

Wiki

Projektübersicht

Projektname: Eventsteuerungssystem
Ziel: Ziel unseres Projekts ist es, die Bewegung von Personen auf einem Veranstaltungsgelände zu simulieren und die Auswirkungen bestimmter Ereignisse zu analysieren.

Team und Organisation

Weekly: Mittwochs
Kommunikation: Discord, Whatsapp

Projektmethodik

Arbeitsweise: Scrum
Sprintdauer: 1 Woche
Projektmanagement-Tool: Redmin
Taskstruktur:
Namenskonvention: USx.Ty.Name

(bsp. US1.T1.Interfaces anlegen)
US = User Story
T = Task

Definition of Done

- Die Funktion ist vollständig umgesetzt und funktioniert wie geplant
- Der Code ist verständlich und ordentlich formatiert
- Es wurde getestet (manuell oder mit automatisierten Tests)
- mindestens 1 Testfall pro Methode
- Eine andere Person hat den Code überprüft (4-Augen-Prinzip)
- Die Änderung ist im Hauptzweig integriert
- Relevante Informationen sind dokumentiert (z. B. im Code oder im Readme)
- Build läuft fehlerfrei
- Der Task ist auf "Close" in Redmine gesetzt und das Team wurde informiert

Story-Points Bewertung

Story-Points	Aufwand
1	bis 2h
2	halber Tag(12h)
3	1-2 Tage
5	Halbe Woche
8	Ganze Woche

Technisches Gesamtkonzept

Komponente	Beschreibung
Simulationskern	Steuert den Ablauf, Ticks, Ereignisse. Basierend auf MASON (SimState, schedule)
Agentenmodell	Agenten mit Rollen wie Besucher, Security, Sanitäter, Angreifer. Unterschiedliches Verhalten (z. B. Panik, Flucht, Hilfe leisten)
Zustandslogik (State Pattern)	Agenten durchlaufen verschiedene Zustände: Schlendert,

Komponente	Beschreibung
	schaut Act, isst, flieht, verletzt. Zustände als eigene Klassen implementiert
Umgebung	Grid oder GeoMASON-Karte, beinhaltet Acts, Ausgänge, Zonen (Toiletten, Essen, etc.)
Eventsystem	Löst Feuer, Unwetter, Schlägereien aus → verändert Agentenzustände und -bewegungen
Acts & Kapazitäten	Main Acts, Side Acts mit begrenztem Platz. Besucher versuchen teilzunehmen
Zonensystem	Orte wie Toiletten, Essensstände, Erste Hilfe – beeinflussen Agentenziele
Notausgangssystem	Exits mit Position, Kapazität, möglicher Blockade. Fluchtverhalten bei Panik
Visualisierung (GUI)	Anzeige in Display2D, ggf. mit Hintergrundkarte (z. B. Olympiapark)

Eingesetzte Technologien

Bereich	Technologie / Tool
Programmiersprache	Java
Simulationsframework	MASON (Multi-Agent Simulation Toolkit)
Agentenlogik	Eigene Klassen + optionales State Pattern
GUI	MASON Display2D, ggf. Hintergrundbild
IDE	IntelliJ IDEA
Versionierung	Git
Build-System	Maven
Tests	JUnit 5
Dokumentation	Markdown

Architekturübersicht (Layer-Modell)

Schicht / Modul	Aufgabe / Inhalt
Simulation Layer	Steuerung der Zeit, Initialisierung, Scheduler (SimState)
Agent Layer	Verhalten einzelner Akteure, Zustände (State Pattern), Ziele, Panikmodell
Environment Layer	Weltmodell mit Acts, Ausgängen, Gefahrenquellen, Zonen (Essen, WC)
Event Layer	Modellierung und Auslösung von Feuer, Unwetter, Schlägereien
GUI Layer	Visualisierung mit MASON Display2D, Darstellung von Agenten und Zonen