**Supplementary Specification – First Draft**

**Common Functionality**:

Logging ?

Save & Load ?

File I/O ?

External Database?

Multi-threaded Programming?

Caching?

**Usability:**

User Interface

Main Menu Screen: Appears when the game executable is launched. It contains only the following three buttons: Start a new game, Help and Exit. If the user clicks the “Start a new game” the game proceeds with the building mode initialized with empty halls.

Game Over Screen: Appears whenthe game ends

Build Mode Window: Lets the user to configure the halls.

Play Mode Window: Lets the user to play the game.

Pause/Resume/Exit Buttons: Lets the user to pause the game during game play and resume later.

Visual indicators for time remaining, lives, and the current hall name.

Help Screen: Explains the game objects hand features, and how to play.

**Ease of Navigation:** The player should be able to move the hero intuitively using keyboard inputs.

A screenshot of a video game

Description automatically generated

Figure 1 Build Mode Window (temporary)

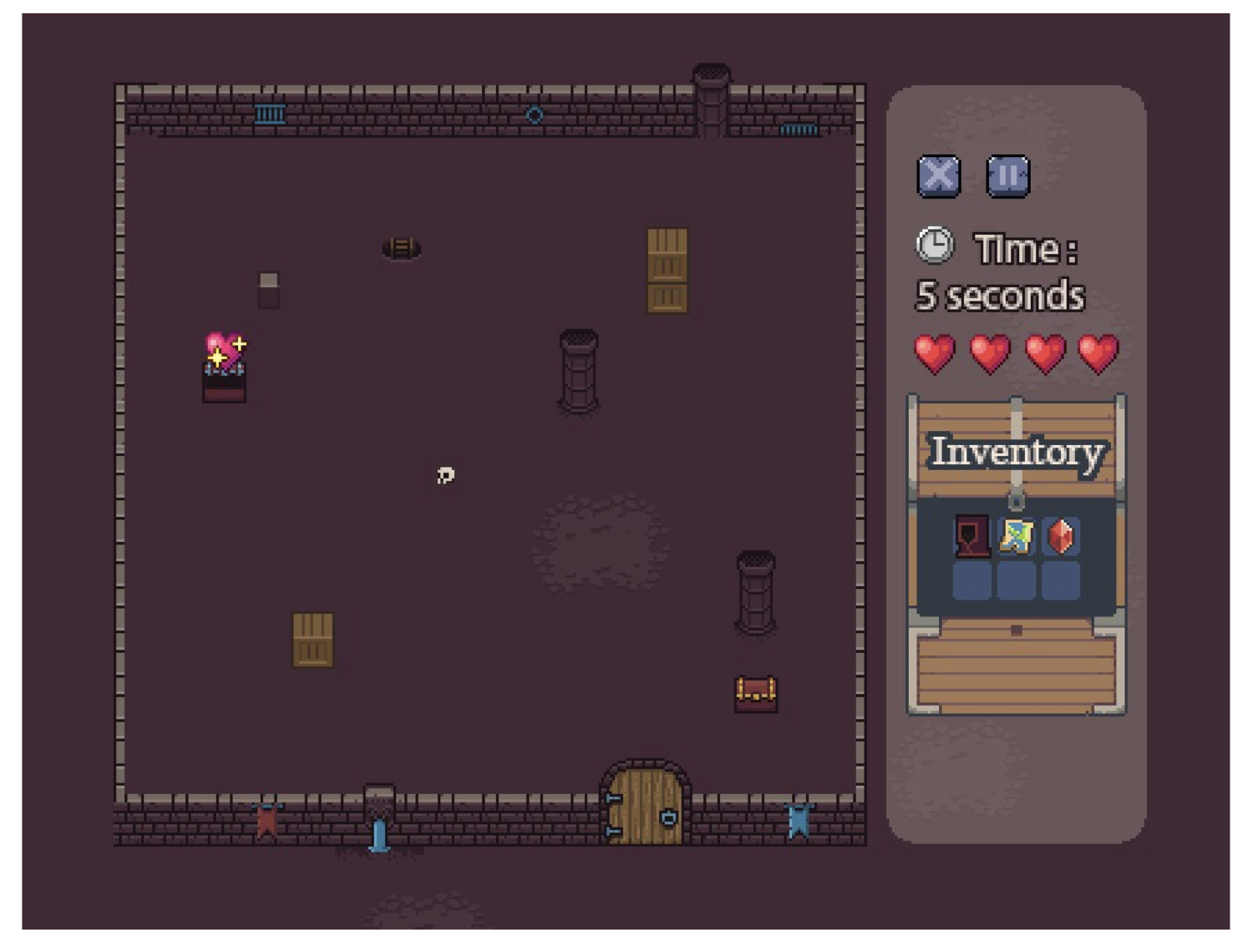


Figure 2 Play Mode Window (temporary)

Player and Monster Images:

A cartoon of a person

Description automatically generatedA pixelated cartoon character

Description automatically generated A pixelated cartoon of a pig

Description automatically generated A pixelated cartoon of a gnome

Description automatically generated

**Reliability:**

Handle invalid user actions gracefully (e.g., clicking on inaccessible objects).

Ensure no entity overlaps (e.g., walls, monsters, hero, objects).

Monsters and enchantments should spawn at predictable intervals (every 8 seconds and 12 seconds, respectively).

**Performance:** response times, throughput, accuracy, availability, resource  usage.

Frame Rate: 30 FPS or greater.

Handle sound effects without causing noticeable delays or performance drops.

Smooth rendering of animations, such as monster movements and rune discovery.

No excessive CPU or memory usage.

Proper file handling to avoid data corruption (e.g., save files).

Save & Load speed? Caching? In-memory storage vs disk storage vs external database?

**Supportability:**.

Modular codebase for easy updates and expansions (e.g., adding new halls or monster types).

Comments and documentation for all major code components to facilitate future maintenance.

Language support: Only English.

**Design/Implementation Constraints:**

For GUI, either Java Swing or JavaFX is permited. Java as the programming language.

IDE: IntelliJ ?

Sound effects should not exceed 5 seconds in length to ensure quick playback

**Free Open Source Components:**

Candidate tools we might use:

VisualParadigm, Lucidchart > for drawing UML diagrams

JUnit > for unit testing

SLF4J or log4j > if logging will exist

Github > for version management

Maybe some permitted graphics/gaming libraries ?

**Hardware/Software Constraints**

Player must have a keyboard and mouse to send inputs to system.

**Legal**

The graphical assets used must be credited properly if taken from the provided assets package.