



**HOCHSCHULE
HANNOVER**
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

Maze Adventure – Virenjagd

Use Case Spezifikationen

Projekt: darkIT Spielesammlung

Gruppe: 03

Entwickler: Artur Sanamyan (SanamTT)

Semester: 3. Semester BIN

Hochschule Hannover

Betreuer: Andreas Holitschke

Oktober 2025

Überblick

Dieses Dokument beschreibt die Akteure, die einzelnen Use Cases und das grundlegende Verhalten des Spiels **Maze Adventure – Virenjagd**. Es umfasst die drei Schwierigkeitsstufen (*EASY*, *MEDIUM*, *HARD*) sowie typische Systemfunktionen wie Bewegung, Sammeln von sicheren Dateien, Viruskollision, Levelauswahl, Punktevergabe, Pausieren und Neustart.

Akteure

- **Spieler (Hacker)** – Hauptakteur, steuert die Spielfigur, sammelt sichere Dateien, vermeidet Viren und sucht den Ausgang.
- **Spielsystem** – verwaltet die Labyrinthzeugung, Kollisionen, Punktestand und Benutzeroberfläche.
- **Benutzeroberfläche (Launcher/Menu)** – ermöglicht die Levelauswahl und den Spielstart.

Haupt-Use-Cases

1. Level auswählen (*EASY* / *MEDIUM* / *HARD*)
2. Spiel starten / Level laden
3. Spieler bewegen
4. Sichere Datei sammeln
5. Virus treffen (negativ)
6. Ausgang erreichen / Level abschließen
7. Spiel pausieren / fortsetzen
8. Level neu starten
9. Punktestand anzeigen / aktualisieren
10. Labyrinth automatisch erzeugen
11. Spielende / Zurücksetzen

Detaillierte Use Cases

UC-01: Level auswählen

Primärer Akteur: Spieler

Vorbedingungen: Launcher läuft, Hauptmenü sichtbar

Ablauf:

1. Spieler wählt Easy, Medium oder Hard.
2. UI bestätigt Auswahl und startet das Spiel mit dem gewählten Level.

Nachbedingungen: Das Spielsystem erstellt ein Level mit den entsprechenden Parametern (Zeilen \times Spalten).

UC-02: Spiel starten / Level laden

Primärer Akteur: Spieler

Vorbedingungen: Level wurde gewählt

Ablauf:

1. Das Spielsystem generiert das Labyrinth (Wände, Ausgang, Viren, Dateien, Startpunkt).
2. Das Spielfeld wird angezeigt und wartet auf Eingaben.

Nachbedingungen: Das Level ist aktiv und spielbar.

UC-03: Spieler bewegen

Primärer Akteur: Spieler

Vorbedingungen: Spiel läuft

Ablauf:

1. Spieler drückt eine Pfeiltaste.
2. System prüft, ob das Ziel frei ist.
3. Bewegung wird ausgeführt, falls erlaubt.
4. System prüft auf Kollisionen und aktualisiert die Anzeige.

Nachbedingungen: Spielerposition wird aktualisiert.

UC-04: Sichere Datei sammeln

Primärer Akteur: Spieler

Vorbedingungen: Spieler bewegt sich auf eine Zelle mit sicherer Datei.

Ablauf:

1. Kollision mit Datei erkannt.
2. Datei wird entfernt.
3. Punktestand wird erhöht.
4. Nachricht: „Sichere Datei gesammelt!“ wird angezeigt.

Nachbedingungen: Datei entfernt, Punktestand erhöht.

UC-05: Virus treffen (negativ)

Primärer Akteur: Spieler

Vorbedingungen: Spieler bewegt sich auf eine Zelle mit Virus.

Ablauf:

1. System erkennt Kollision mit Virus.
2. Nachricht: „Oh nein! Virus getroffen!“ wird angezeigt.
3. Spieler wird an Startposition zurückgesetzt.
4. Optional: Punktabzug oder Leben verlieren.

Nachbedingungen: Spieler an Startposition, ggf. Punktabzug.

UC-06: Ausgang erreichen / Level abschließen

Primärer Akteur: Spieler

Vorbedingungen: Spieler erreicht Ausgangsfeld.

Ablauf:

1. System erkennt Spieler am Ausgang.
2. Nachricht: „Level abgeschlossen! Punktestand: X“.
3. Rückkehr zum Menü oder Start des nächsten Levels.

Nachbedingungen: Level abgeschlossen, Score gespeichert.

UC-07: Spiel pausieren / fortsetzen

Primärer Akteur: Spieler

Vorbedingungen: Spiel läuft.

Ablauf:

1. Spieler drückt Pause-Taste.
2. System stoppt Eingaben und Timer.
3. Spieler drückt erneut Pause-Taste.
4. System setzt das Spiel fort.

Nachbedingungen: Spiel pausiert oder fortgesetzt.

UC-08: Level neu starten

Primärer Akteur: Spieler

Vorbedingungen: Level aktiv.

Ablauf:

1. Spieler wählt „Neu starten“ aus dem Menü.
2. Spielsystem setzt Spielfeld, Spieler und Punktestand zurück.

Nachbedingungen: Level in Ausgangszustand.

UC-09: Punktestand anzeigen / aktualisieren

Primärer Akteur: Spieler

Ablauf:

1. Während des Spiels wird Punktestand ständig angezeigt.
2. Nach Levelende wird der Endstand eingeblendet.

Nachbedingungen: Aktueller Punktestand sichtbar.

UC-10: Labyrinth automatisch erzeugen

Primärer Akteur: Spielsystem

Vorbedingungen: Levelstart.

Ablauf:

1. System generiert zufällige Wände und freie Wege.
2. System setzt Start- und Endpunkt fest.
3. Viren und Dateien werden platziert.

Nachbedingungen: Gültiges, spielbares Labyrinth erstellt.

UC-11: Spielende / Zurücksetzen

Primärer Akteur: Spieler

Vorbedingungen: Spieler hat keine Leben mehr oder Levelserie abgeschlossen.

Ablauf:

1. Spielsystem zeigt „Game Over“ und finalen Punktestand.
2. Spieler kann ins Hauptmenü zurückkehren oder neu starten.

Nachbedingungen: Spiel beendet oder neu gestartet.

Schwierigkeitsgrade

- **EASY:** 10×10 Gitter, wenige Viren, niedrige Wanddichte.
- **MEDIUM:** 15×15 Gitter, mittlere Virenanzahl.
- **HARD:** 20×20 Gitter, viele Viren, hohe Wanddichte.

Implementierungshinweise

- Wandwahrscheinlichkeit abhängig vom Level.
- Viren und Dateien nicht auf Start- oder Zielposition platzieren.
- Lives oder Timer als Zusatzfunktion möglich.
- Dialogfenster durch Overlay ersetzen für flüssigeres Gameplay.

Use Case Diagramm: Maze Adventure – Virenjagd

Das folgende Diagramm zeigt die Beziehungen zwischen dem Akteur (Spieler) und den Use Cases des Spiels.

Legende

- Akteur (Spieler): Strichmännchen
- Use Cases: Ovale
- Assoziation: Durchgezogene Linie (Spieler nutzt Use Case)
- Include: Gestrichelte Linie mit <<include>> (Use Case nutzt immer einen anderen)
- Extend: Gestrichelte Linie mit <<extend>> (Use Case erweitert optional einen anderen)

Beziehungen (Include / Extend)

- UC2 (Spiel starten / Level laden) <<include>> UC10 (Labyrinth erzeugen)
- UC3 (Spieler bewegen) <<include>> UC4 (Sichere Datei sammeln)
- UC3 (Spieler bewegen) <<include>> UC5 (Virus treffen)
- UC6 (Ausgang erreichen) <<extend>> UC11 (Spielende / Zurücksetzen)

Use-Diagramm

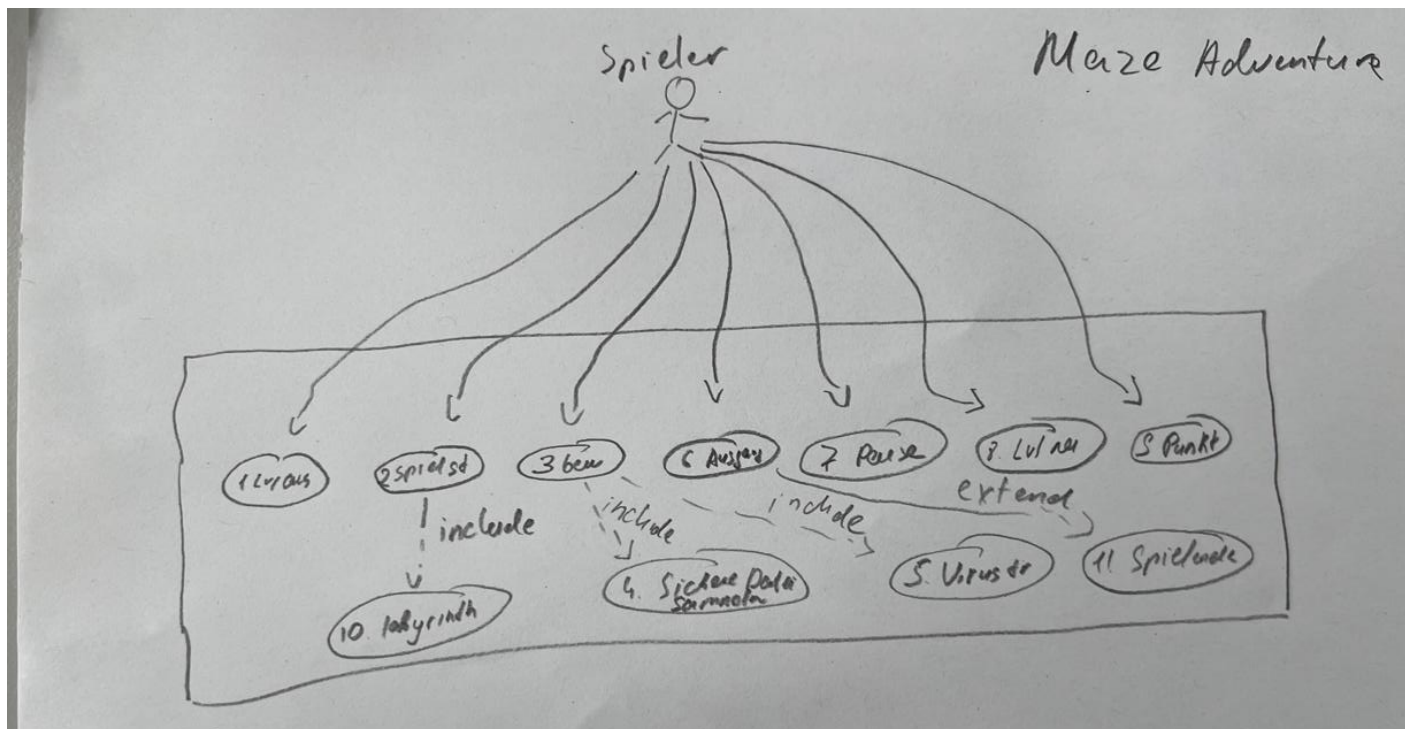


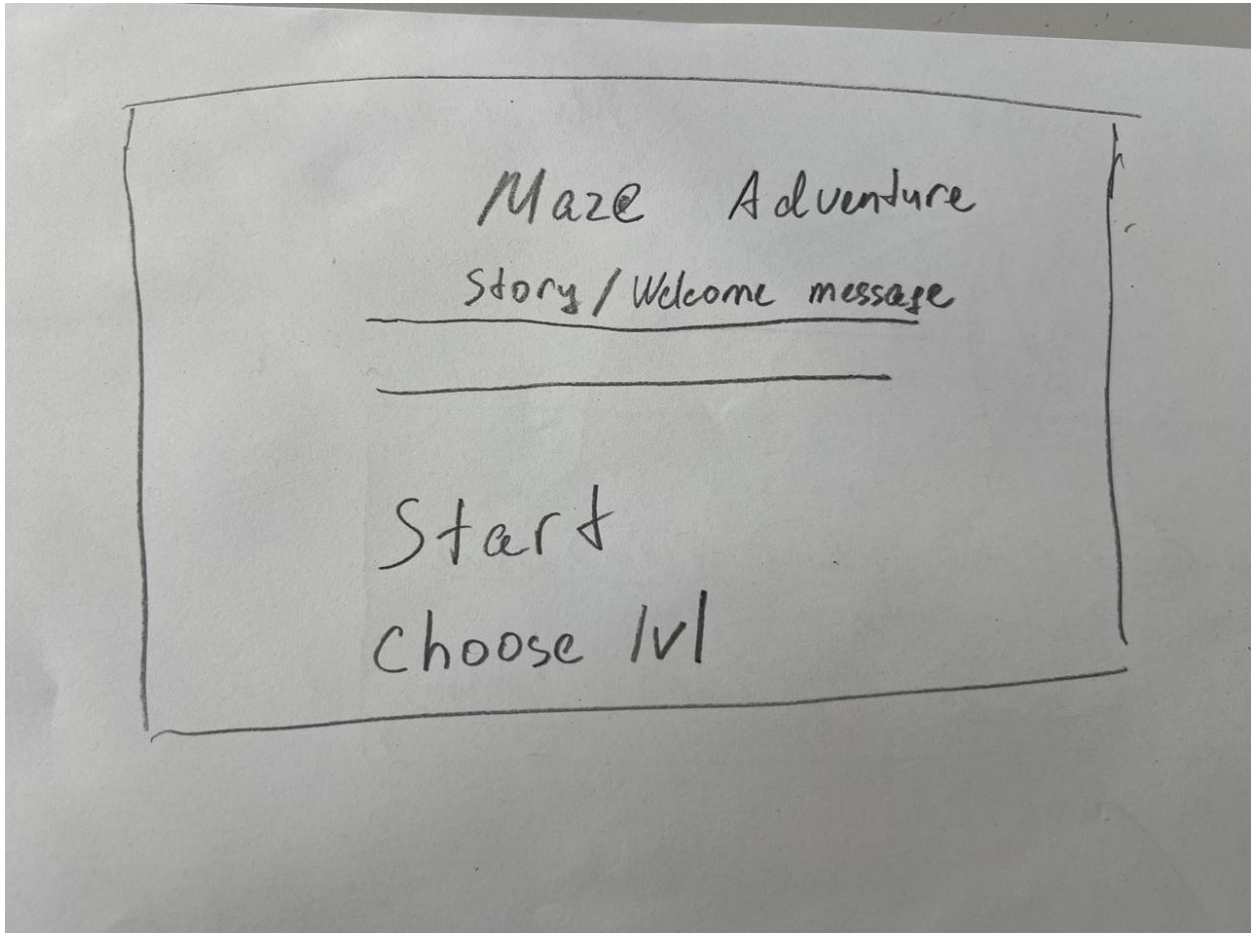
Abbildung: Use Case Diagramm – Maze Adventure – Virenjagd

12 GUI-Prototypen (Papier)

Die folgenden Abbildungen zeigen die GUI-Prototypen der wichtigsten Screens des Spiels **Maze Adventure – Virenjagd**.

12.1 Welcome Screen

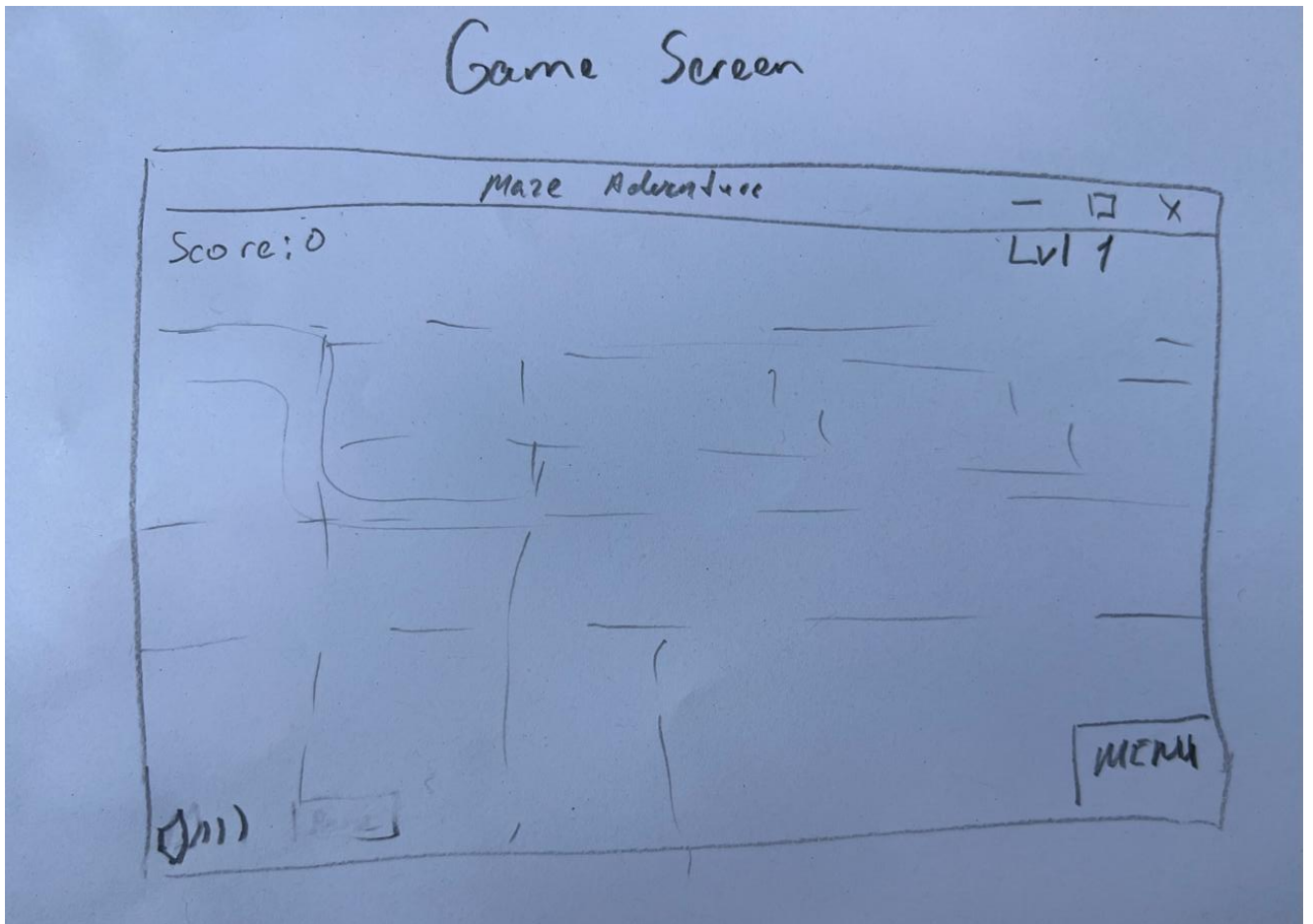
Zeigt: Logo, Welcome Message, **START GAME** Button, **Choose Level** Button.



GUI-Prototyp: Welcome Screen

12.3 Hauptspiel-Bildschirm

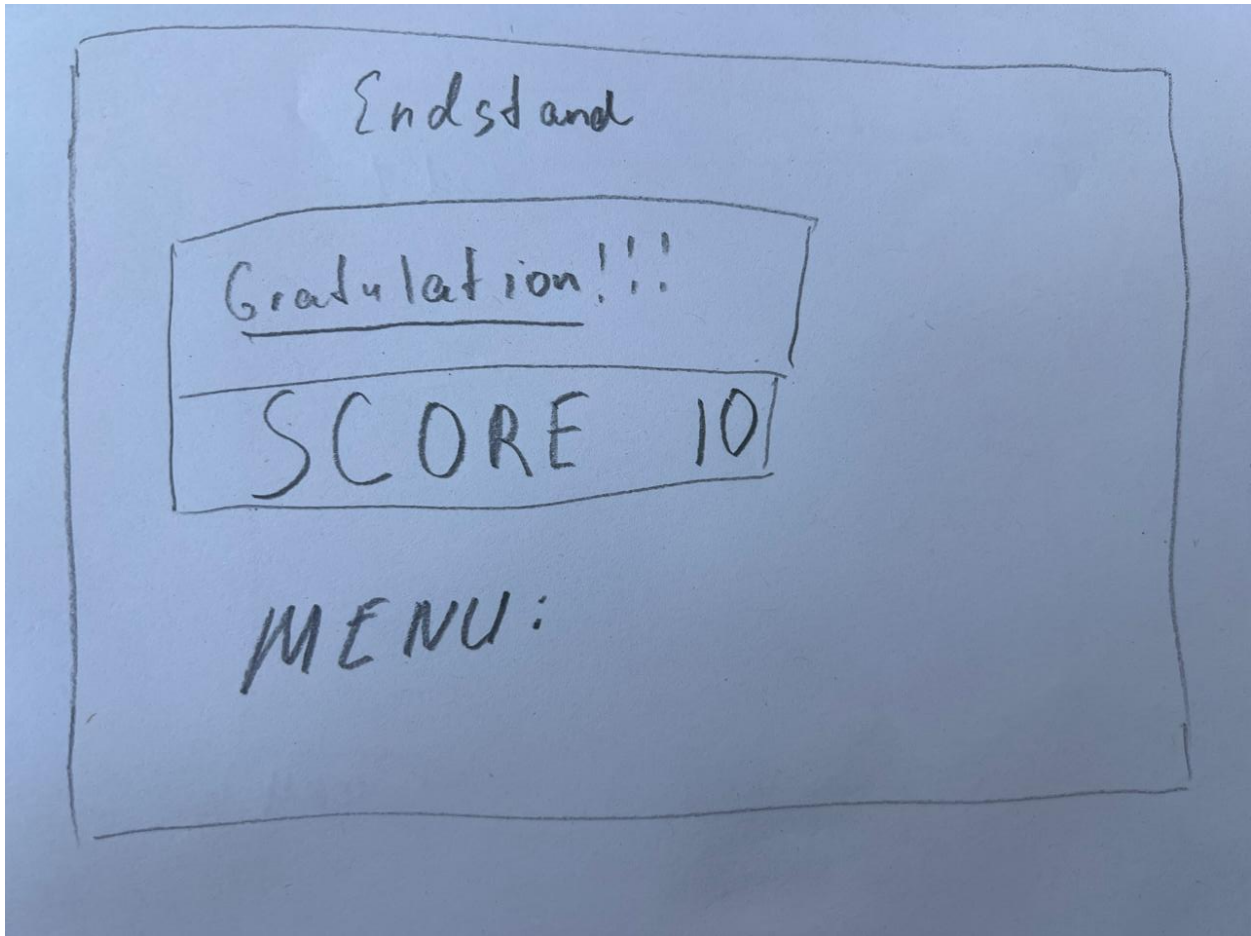
Zeigt: Labyrinth, lvl, Menu button, Score, Volume.



GUI-Prototyp: Hauptspiel-Bildschirm

12.4 Endstand / Gratulation

Zeigt: Endstand nach Levelabschluss, Punktestand und Menü-Button.



GUI-Prototyp: Endstand / Gratulation