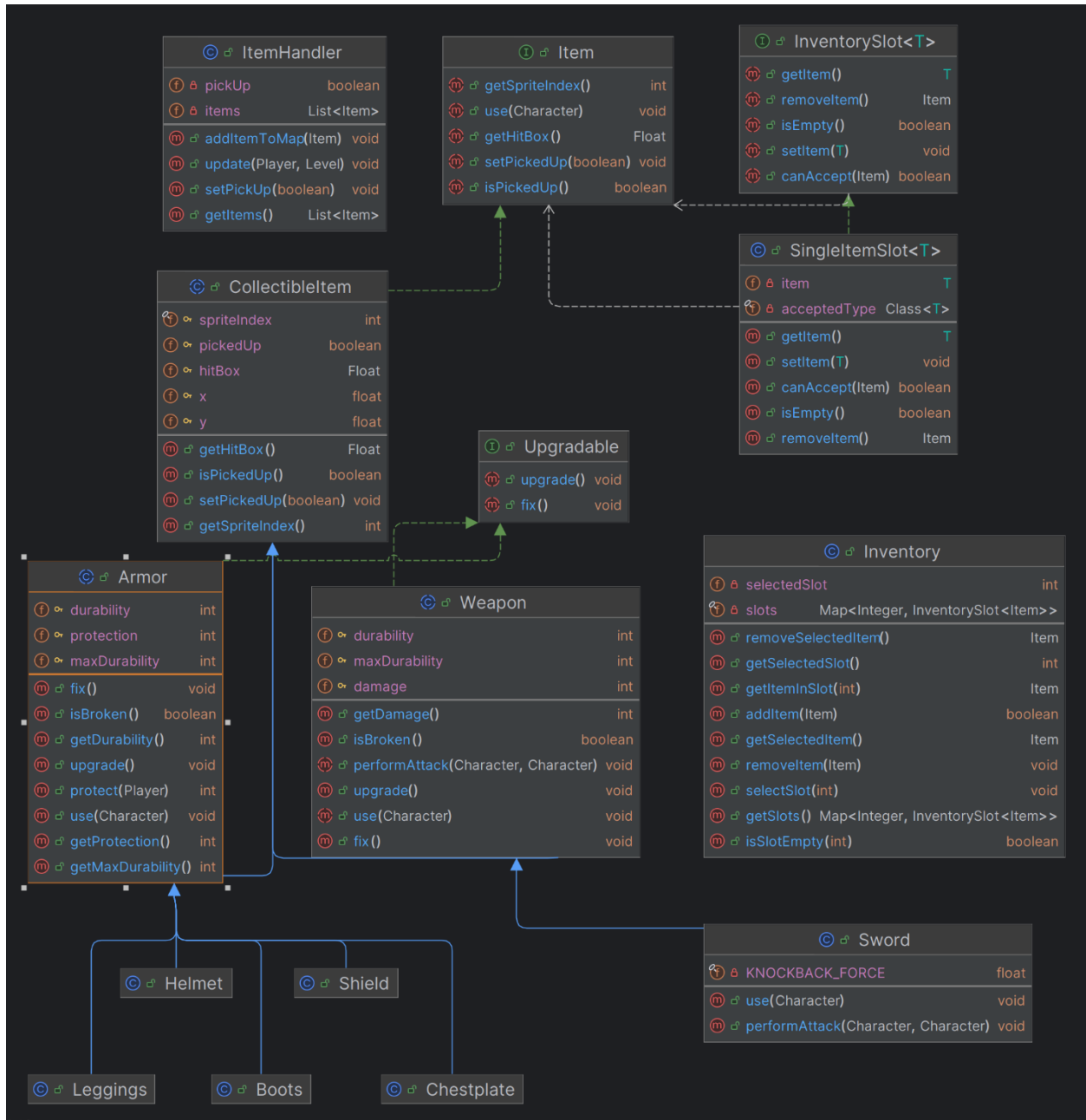
Zmena (nebolo implementovane)

Podstata hry spočíva v tom, že hráč prechádza rôznymi lokáciami. Cieľom hry je prejsť všetky lokácie. Každá lokácia pozostáva z niekoľkých úrovní. Každá úroveň sa skladá z dvoch častí: parkouru a boja s nepriateľmi. V priebehu hry musí hráč dokončiť obe tieto časti, aby sa posunul do ďalšej úrovne. **Na konci každej lokácie musí hráč poraziť bossa, aby mohol postúpiť do ďalšej lokácie.** V každej fáze hráč získa určitý počet XP, za ktoré bude môcť zlepšovať určité zručnosti zbraní **alebo získavať špeciálne schopnosti.** Taktiež platí, že **čím ďalej sa hráč posunie, tým silnejší budú nepriatelia, a zároveň sa bude zvyšovať náročnosť samotného prechodu úrovňou.** Ak hráč počas prechádzani úrovňou stráca všetko svoje zdravie, zomrie a objaví sa na checkpointe.

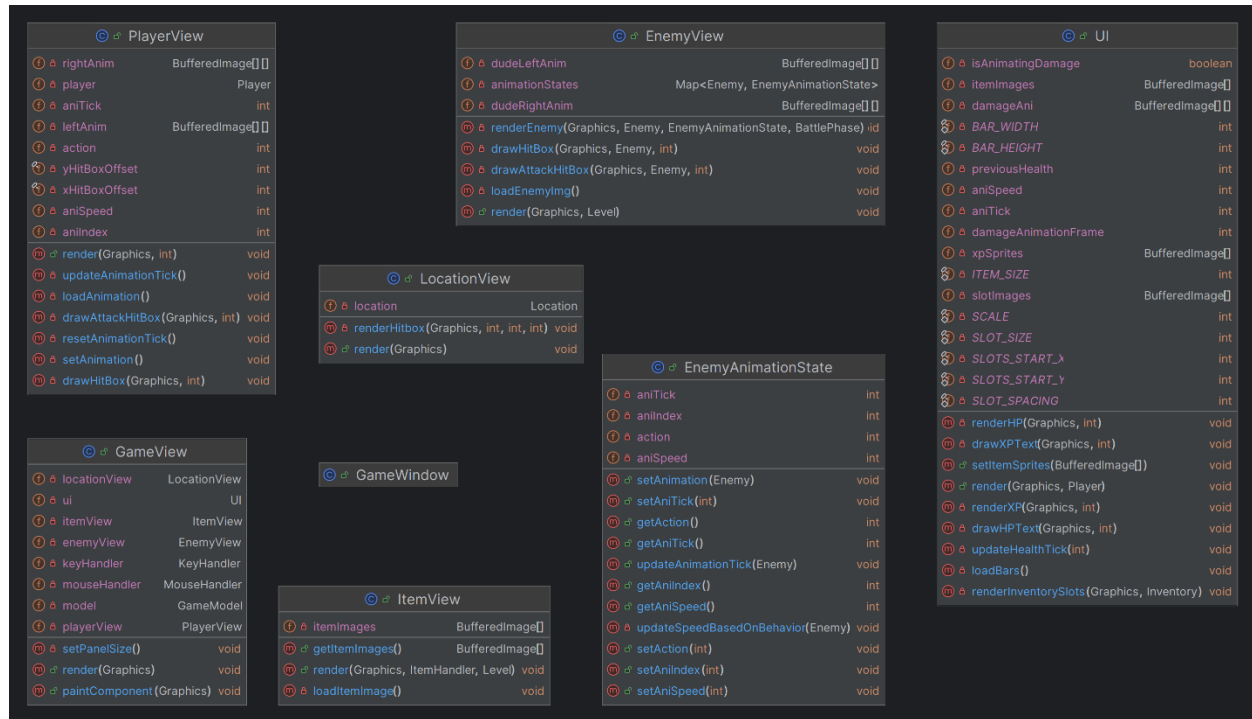
Žáner: *Action Platform Game.*

UML diagram tried

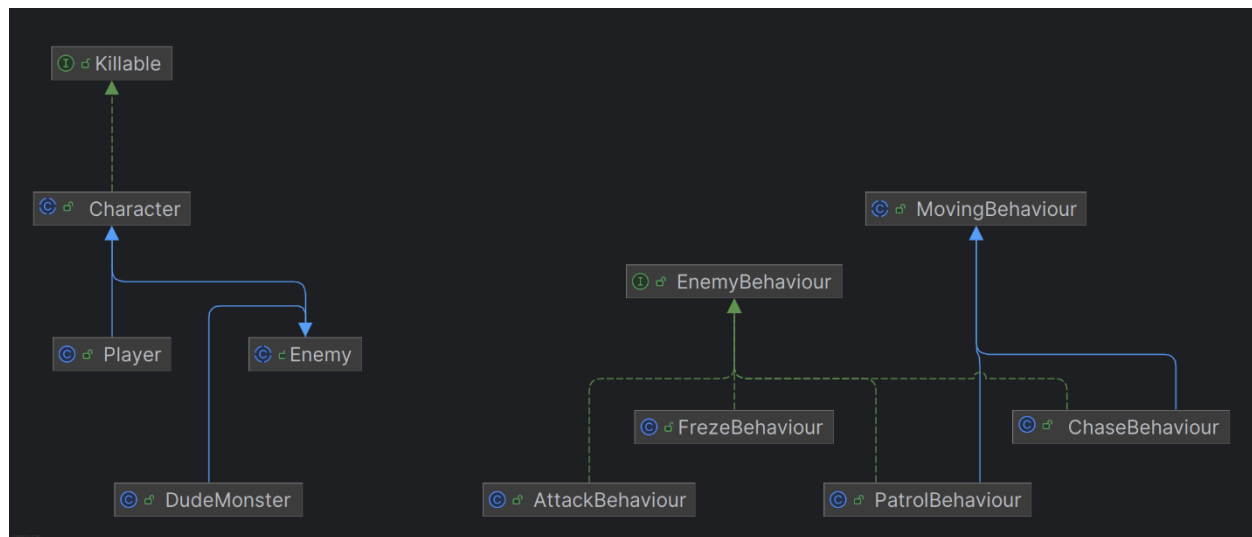
src/core



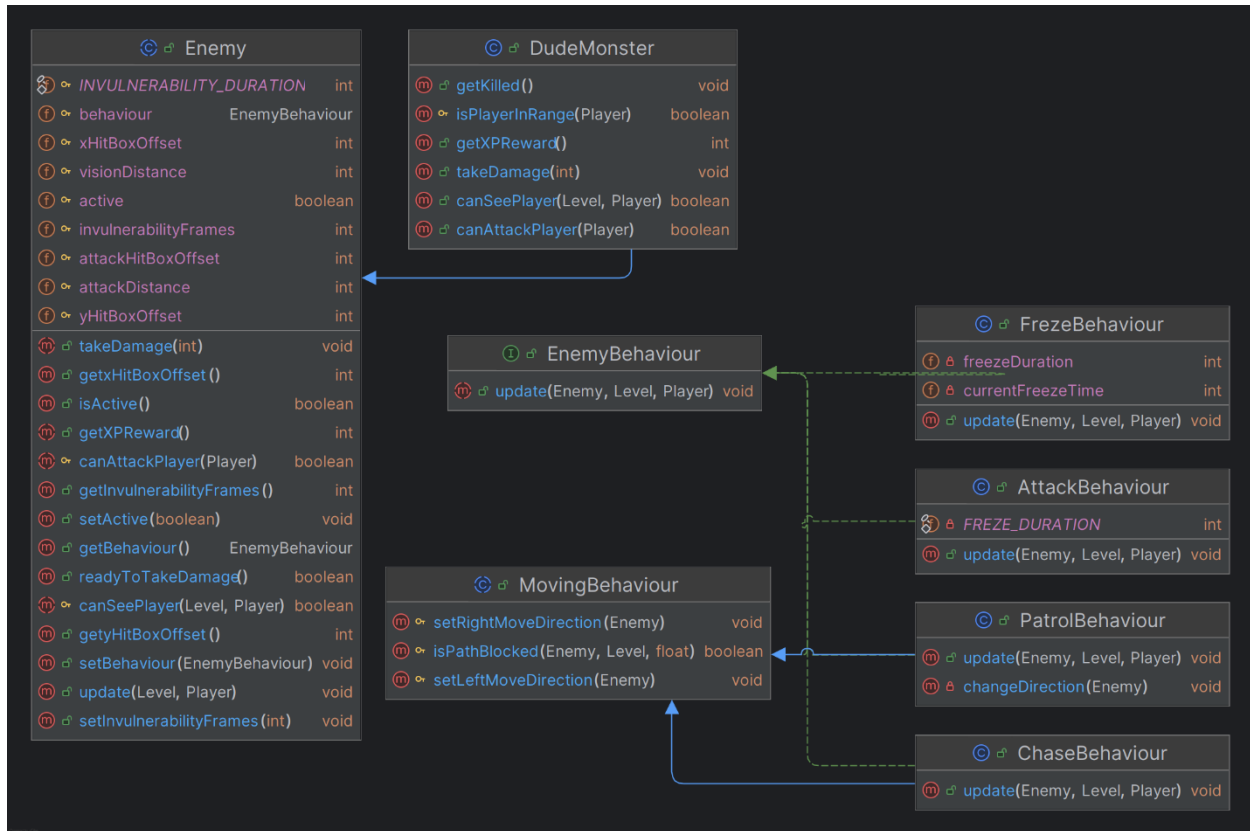
src/gui



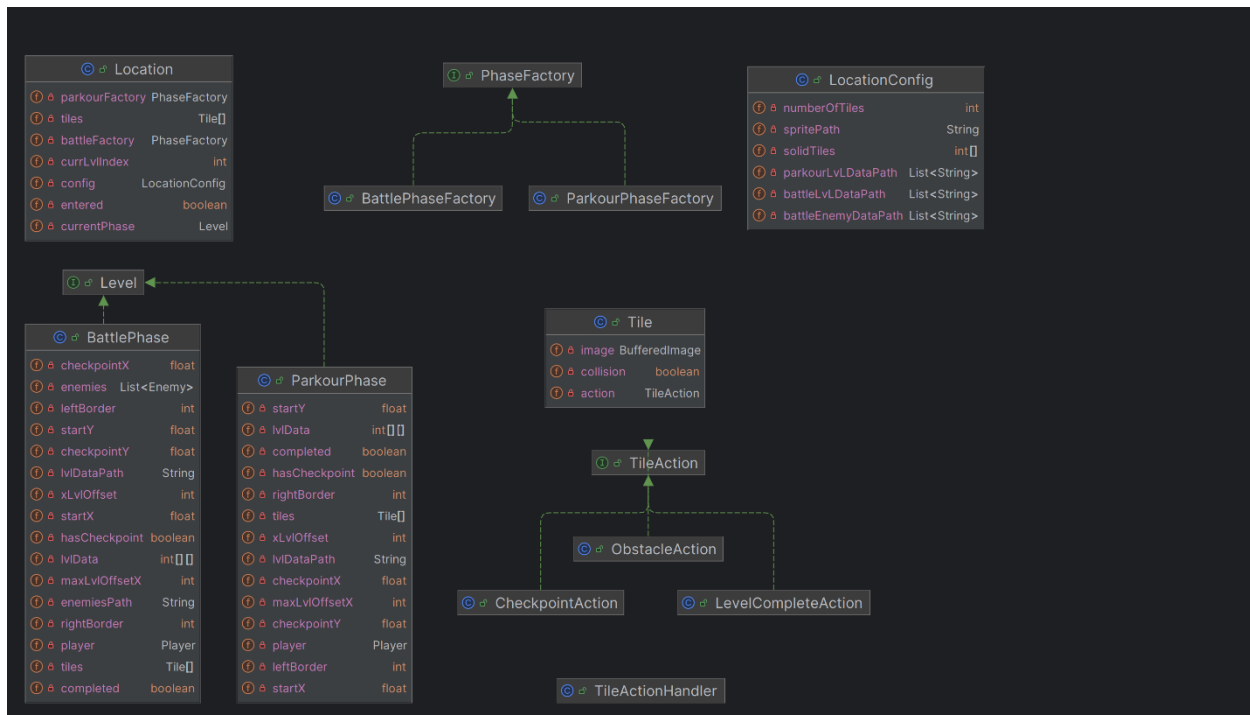
src/entity



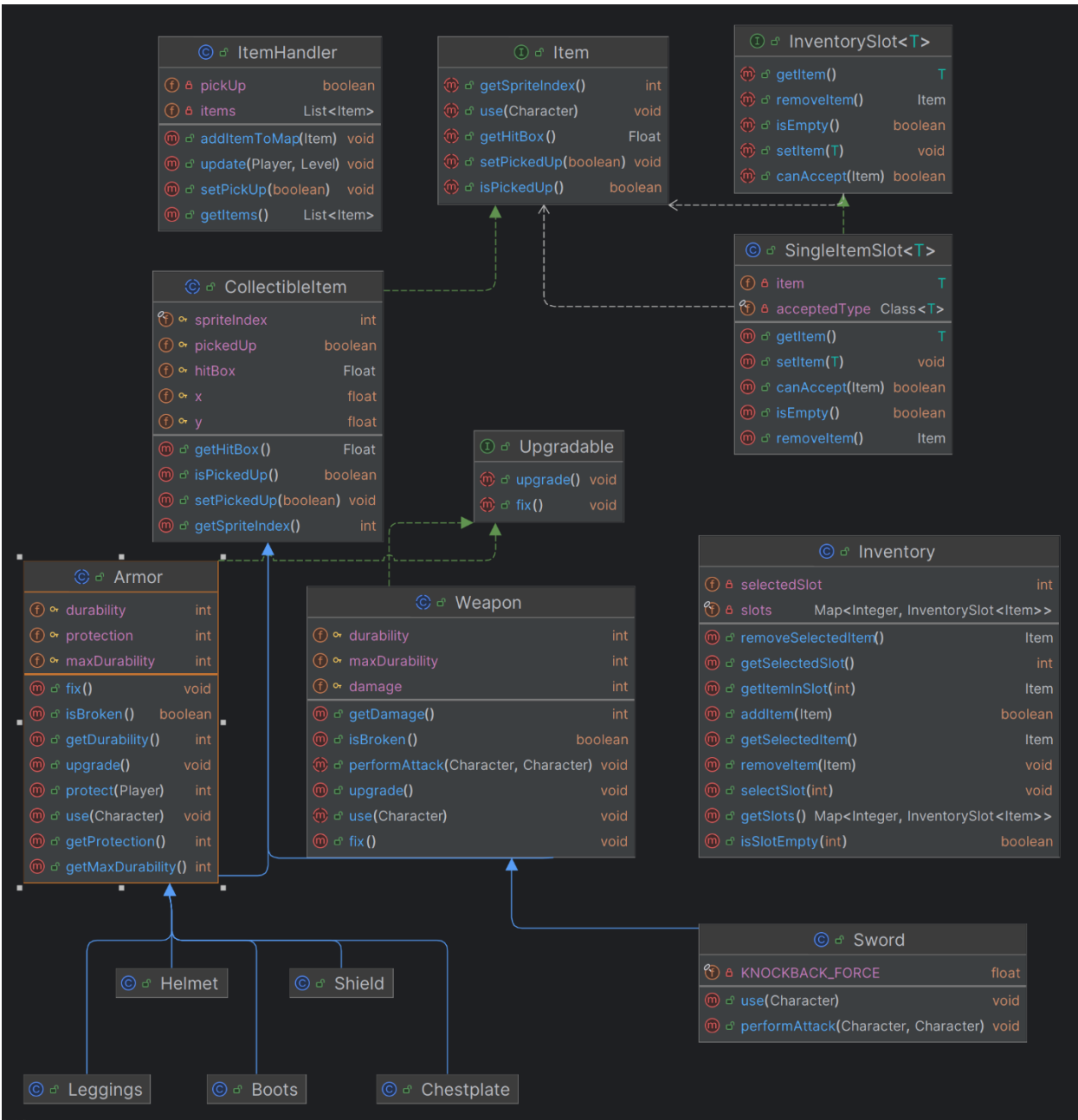
src/entity/enemies



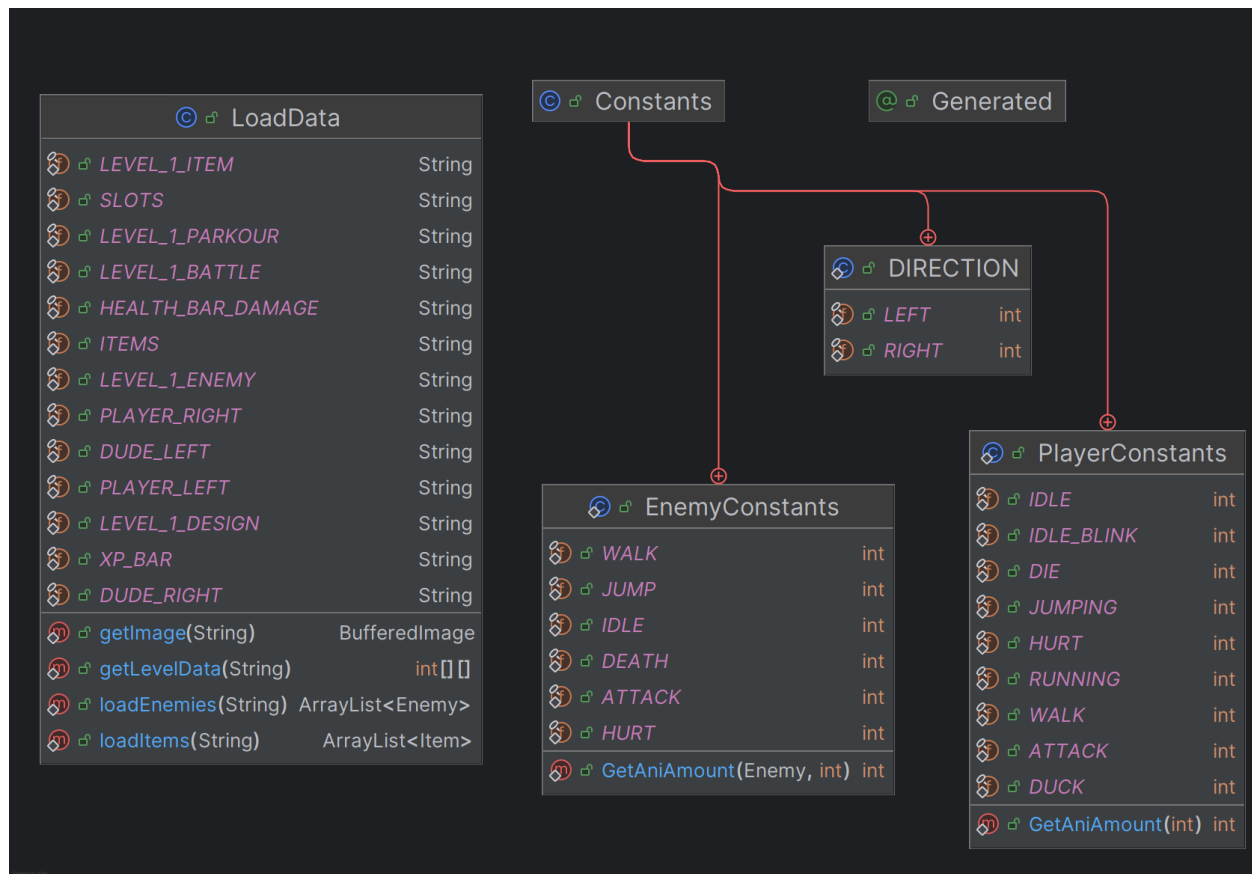
src/map



src/item



src/util



Použité knihovny

Swing, Awt

Vyjadrenie sa k splneniu podmienok (nutnych, aj dalsich)

- **Obsahovat dedenie a rozhrania:**

- ```
public abstract class Character implements Killable {
```
- ```
public class Player extends Character{
```

- **Pouzitie zakladnych OOP principov (enkapsulacia, dedenie, polymorfizmus, abstrakcia)**

- *Enkapsulacia*

- src/core/GameController:

- ```
private GameModel model;
```
- ```
private GameView view;
```
- ```
private GameWindow window;
```
- ```
public GameModel getModel() {
```
- ```
 return model;
```
- ```
}
```
- ```
public GameView getView() {
```
- ```
    return view;
```
- ```
}
```
- ```
public GameWindow getWindow() {
```
- ```
 return window;
```
- ```
}
```

- *Dedenie:*

- src/entity/enemies

- ```
public class PatrolBehaviour extends MovingBehaviour
```

- *Polymorfizmus:*

- src/map/ (Runtime polimorfizmus)

- ```
private Level currentPhase;
```
- ```
private void loadParkourPhase(Player player) {
```
- ```
    String levelPath = config.getParkourLvLDataPath(currLvlIndex);
```
- ```
 currentPhase = parkourFactory.createPhase(levelPath, null, tiles);
```
- ```
    currentPhase.start(player);
```
- ```
}
```
- ```
public class ParkourPhaseFactory implements PhaseFactory
```
- ```
@Override
```
- ```
public Level createPhase(String lvlData, String enemies, Tile[] tiles)
```
- ```
{
```
- ```
    ParkourPhase phase = new ParkourPhase(lvlData, tiles);
```
- ```
 phase.setStartPoint(startX, startY);
```
- ```
    return phase;
```
- ```
}
```

-

- src/gui/EnemyView (Downcast)
  - ```
public void render(Graphics g, Level level) {
    if (level instanceof BattlePhase) {
        BattlePhase battlePhase = (BattlePhase) level;
        for (Enemy enemy : battlePhase.getEnemies()) {
```
- src/util/LoadData (Upcast)
 - ```
ArrayList<Enemy> monsters = new ArrayList<>();
```
  - ```
monsters.add(new DudeMonster(col * TILE_SIZE, row * TILE_SIZE));
```
- **Abstrakcia:**
 - src/item/weapon
 - ```
public abstract class Weapon extends CollectibleItem implements Upgradable {
 public abstract void performAttack(Character attacker, Character target);
}
```
    - ```
public class Sword extends Weapon {
    @Override
    public void performAttack(Character attacker, Character target) {
        target.takeDamage(damage + attacker.getDamage());
        ...
    }
}
```

- **Musi obsahovat dostatok komentorov a anotacii**

(JavaDoc) : súbor s JavaDoc je v korene projektu

- **Musi byt jednotkovo otestovany (line coverage > 80% (bezne sa toto blizi k 100%))**

Element	Class, %	Method, %	Line, % ^	Branch, %
✓ all	88% (37/42)	87% (221/254)	86% (661/767)	64% (222/346)
Main	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
> item	100% (12/12)	71% (38/53)	73% (100/136)	46% (31/67)
> util	60% (3/5)	100% (6/6)	83% (49/59)	82% (32/39)
> map	81% (9/11)	88% (61/69)	87% (168/193)	67% (54/80)
> core	100% (4/4)	89% (25/28)	88% (94/106)	56% (25/44)
> entity	100% (9/9)	93% (91/97)	91% (250/272)	68% (80/116)
> gui	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
> input	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)

- **Pouzitie navrhovych/architektonickych vzorov**

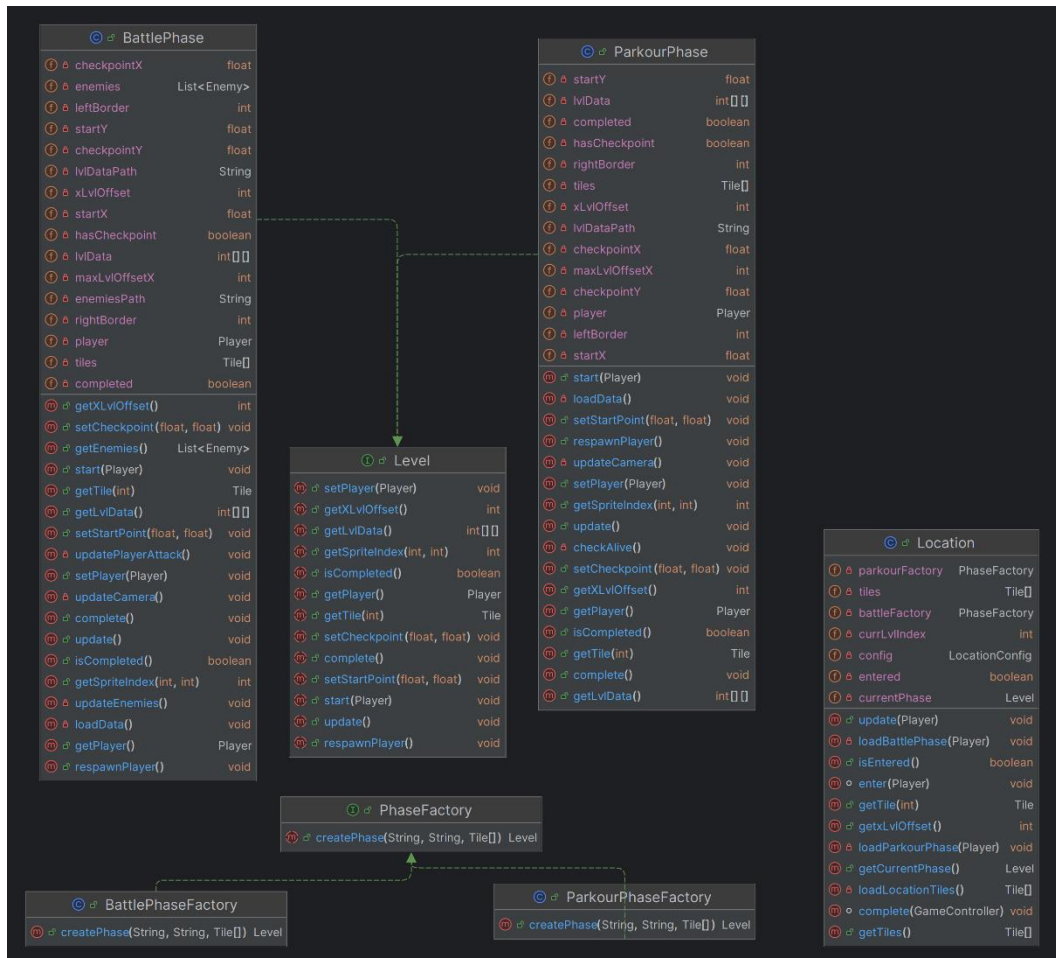
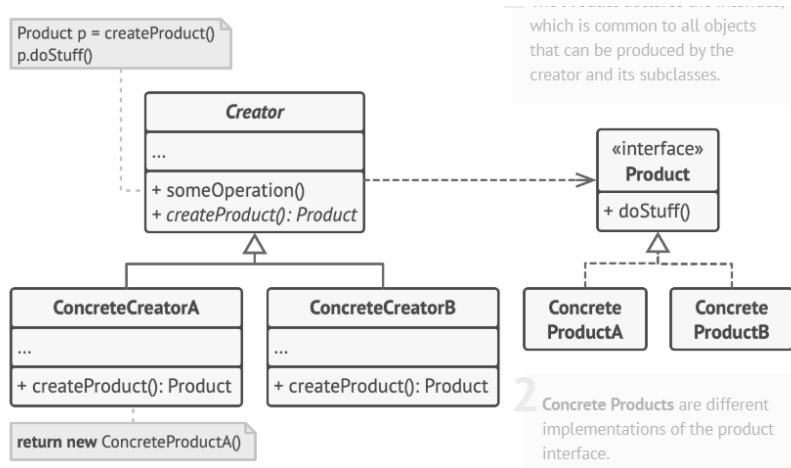
- Architektonicky vzor MVC

© GameController

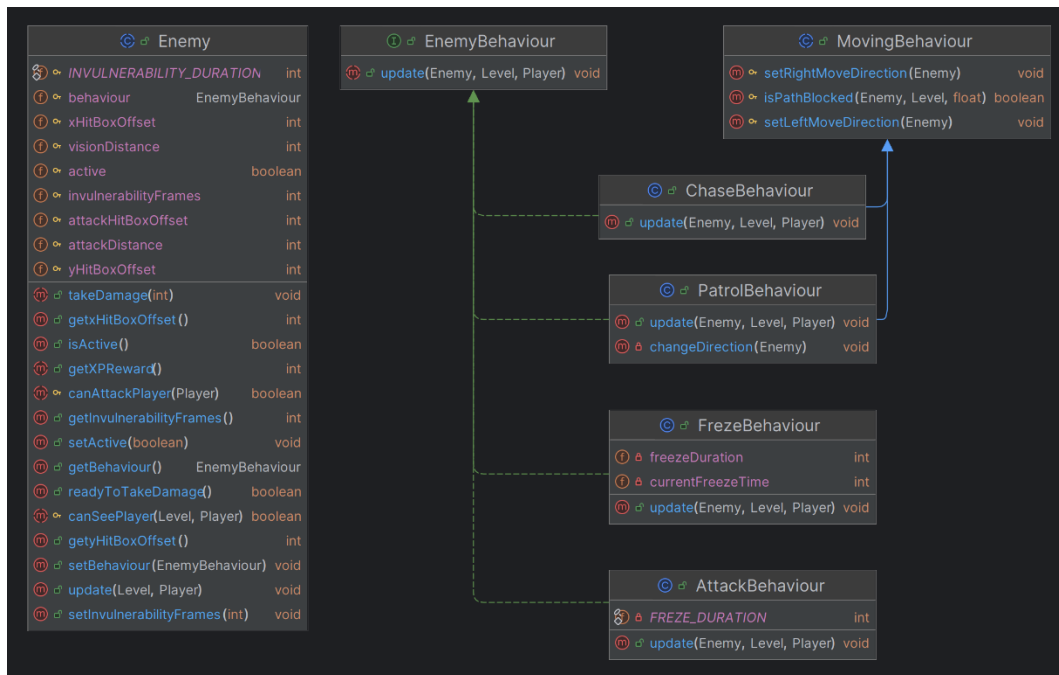
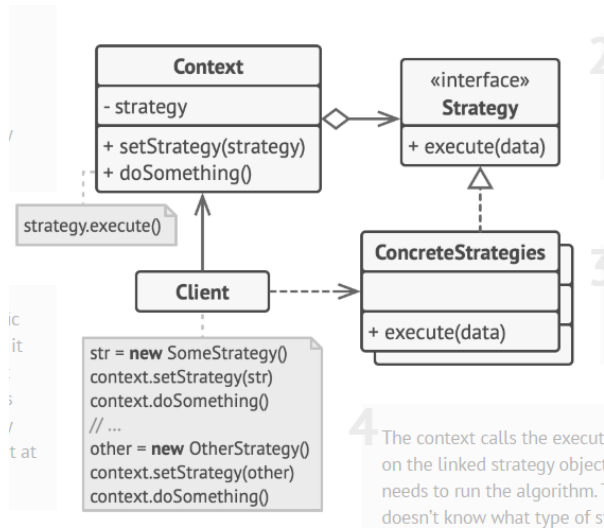
© GameView

© GameController

◦ Factory Method



- Strategy



Enemy is Context

EnemyBehaviour je Strategy

ChaseBehaviour, PatrolBehaviour, FreezeBehaviour, AttackBehaviour su concrete strategies

• Implementacia a vytvorenie GUI



• Použitie generickovosti vo vlastných triedach

- src/item/inventory

```
• public interface InventorySlot <T extends Item>{
    boolean canAccept(Item item);
    boolean isEmpty();
    T getItem();
    void setItem(T item);
    Item removeItem();
}

• public class SingleItemSlot<T extends Item> implements InventorySlot<T>
{
    private T item;
    private final Class<T> acceptedType;
    ...
}

• public class Inventory {
    private final Map<Integer, InventorySlot<? extends Item>> slots =
    new HashMap<>();
    public Inventory() {
        slots.put(1, new SingleItemSlot<>(Weapon.class));
        slots.put(2, new SingleItemSlot<>(Helmet.class));
        slots.put(3, new SingleItemSlot<>(Chestplate.class));
        slots.put(4, new SingleItemSlot<>(Leggings.class));
        slots.put(5, new SingleItemSlot<>(Boots.class));
        slots.put(6, new SingleItemSlot<>(Shield.class));
    }
    ...
}
```

• Použitie lambda vyrazov, referencii na metody

- src/map/level/BattlePhase

```
• if(enemies.stream().allMatch(enemy -> !enemy.isActive())) {
```

- src/item/ItemHandler

```
• items.removeIf(Item::isPickedUp);
```

- src/giu/EnemyView

```
• EnemyAnimationState state = animationStates.computeIfAbsent(enemy,  
    k -> new EnemyAnimationState(enemy));  
• animationStates.keySet().removeIf(enemy -> !enemy.isActive());
```

Navod ako spustit projekt

Download projekt z githubu, otvorit, prejsť na main, spustiť main

Používateľská príručka

Spustíte hru cez hlavnú obrazovku, môžete bežať doľava doprava pomocou A a D, môžete skákať pomocou SPACE.



Ak stojíte vedľa predmetu mozete ho prevziať pomocou E



V inventari sa mozete pohybovať pomocou tlacidiel 1-7, a vyhadzať predmety pomocu Q ak ste nastaveny na poziciu s predmetom.



Prechod na nový lvl



Mozete atakovat neprijatela pomocou RIGHT CLICK



Checkpoint, staci iba prejsť cez neho



Voda zabija