

Generative AI Agents for Minecraft: Automating Creative World Tasks with Amazon Bedrock


Team Members

Taesh Azal Assadi
Sanjna Kumari
Faraz Ulhaq Shah


February 23, 2025

1 User Personas


1.1 User Persona 1 – New Minecraft Server Owner

Name & Picture	Details	Goal
<div></div> <div>Alex Carter</div>	<p>Alex Carter is a 20-year-old college student working part-time in retail. He enjoys sandbox and multiplayer games and plays Minecraft regularly with friends. While he understands the basics of setting up a Minecraft server, he finds large-scale world-building and minigame design overwhelming. He has a moderate level of technical proficiency but lacks experience with advanced Minecraft plugins and scripting. Due to his busy schedule, he wants a quick and easy way to create an engaging multiplayer experience.</p>	<p>Alex wants to launch a unique multiplayer Minecraft server without spending weeks manually constructing the world. He needs an AI assistant that can generate landscapes, structures, and interactive features using simple chat commands. Ideally, he wants an intuitive tool that allows him to customize game elements, making the server feel unique while minimizing the effort required to maintain it.</p>

1.2 User Persona 2 – Content Creator

Name & Picture	Details	Goal
<div></div> <div>Priya Mehta</div>	<p>Priya Mehta is a 25-year-old YouTuber and Twitch streamer based in the UK. She creates Minecraft content, including role-playing series, challenge videos, and interactive streams. Priya is tech-savvy, frequently using mods, plugins, and commands to enhance her content. However, manually designing new worlds and minigames is time-consuming and takes away from her ability to interact with her audience. She needs a tool that enables her to generate visually striking environments instantly while allowing some degree of customization.</p>	<p>Priya wants to streamline her content creation by using AI-generated worlds and minigames. Instead of spending hours preparing, she wants to type simple commands like ”Create an arena for a parkour challenge” or ”Generate a haunted mansion for a horror-themed stream” and have the AI handle the rest. This will help her keep her content fresh, boost engagement, and allow her to focus more on audience interaction and storytelling.</p>

1.3 User Persona 3 – Educator

Name & Picture	Details	Goal
<div></div> <div>Dr. James Thornton</div>	<p>Dr. James Thornton is a 45-year-old high school teacher in Canada who specializes in STEM education. He uses Minecraft as a teaching tool to make learning interactive and immersive. While he is comfortable using Minecraft Education Edition, he struggles with creating large-scale historical landmarks and scientific simulations. He lacks the time and expertise to manually design custom educational builds for every lesson.</p>	<p>Dr. Thornton wants a way to quickly generate historical landmarks, scientific experiments, and educational builds inside Minecraft to enhance student engagement. He needs an AI-powered assistant that can instantly create lesson-based environments, such as an ancient Egyptian pyramid for history class or a math simulation for his lessons. The AI should be simple to use, allowing both him and his students to generate content with minimal effort.</p>

2 User Stories

1. **Quick World Generation for New Server Owners**
As a new server owner,
I want to generate a fully designed world with AI,
So that I can start my Minecraft server without spending hours building manually.
2. **Minigame Creation for Content Creators**
As a content creator,
I want to use AI to create interactive minigames like hide-and-seek or parkour challenges,
So that I can produce engaging content without spending hours on world design.
3. **AI-Assisted Historical Builds for Educators**
As an educator,
I want to use AI to generate historical landmarks and educational models,
So that my students can explore and learn interactively within Minecraft.
4. **Server Automation for Experienced Admins**
As a Minecraft server administrator,
I want to automate seasonal event builds with AI,
So that I can keep my server fresh without manually redoing large-scale changes.
5. **Dynamic NPC and Quest Creation**
As a role-playing server owner,
I want to generate AI-powered NPCs and questlines,
So that players can have engaging story-driven experiences in my Minecraft world.
6. **Custom Environment Generation for Storytelling**
As a Minecraft storyteller,
I want to instantly generate thematic environments like dystopian cities or fantasy kingdoms,
So that I can immerse my players in unique and visually appealing worlds.
7. **AI-Based Terraforming for Large-Scale Projects**
As an advanced builder,
I want to use AI to automate terrain modifications like mountains, caves, and lakes,
So that I can focus on refining the final details rather than manually shaping landscapes.
8. **Real-Time Chat Interaction with AI**
As a casual Minecraft player,
I want to give AI commands through Minecraft chat,
So that I can build and modify my world without switching to external apps or scripts.
9. **AI-Generated Puzzle and Challenge Maps**
As a puzzle map creator,
I want to generate AI-crafted escape rooms and challenge maps,
So that I can provide engaging brain teasers for my players with minimal effort.
10. **AI-Driven Adaptive Gameplay Adjustments**
As a server owner running a live event,
I want to have AI adjust challenges, spawn structures, or modify rules dynamically,
So that I can keep players engaged without manually managing every detail.

3 Scrum

3.1 Product Backlog - Prioritized User Stories, Tests, & Tasks

User Story	Tests (Acceptance Criteria)	Simplified Tasks	Est. Time
Quick World Generation for Server Owners	AI generates a world instantly based on a simple command. Structures and landscapes match expected themes (e.g., “medieval city”). No major performance issues on the Minecraft server.	Set up AWS Bedrock access and enable AI models (2h). Install Mineflayer for bot connection (3h). Modify agent commands to recognize world-gen inputs (3h). Deploy CloudFormation stack to set up agent (3h). Test structure placement and refine AI responses (3h).	14h
Minigame Creation for Content Creators	AI successfully generates playable minigames (e.g. parkour, hide-and-seek). Minigames function without breaking mechanics (traps, timers, scoring). Commands allow customization (difficulty, arena size).	Implement basic AI command processing (6h). Add game rules and logic execution (6h). Integrate AI with Mineflayer actions (5h). Deploy via AWS (5h).	22h
AI-Assisted Historical Builds for Educators	AI generates accurate historical landmarks upon request. Builds match key characteristics (e.g., “Roman Colosseum”). Simple command-based generation for easy classroom use.	Train AI to understand historical landmark structures (6h). Implement chat commands for requesting historical builds (5h). Test and refine build accuracy and response time (3h).	14h
Server Automation for Experienced Admins	AI schedules and executes seasonal/world events automatically. Admins can configure event timing and scaling via chat commands. AI does not interfere with existing server functions.	Implement basic event scheduling in AI (5h). Add admin commands to start/stop automated events (4h). Test server automation with multiple events (4h).	13h
Dynamic NPC and Quest Creation	AI generates NPCs with interactive questlines. Players can interact and progress through AI-generated quests. AI ensures variety in NPC dialogue and tasks.	Add basic NPC behavior and interactions (8h). Implement simple quest generation logic (8h). Deploy AI bot to AWS CDK for cloud-hosted execution (6h). Test NPC behavior and player interactions (5h).	27h

3.2 3.2 Product Roadmap

Take the list of high-level user stories you created and break them out into a Product Roadmap, specifying which features will be delivered in which sprint. Keep in mind, some of your features may be prerequisites for other features in later sprints. You will have three (3) sprints during the semester, so plan accordingly.

3.2.1 Sprint 1 (Weeks 5–8) – Foundational AI & Basic Features

Key Focus:

Setting up AI infrastructure (AWS Bedrock, Mineflayer, CloudFormation).

Implementing basic AI world-generation commands for simple structures.

Executing and testing early AI interactions in Minecraft.

Features Delivered:

- Quick World Generation for Server Owners
- Basic AI Commands Execution (Building Basic Structures)
- Backend Setup & NLP Integration

Tasks Breakdown:

- Week 5: AI Model Setup & Backend Prep (AWS Bedrock, API, CloudFormation Setup)
- Week 6: AI Command Execution (Basic structure generation & connection to Mineflayer)
- Week 7: Testing & Refinement (AI-generated builds optimization)
- Week 8: Sprint 1 Review & Planning for Sprint 2

Deliverables:

- Basic AI Commands Executed in Minecraft
- Testing & Refinement of AI Builds
- Sprint 1 Deliverable

3.2.2 Sprint 2 (Weeks 9–11) – Advanced AI & Automation

Key Focus:

Enhancing AI structure generation with complex builds.

Introducing multi-command processing and real-time AI execution.

Optimizing AI to reduce response time and improve accuracy.

Features Delivered:

- AI-Assisted Historical Builds for Educators
- Server Automation for Experienced Admins
- AI Generating Complex Structures

Tasks Breakdown:

- Week 9: Complex Structure Generation (AI generates detailed buildings & terrains)
- Week 10: Real-Time Execution Testing (Optimizing AI's response handling in AWS Bedrock)
- Week 11: Review & Improvements (Testing full structure & terrain generation, optimizing AI accuracy)

Deliverables:

- AI Generating Structures in Minecraft
- Real-Time AI Execution Testing Completed
- Sprint 2 Deliverable (Builds & Real-Time Testing)

3.2.3 Sprint 3 (Weeks 12–14) – Interactive Gameplay & Minigames

Key Focus:

Implementing interactive elements and multi-command AI capabilities.

Developing AI-generated minigames & automated server events.

Finalizing the system with bug fixes & performance tuning.

Features Delivered:

- Dynamic NPC and Quest Creation
- Minigame Creation for Content Creators
- Interactive Elements & Automated Server Events

Tasks Breakdown:

- Week 12: AI-driven interactive elements & multi-command processing
- Week 13: Minigame & Server Events implementation (hide-and-seek, automated builds)
- Week 14: Final Testing & Review, bug fixes, and performance optimization

Deliverables:

- Basic Interactive AI Features
- Minigames & Automated Events Implemented
- Sprint 3 Deliverable

Week 15 – Final Product Submission

Final Deliverables:

- Full System Demo Video
- Technical Documentation & User Manual
- Final Performance Testing & Bug Fixes
- Final Product Submission

3.3 Product Burndown Chart

This is the burndown chart based on the product roadmap giving a week by week breakdown where the total effort is 90 hours distributed among 3 team members. Here the planned and actual graphs are overlapping as the sprint starts from tomorrow and the value of actual tasks will change as the sprint progresses.

SPRINT WEEKS	Planned	Actual
24-Feb-25	90	90
3-Mar-25	81	81
10-Mar-25	73	73
17-Mar-25	63	63
24-Mar-25	55	55
31-Mar-25	45	45
7-Apr-25	36	36
14-Apr-25	27	27
21-Apr-25	17	17
28-Apr-25	7	7
5-May-25	0	0

Figure 1: Planned vs. Actual tasks (Excel sheet table)

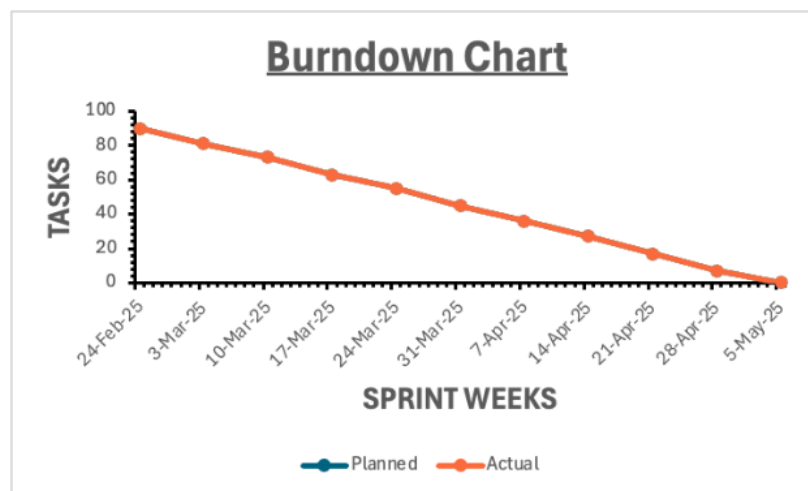


Figure 2: Burndown Chart based on Planned and Actual values

4 AWS Usage and Status of our Proposal to WISE Students

We plan to utilize **AWS Bedrock** and **AWS ECS** to power AI-driven automation within a Minecraft server. AWS Bedrock will process natural language commands from players, generate structured build instructions, and interact with the Minecraft Bot for executing tasks. AWS ECS (Elastic Container Service) will be used to deploy and manage containerized applications, ensuring scalable and efficient execution of the AI-powered bot within the Minecraft environment. This integration will enable seamless automation for building structures, assisting players, and enhancing overall gameplay interactivity.

On February 10th, 2025, we had a meeting with WISE student **Ruba Vignesh**, where we discussed the usage of Amazon Bedrock for our project. During the discussion, we were advised to submit an architecture diagram along with an AWS pricing estimate for further evaluation. Based on this guidance, we are currently preparing the necessary documents and plan to submit them around the week of February 24th, 2025, which marks the start of Sprint 1. This submission will provide a structured overview of our planned implementation and associated costs, ensuring alignment with the project requirements and expectations set during the meeting.

5 GitLab Setup

We have created and set up our GitLab repository at the following URL:

<https://code.vt.edu/cs-5934-spring25/group-10/generative-ai-agents-for-minecraft-automating-creative-world-tasks-with-amazon-bedrock>

This repository contains our project-related documents, including a **README.md** file. We will continue to populate the repository with updates to our code, documentation, and any additional materials as the project progresses.