

Semestrální práce TS1

Projekt semestrální práce y PJV, herní engine Dungeon Escape.

Autor: Anna Sivongsak

Cvičící: Maximilián Herczeg

Contents

Návrh testovací strategie	2
Popis aplikace.....	2
Testovací strategie	2
1. Rozdělení aplikace na testovatelné části	2
2. Prioritizace testování	2
3. Test levels.....	3
Testovací scénáře.....	3
Testy vstupů	3
Testy průchodů	4
Detailní testovací scénáře	5
Implementace testů	6
Unit testy.....	6
Unit testy.....	6
Mockito	6
Integrační testy	6

Návrh testovací strategie

Popis aplikace

Aplikace je **herní engine Dungeon Escape**, kde hráč ovládá postavu a prozkoumává herní svět.

Hlavní funkcionality zahrnují:

- Pohyb a interakce** – hráč se pohybuje po mapě, sbírá předměty (klíče, lektvary, zbraně), bojuje s nepřáteli (monstra, slizové).
- Inventář** – ukládání a používání sebraných předmětů.
- Ukládání a načítání hry** – možnost uložit aktuální stav hry (pozice hráče, zdraví, sebrané předměty) a později pokračovat.
- Nepřátelé a překážky** – dveře (potřebují klíč), rostliny, které mohou bránit postupu.
- Grafické rozhraní** – hlavní menu, herní obrazovka s vykreslováním postav a prostředí.

Testovací strategie

1. Rozdělení aplikace na testovatelné části

Aplikaci rozdělíme podle MVC (Model-View-Controller):

Vrstva Komponenty

- Model:** Player, Game, Enemy, Monster, Slime, Key, Potion, Door, GameState, SaveLoad, LevelLoader
- View:** GamePanel, MainMenu (vykreslování, UI)
- Controller:** GameController, InputHandler (zpracování vstupů, logika hry)

2. Prioritizace testování

Priorita	Komponenta	Důvod
Vysoká	Player, Game, Enemy	Základní herní mechaniky (pohyb, boj, kolize).
Vysoká	SaveLoad, LevelLoader	Ovlivňuje průběh hry a stabilitu.
Střední	Inventory, Crafting	Sběr předmětů a craftování lektvarů.
Nízká	UI (GamePanel, MainMenu)	Vizuální reprezentace (testuje se manuálně).

3. Test levels

Úroveň testování	Co testujeme	Důvod	Příklady
Unit Testy	Jednotlivé třídy např. Player, Enemy, SaveLoad v izolaci.	Odhalí chyby v základní logice	Testování pohybu hráče např. player.move()
Integrační Testy	Interakce mezi třídami např. Player + Inventory	Odhalí chyby ve spolupráci komponent	Test craftování lektvaru (Inventory + Plant)
UI Testy	Reakce na vstup např. Klávesnice nebo Vykreslování (GamePanel, MainMenu)	UI chyby jsou pro hráče nejviditelnější	Ověření, že tlačítko "Start" v MainMenu spustí hru.
Systémové Testy (End-to-End)	Celková hratelnost – od hlavního menu po konec hry. Kritické scénáře - řechody mezi úrovněmi (LevelLoader)	Odhalí globální problémy , které unit/integrační testy nechytí	Hráč dokončí úroveň -> načte se další úroveň.

Testovací scénáře

Testy vstupů

Pohyb hráče player.move()

Kombinace pohybových příkazů (left, right, jump)

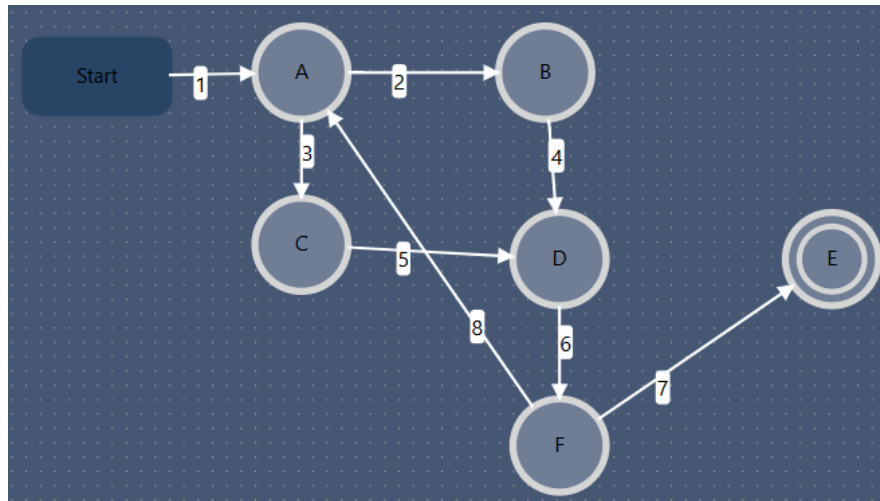
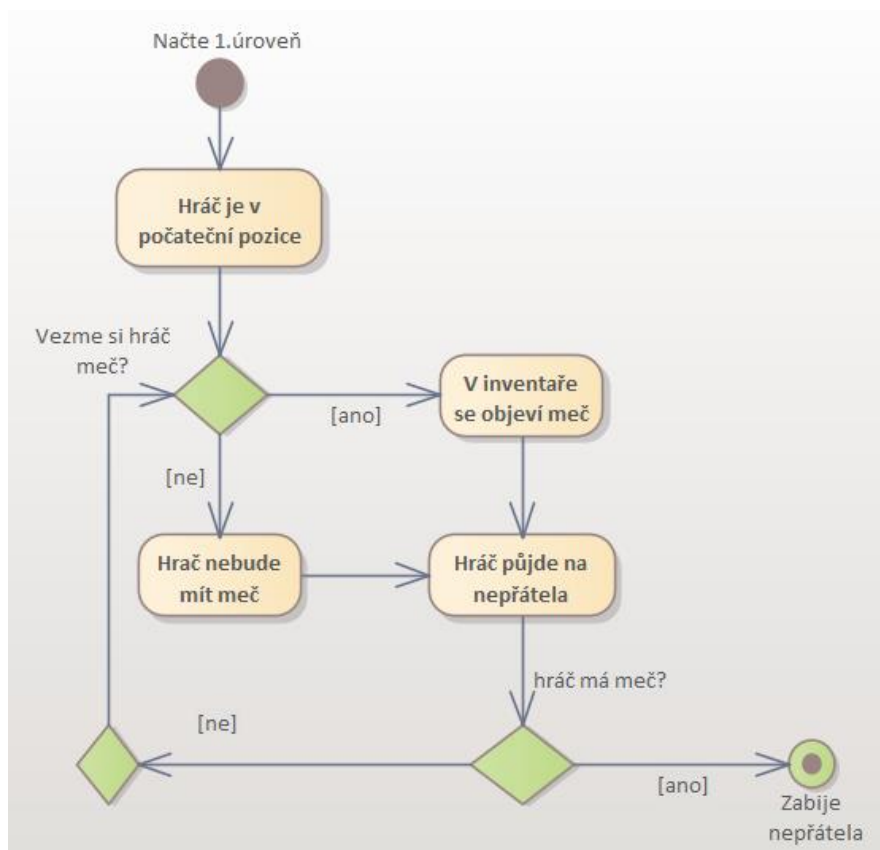
Analýza ekvivalentních tříd (EC) a mezních podmínek:

Parametr	EC	Mezní podmínky
Left	True False	Stisk klávesy A (pohyb doleva)
Right	True False	Stisk klávesy D (pohyb doprava)
Jump	True False	Stisk mezerníku pouze při isOnGround=true (Skok)

Pairwise testovací kombinace:

Test Case	Left	Right	Jump	Očekávaný výsledek
1	True	False	False	Pohyb vlevo
2	False	True	True	Pohyb vpravo + skok
3	True	True	False	Žádný pohyb
4	False	False	True	Pouze skok
5	True	False	True	Pohyb vlevo + skok

Testy průchodů



Algorithm: PCT

Test depth level: 2

Reduction: TRUE

Sequences:

Show nodes ☐

1.

1 - 2 - 4 - 6 - 7

☐

2.

1 - 3 - 5 - 6 - 8 - 2 - 4 - 6 - 7

☐

3.

1 - 2 - 4 - 6 - 8 - 3 - 5 - 6 - 7

☐

Detailní testovací scénáře

Cíl scénáře:

Ověřit, že hráč může správně vyrobit lektvar, když splní všechny podmínky:

1. Má v inventáři alespoň 2 rostliny.
2. Nachází se u crafting stanice (velká rostlina PLANT_BIG).

Kroky scénáře:

1. Příprava prostředí
 - Hra načte úroveň, kde se nachází:
 - 2 malé rostliny (Plant) k sebrání
 - 1 crafting stanice (PLANT_BIG)
2. Sbírání rostlin
 - Hráč se přesune k první rostlině a sebere ji.
 - *Ověření:* Inventář hráče nyní obsahuje 1 rostlinu.
 - Hráč se přesune k druhé rostlině a sebere ji.
 - *Ověření:* Inventář hráče nyní obsahuje 2 rostliny.
3. Přiblížení ke crafting stanici
 - Hráč dojde k crafting stanici (PLANT_BIG).
 - *Ověření:* Hra detekuje, že hráč je u crafting stanice (`isNearCraftingPlant() == true`).
4. Výroba lektvaru
 - Hráč aktivuje craftování stiskem tlačítka "E".
 - *Ověření:*
 - Z inventáře zmizí 2 rostliny.
 - Přibude 1 lektvar (Potion).
 - V logu se objeví zpráva "*Potion crafted!*".
5. Alternativní scénáře
 - Nedostatek rostlin:
 - Hráč má v inventáři 0 nebo 1 rostlinu.
 - *Ověření:* Craftování selže, inventář se nezmění.
 - Vzdálenost od stanice:
 - Hráč není u crafting stanice.

- *Ověření:* Craftování nelze aktivovat.

Kritické body:

- Správná detekce pozice hráče vůči crafting stanici.
- Kontrola počtu rostlin před craftováním.
- Synchronizace změn v inventáři (odebrání rostlin + přidání lektvaru).

Očekávaný výsledek:

Po splnění všech podmínek se v inventáři objeví lektvar a systém zaznamená úspěšný craft.

Implementace testů

Unit testy

Unit testy

Player

- `update_AppliesGravity_WhenNotOnGround()`
- `move_Right_IncreasesX()`
- `takeDamage_DecreasesHealth()`
- `heal_IncreasesHealth()`
- `monsterPatrolsRightInitially()`
- `monsterSwitchesDirectionAtPatrolMax()`
- `monsterAttacksPlayerWhenClose()`
- `monsterStopsAttackingAfterDelay()`

SaveLoad

Mockito

- `saveGameCreatesFile()`
- `testHasSavedGameReturnsFalseIfMissing()`
- `loadGameRestoresState()`

Game, Player, Inventory, Enemy

Integrační testy

- `toggleInventoryAffectsItemsToDraw()`
- `resetRestoresGameState()`
- `craftPotionSuccess()`
- `playerAffectedByGravity()`
- `playerJumpOnlyFromGround()`
- `monsterPatrols()`
- `enemyAttacksPlayer()`
- `playerAttacksEnemy()`

Celkem: 8 unit testů + 3 mockito + 8 integračních testů