SPŠE Ječná

Informační technologie - Programování a digitální technologie

Ječná 517, 120 00 Nové Město

BLOOD KEEP

Filip Hofman, C2c
Informační a komunikační technologie
2025

Content:

BloodKeep

1. The goal of the work

The goal of this project was to develop a simple text-based adventure game inspired by dark fantasy themes. The player explores a cursed facility known as BloodKeep, which was once a place of scientific research but has now fallen under the influence of demonic forces. The main objective is to survive, gather three key fragments, unlock the final gate, and defeat the final boss to bring peace to the world.

2. Description

BloodKeep is a text-based adventure game programmed in Java. It uses a command-based structure that allows the player to navigate through rooms, interact with the environment, battle enemies, and progress through a story-driven experience. The application utilizes object-oriented principles, external data sources for dynamic content loading, and console interaction as its interface.

2.1 Story

The ruins of BloodKeep once hosted a powerful research base studying interdimensional portals. However, during one experiment, a portal to a dark realm was opened, allowing demonic entities to flood the facility. The scientists were corrupted or destroyed, and the complex became a twisted, dangerous labyrinth.

2.2 Characters

The player can choose between three unique classes: Swordsman, Trickster, and Gunslinger. Each class starts with different amounts of health and weapon damage, affecting gameplay style and strategy. As the game progresses, the player collects items, improves stats, and uses strategic decisions to overcome challenges. There are four primary non-player characters (NPCs) the player can encounter. The Gatekeeper stands in front of the sealed Heart of BloodKeep and informs the player about the three shards required to unlock the gate. The Freed Prisoner, discovered in the Crimson Vault, expresses gratitude for being rescued. The Armorer resides in the Forgotten Armory and offers the player free upgrades for their weapon. In the Thorned Garden, the Druidess provides healing items as a gesture of aid. Demons serve as the primary enemies and appear frequently in hallways. They must be defeated to progress and reward the player with orbs, which are used as in-game currency. A miniboss guards the Graven Crypt and drops a vital key shard upon defeat. The final boss, Velkhar the Crimson Tyrant, awaits within the Heart of BloodKeep as the ultimate challenge.

2.3 Game mechanics

The game contains a variety of commands that allow the player to interact with the world. Using the go command, the player can move between rooms in four basic directions. However, some rooms are locked and the appropriate keys must be found or purchased. The examine command is used to explore a room - a list of items located in the room is printed out, and the player has the option to take the selected items in a backpack. If there is an NPC character in the room, the player can start a conversation with them using the talk command, which will trigger their predefined dialogue. The fight command allows the player to enter into a fight with the demon if it is in the room. The player first chooses whether to fight or flee. If he chooses to fight, a turn-based duel takes place in which the player and monster alternate attacks. If the player dies, the game ends. If, on the other hand, he wins, he gets a certain amount of orbs as a reward, which he can use in the shop. The shop is activated with the shop command, but only works in rooms with a shopkeeper. After activating it, the player chooses whether to buy or sell. If buying, a menu of items available to the shopkeeper is displayed. For selling, items from the player's backpack will be displayed and the player chooses which one he wants to offer. The player can open his backpack at any time using the backpack command. He can choose to display all his items or use some of them - for example, to heal himself with healing items. If he wants to see what commands are available, he can use the commands command, which will list them from a file. There is also a playerstats command that will list the player's current stats - number of lives, weapon strength, amount of orbs collected, weapon name, and player name. The last command is end, which is used to end the game immediately.

3. System requirements

The game requires Java 17 or later and must be run in a console environment or an IDE that supports standard input and output. It is compatible with Windows, Linux, and macOS systems. The game also depends on access to a local file system to load data such as the map, items, NPCs, and shop inventories.

4. Basic structure

The *BloodKeep* game is implemented using a clean object-oriented structure and follows well-established design principles for modularity and extensibility. The code is organized into several logically divided packages, each responsible for specific aspects of the game's functionality. This structure ensures maintainability and readability throughout the project.

At the heart of the application is the **World** package, which contains the `WorldMap`, `Location`, and `Item` classes. The `WorldMap` manages all game rooms, their connections, items placed in those rooms, and entities like demons and NPCs. The `Location` class represents each individual room and its adjacent directions. The `Item` class models interactive objects such as keys, healing items, lore scrolls, and weapon upgrades.

The **characters** package includes all game characters such as `Player`, `Demon`, `NPC`, and the player's weapon. The `Player` class holds attributes like health, name, class, and equipped weapon. Demons are hostile entities tied to specific locations, while NPCs provide narrative dialogues and additional interaction.

The **command** package implements the **Command design pattern**. Each user action is encapsulated in its own class such as `Go`, `Fight`, `Talk`, `Shop`, `Examine`, `Backpack`, `PlayerInfo`, and more. These commands are initialized in the `Console` class, which also handles user input and coordinates command execution.

The `Console` class functions as the game's controller and main loop, processing player input and displaying the game's narrative feedback. It initializes the world, loads data from files, and ties together all components of the system.

Additional functionality is provided through file-based initialization, where map structure, items, NPCs, and demons are loaded from external text files. This ensures flexibility and ease of content updates without modifying the source code.

Overall, the project is structured in a clear and extensible way, which not only facilitates future modifications and testing but also keeps the game logic easy to follow.

5. Testing data

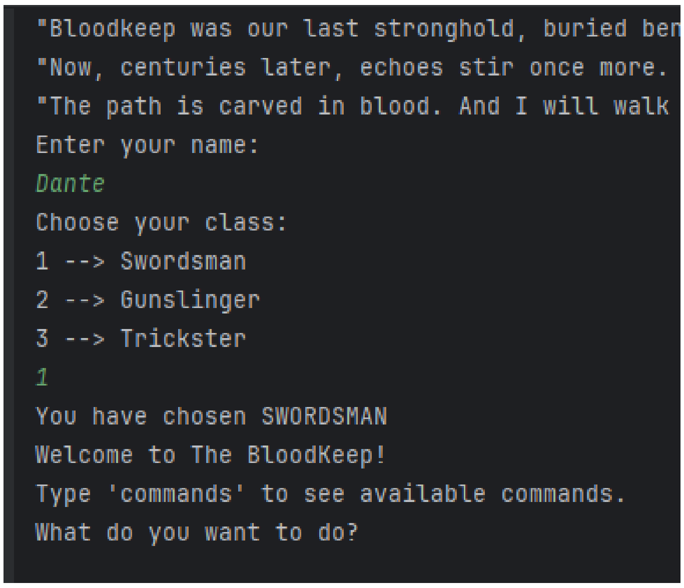
The game was tested using unit tests focused on verifying the core functionalities of several key classes. Tests were written for `Backpack`, `Examine`, `Fight`, `PlayerInfo`, `End`, and `Weapon`.

In the `Backpack` class, tests confirmed that items are correctly added and removed, and also verified whether the player possesses all three BloodKeep shards required to open the final door. The `Examine` class was tested primarily for correct behavior with invalid or valid room locations. The `Fight` class was tested to ensure that the command correctly identifies when there is no demon present in a location.

Other tests such as those for `PlayerInfo` or `End` were simpler and primarily verified that the methods return the correct strings or that the game ends as expected. All tests helped ensure the reliability and stability of the application's key features.

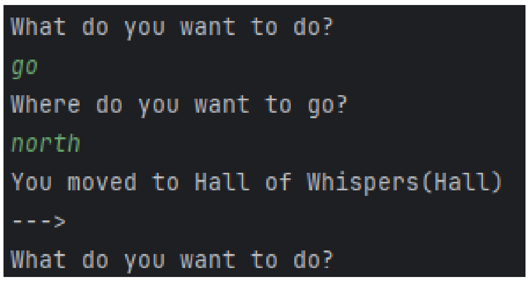
6. User manual

The game is operated entirely through a text-based interface. Upon launching the program, the player is first introduced to the story, setting the atmosphere and context for the adventure. After this introduction, the player is prompted to enter their character's name and choose a class from three available options: **Swordsman, Gunslinger, or Trickster**. Each class provides different stats, influencing the player's strengths and playstyle. Once the character is created, the game welcomes the player and begins with the character positioned in the starting location called Withered Gate, which is assigned the ID 0.



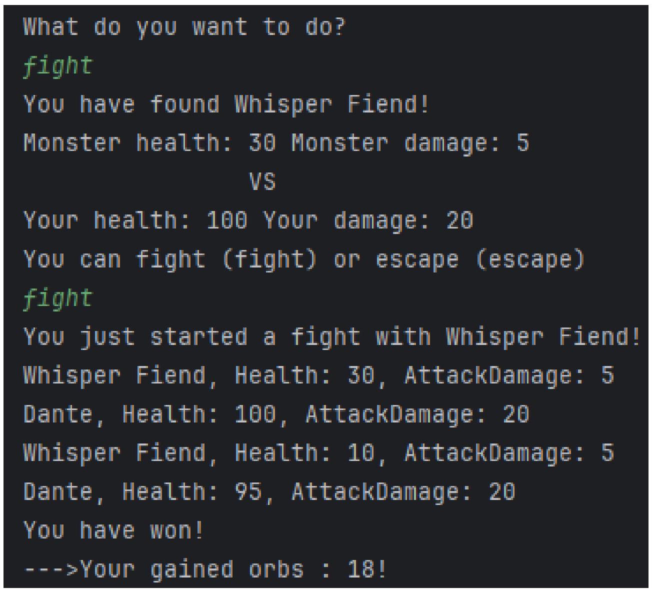
```
"Bloodkeep was our last stronghold, buried ben
"Now, centuries later, echoes stir once more.
"The path is carved in blood. And I will walk
Enter your name:
Dante
Choose your class:
1 --> Swordsman
2 --> Gunslinger
3 --> Trickster
1
You have chosen SWORDSMAN
Welcome to The BloodKeep!
Type 'commands' to see available commands.
What do you want to do?
```

Players interact with the game world by typing commands into the console. To discover all the available commands, the player can enter the **commands** command, which displays a list of supported commands along with brief explanations. Movement throughout the game world is handled using the **go** command. After typing **go**, the player must specify the direction they wish to travel, such as **north, east, south, or west**. If there is a valid location in the chosen direction, the player's character will move there. Otherwise, the game will notify the player that movement in that direction is not possible.



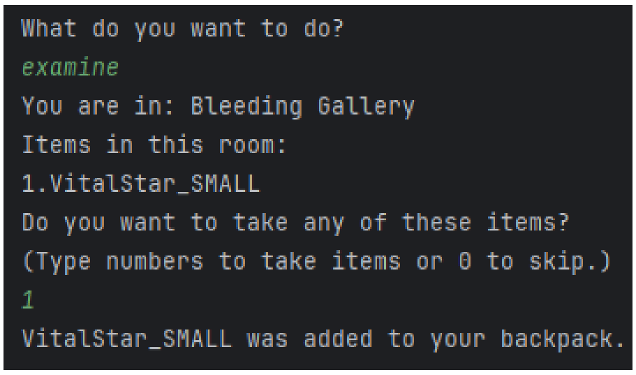
```
What do you want to do?
go
Where do you want to go?
north
You moved to Hall of Whispers(Hall)
--->
What do you want to do?
```

When a demon is present in the player's current location, the **fight** command becomes available. Using this command shows the demon's stats alongside the player's own stats, allowing the player to assess whether to engage in battle. The player is then prompted to choose between **fight** or **escape**. Selecting **fight** initiates combat between the player and the demon, with victory rewarding the player with orbs. If the player loses, the game ends. Choosing **escape** cancels the combat and returns the player to the exploration state.



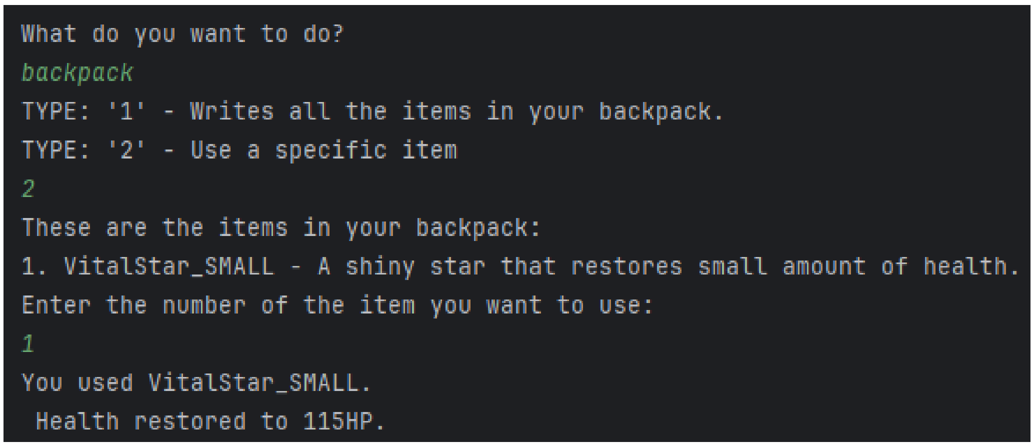
```
What do you want to do?
fight
You have found Whisper Fiend!
Monster health: 30 Monster damage: 5
                VS
Your health: 100 Your damage: 20
You can fight (fight) or escape (escape)
fight
You just started a fight with Whisper Fiend!
Whisper Fiend, Health: 30, AttackDamage: 5
Dante, Health: 100, AttackDamage: 20
Whisper Fiend, Health: 10, AttackDamage: 5
Dante, Health: 95, AttackDamage: 20
You have won!
--->Your gained orbs : 18!
```

The **examine** command provides detailed information about the player's current surroundings. It reveals the name of the location, any items available there, and whether any demons or NPCs are present. The player can also pick up items from the location and add them to their backpack during examination.



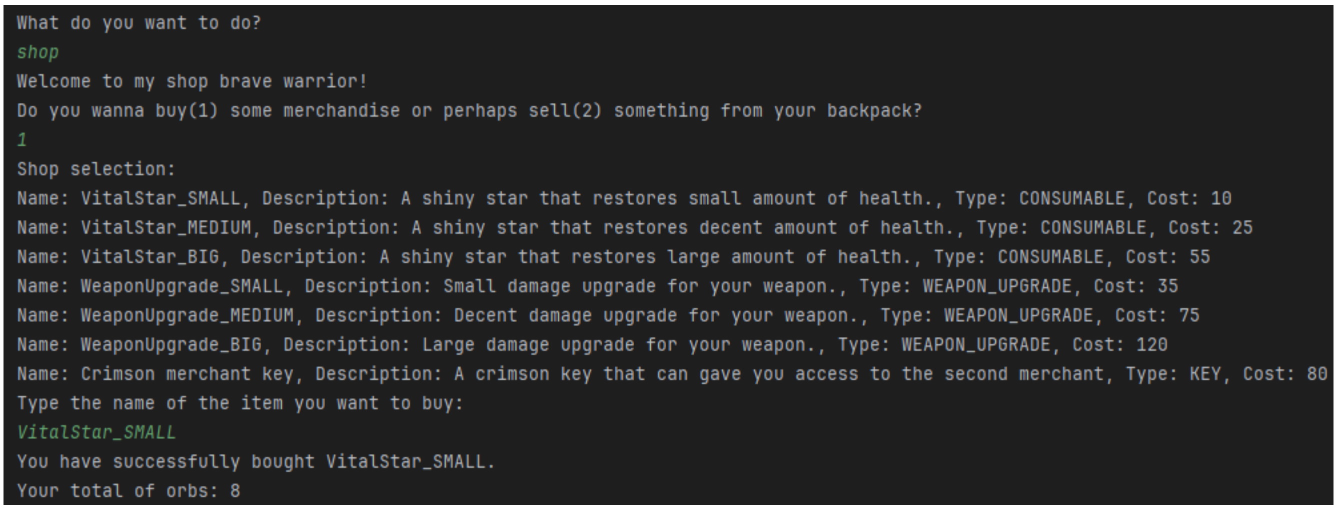
```
What do you want to do?
examine
You are in: Bleeding Gallery
Items in this room:
1.VitalStar_SMALL
Do you want to take any of these items?
(Type numbers to take items or 0 to skip.)
1
VitalStar_SMALL was added to your backpack.
```

The player's backpack can be managed using the **backpack** command. This opens a menu where the player can view all items currently held or use selected items.



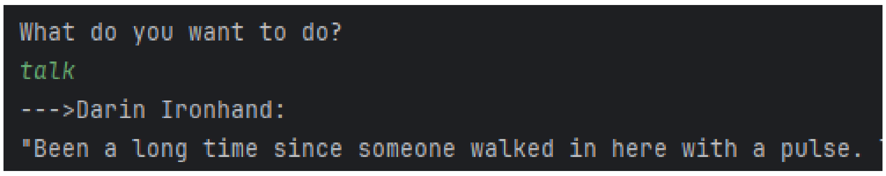
```
What do you want to do?
backpack
TYPE: '1' - Writes all the items in your backpack.
TYPE: '2' - Use a specific item
2
These are the items in your backpack:
1. VitalStar_SMALL - A shiny star that restores small amount of health.
Enter the number of the item you want to use:
1
You used VitalStar_SMALL.
 Health restored to 115HP.
```

In locations where a merchant is present, the **shop** command allows the player to interact with the merchant's inventory. The player can choose to buy or sell items. When buying, the game lists all available items with descriptions and prices in orbs, and the player can purchase items by entering their names. When selling, the player selects items from their backpack to sell in exchange for orbs.



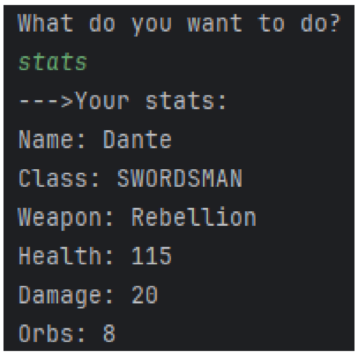
```
What do you want to do?
shop
Welcome to my shop brave warrior!
Do you wanna buy(1) some merchandise or perhaps sell(2) something from your backpack?
1
Shop selection:
Name: VitalStar_SMALL, Description: A shiny star that restores small amount of health., Type: CONSUMABLE, Cost: 10
Name: VitalStar_MEDIUM, Description: A shiny star that restores decent amount of health., Type: CONSUMABLE, Cost: 25
Name: VitalStar_BIG, Description: A shiny star that restores large amount of health., Type: CONSUMABLE, Cost: 55
Name: WeaponUpgrade_SMALL, Description: Small damage upgrade for your weapon., Type: WEAPON_UPGRADE, Cost: 35
Name: WeaponUpgrade_MEDIUM, Description: Decent damage upgrade for your weapon., Type: WEAPON_UPGRADE, Cost: 75
Name: WeaponUpgrade_BIG, Description: Large damage upgrade for your weapon., Type: WEAPON_UPGRADE, Cost: 120
Name: Crimson merchant key, Description: A crimson key that can gave you access to the second merchant, Type: KEY, Cost: 80
Type the name of the item you want to buy:
VitalStar_SMALL
You have successfully bought VitalStar_SMALL.
Your total of orbs: 8
```

If the player is in the presence of a non-player character (NPC), the **talk** command lets the player engage in dialogue, displaying the NPC's messages.



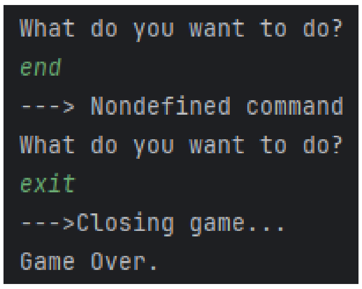
```
What do you want to do?
talk
--->Darin Ironhand:
"Been a long time since someone walked in here with a pulse.
```

At any time, the player can view their current character stats using the **player info** command by typing "stats". This includes information such as the character's name, class, weapon damage, orb count, and health status.



```
What do you want to do?
stats
--->Your stats:
Name: Dante
Class: SWORDSMAN
Weapon: Rebellion
Health: 115
Damage: 20
Orbs: 8
```

Finally, the player can quit the game immediately by entering the **end** command, which terminates the game session.



```
What do you want to do?
end
---> Nondefined command
What do you want to do?
exit
--->Closing game...
Game Over.
```

The game provides clear and immediate feedback after every command, ensuring the player remains informed about the outcomes of their actions. All commands are executed sequentially, and the game continues running until the player chooses to exit.

7. Conclusion

Working on this project has been a valuable and rewarding experience, especially in deepening my understanding of object-oriented programming in Java. I approached the development process with a structured plan, aiming to work on the game several times each week. Early in the process, I divided the project into logical parts and tackled them step by step—starting with building the world map and implementing movement, followed by adding features such as the backpack system and the examine command.

Throughout development, I encountered several challenges, particularly when it came to loading data from files—for example, items or demons. Although this slowed down my progress initially, once I managed to implement the first loading method successfully, others followed a similar structure and became much easier to handle.

One aspect of the project I'm especially proud of is the use of the Command Pattern, which made command handling clean and modular. Additionally, all dynamic elements in the game—such as NPCs and demons—are loaded from external files, which contributes to flexibility and scalability. I also believe the use of packages greatly improves the organization of the project, making it easier to navigate and maintain.

On the other hand, there are areas that could be improved. For instance, I could have added more commands to enhance gameplay variety, implemented background music for atmosphere, and invested more time into creating robust unit tests. These are features I would consider adding or refining in the future.

Overall, this project significantly enhanced my skills in Java programming and design. It gave me practical experience in structuring codebases, working with file I/O, and building interactive systems—all of which I consider highly beneficial for my development as a programmer.

8. Sources

- **OpenAI ChatGPT**
- **Previous IT projects**
- https://www.geeksforgeeks.org/java/