Dmytro Shapovalov , Oliver Udvardi, -2.01/b(CPUb), Utorok 14:00

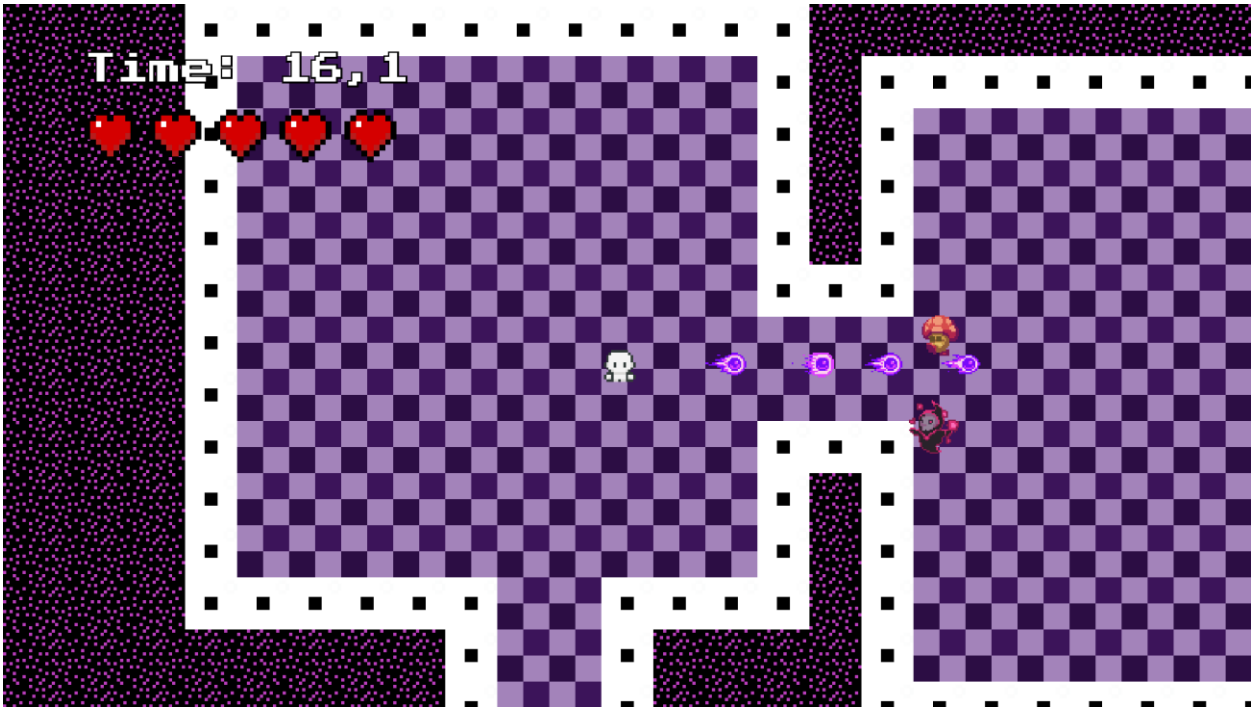
Projekt z objektovo orientovaného programovania LS2025

Zámer projektu

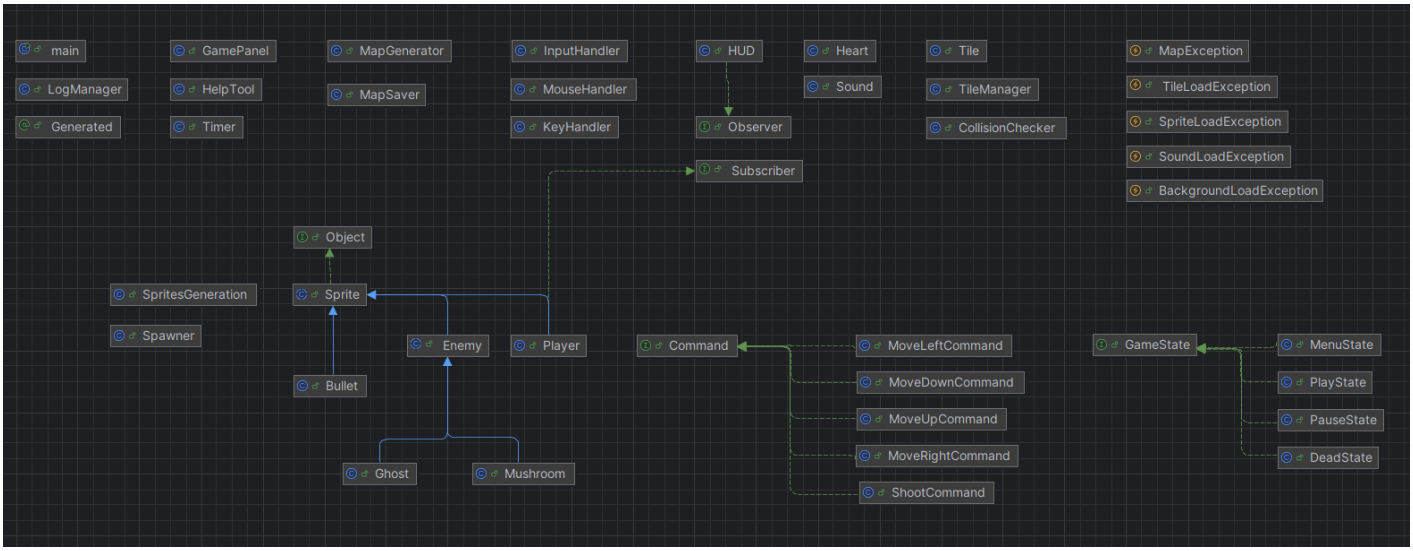
Overturn is a fast-paced 2D action game with roguelike elements, where the goal is to survive as long as possible. Thanks to the random generation of rooms and enemies, each run is unique, turning every escape attempt into a new challenge. The player explores endlessly changing corridors, battles opponents. The controls are intuitive — the character can move in four directions and shoot at enemies. But it won't be easy. The difficulty constantly increases: the number of enemies grows over time. But the player's health increases with the number of enemies he kills. Yet, every run is unpredictable, and there is no universal path to victory.
There 2 types of enemies: mushroom and ghost. Mushrooms walk slower but do more damage, and ghosts are fast but do less damage.
The sound design and visual style create an unsettling dreamlike atmosphere, where familiar places turn into threats, and the exit seems unreachable.
Overturn is a race against time — a game with no final destination, only the struggle for survival and the exploration of the unknown.

Žáner: *roguelike*.


OVERTURN
>PLAY
QUIT


Time: 16,1

UML diagram tried



Použité knižnice

Swing, Awt: for drawing picture.

Junit5, mockito: for unit testing.

Vyjadrenie sa k splneniu podmienok (nutnych, aj dalsich)

Ku kazdej podmienke napiste aspon jeden priklad kde v projekte je splnena, dodrzujte prosim poradie ako je na uvedene na stranke projekt:

- Obsahovat dedenie a rozhrania:
  - class Player extends Sprite, interface Object
- Pouzitie zakladnych OOP principov (enkapsulacia, dedenie, polymorfizmus, abstrakcia):
  - In class Player: methods getKills(), setKills(int kills).
  - `class Ghost extends Enemy`
  - `interface Object`
  - `abstract class Sprite`
- Musi obsahovat dostatok komentorov a anotacii (JavaDoc):
  - JavaDoc in the repository
- Musi byt jednotkovo otestovany (line coverage > 80%):
  - All tests in test folder
- Pouzitie navrhovych/architektonickych vzorov:
  - Design pattern Observer in System.Observer. Used for notifying HUD when player's health changes.

    - Design pattern States in System.States. Used to separate and conveniently switch between different stages of the game (Menu, Play, Pause, Death).
    - Design pattern Commands in Sprites.Player.Commands to organize the control of the player's actions(Move up, down, left, right and shoot).
- Logovanie zakladnych cinnosti:
    - Logging all basic activities. For example in main: *LOGGER*.info("Background image loaded");
- Implementacia a pouzitie vlastnych vynimiek:
    - In System.Exceptions. For example: public class BackgroundLoadException extends RuntimeException
- Implementacia a vytvorenie GUI:
    - Drawing tiles, player, enemies, HUD and others. For example in Player class: public void draw(Graphics2D g2)
- Explicitne pouzitie viacvlaknovosti:
    - In HUD.Sound: public synchronized void playMusic()
- Pouzitie generickovsti vo vlasnych triedach:
    - In Sprites.Generation in class Spawner: public <T> T spawn(Class<T> clazz, Object... args)
- Pouzitie serializacie/IO:
    - In Map.Genereation in class MapSaver: public static void saveMap(int[][] map, String filepath)

Navod ako spustit project

Run class main.

Pouzivatelska prirucka

In the main menu, as well as in the pause menu and the death menu, switch buttons using the keys: w (up), s (down). To activate the selected button: enter. During the game, the character moves using the keys: w (up), s (down), a (left), d (right), and shooting with the left mouse button. You can also enter the pause menu (pause the game) using the esc key and return to the game also using the esc key.