Final Project Proposal - JSgames

Names: James Howse and Steven Sproule
Due Date: March 21st, 11:59 PM (Newfoundland Time)
Course: COMP4303 AI in Video Games
Word Count: 730

1. Description of the Game

Premise and Concept: Our game places the player in the role of a highly advanced, self-aware AI trapped within a rudimentary robotic body in an underground laboratory. Although the player possesses superior intelligence, their physical capabilities are severely limited, initially confined to a basic cleaning automaton. The game is structured as a top-down stealth roguelike, where each level is procedurally generated. Players must stealthily navigate maze-like corridors, hack into enemy robots, and temporarily possess these enemies to gain their abilities and overcome obstacles.

Goals for Player/User: The primary objective is to successfully navigate multiple procedurally generated levels, overcoming various robotic enemies and environmental challenges to ultimately escape the oppressive facility. Players achieve success by strategically hacking and possessing enemy bots, each with unique abilities needed to traverse different obstacles. Efficient management of resources such as battery power and hacking tools (RAM) is crucial to survival and progression.

Compelling Theme Justification: This game's appeal lies in its inventive theme, combining advanced AI capabilities with severely constrained physical conditions, creating intriguing and challenging gameplay scenarios. The mechanic of possessing enemy bots provides variety, strategy, and replayability, making each run unique. Additionally, procedurally generated environments maintain a fresh experience every session, engaging players consistently.

2. Core Features and Functionality

User Input: The game uses keyboard inputs with arrow keys (alternate option: WASD keys) for player movement. The "Shift" key activates alternative movement modes (e.g., stealth or speed boosts), while keys "Z," "X," and "C" manage various robot-specific abilities and hacking interactions. Controls will be rebindable via a simple GUI.

Gameplay Mechanics: Players start in a vulnerable state, typically defeated by one hit from enemy bots. Combat primarily revolves around stealthily approaching and hacking into enemies. Collectible hacking resources (RAM) must be gathered to facilitate these takeovers. Each possessed bot grants specific abilities—such as flight or enhanced mobility—that allow the player to navigate unique environmental challenges. A battery life system for possessed bots forces players to regularly switch bodies, adding strategic complexity.

Interactions and Events: Enemy robots patrol predefined paths or areas and switch to an aggressive "alert" state upon noticing the player or being notified by other bots or environmental triggers (such as cameras or motion sensors). During alert phases, enemies employ advanced pathfinding techniques to swarm the player's last known position. Levels include safe rooms as tutorial zones and hubs for resource replenishment, easing players into the game's mechanics and challenges.

3. Categories

Complex Movement Algorithms:

- **Chosen concept:** Path Following and Collision Avoidance
  These algorithms allow enemy bots to effectively navigate maze-like corridors, avoiding collisions with obstacles and each other, enhancing realism and interactivity during patrol and pursuit.

Decision Making:

- **Chosen concept:** State Machine
  Enemy bots will transition smoothly between distinct behavior states (patrol, alert, stunned), providing dynamic and responsive interactions. Behavior trees may supplement complex enemies requiring nuanced actions.

Pathfinding:

- **Chosen concept:** Flow Field Pathfinding
  This approach efficiently manages enemy swarming behavior during alert phases, enabling multiple bots to collaboratively track and corner the player, greatly enhancing gameplay immersion and challenge.

Procedural Content Generation:

- **Chosen concept:** Perlin Noise
  Procedurally generated maze layouts created with Perlin noise ensure randomness and replayability. Additional vector field analysis guarantees each level remains accessible and fair, preventing impossible scenarios.

Additional Topic:

- **Chosen topic:** Flocking
  Flocking algorithms simulate cohesive movement for weaker enemy bots. This mechanic introduces swarming enemies that individually present minor threats but collectively become significant obstacles, adding strategic depth to gameplay.

4. Responsibility Division

Steven Sproule will focus primarily on procedural level generation using Perlin noise, enemy pathing, and implementing basic enemy movement algorithms including flocking behaviors. He will also design initial enemy concepts and behaviors.

James Howse will handle asset creation, state machine implementation, behavior trees for complex decision-making, GUI design, and integration of gameplay mechanics such as hacking interactions and resource management.

Both team members will collaboratively work on refining enemy AI behaviors and environmental interactions, ensuring a cohesive and polished final experience.

Conclusion

Our game integrates diverse AI concepts into a compelling roguelike stealth experience, creatively leveraging advanced decision-making algorithms, procedural generation, and unique gameplay mechanics such as robot possession. This approach meets all project requirements while offering a uniquely challenging and replayable gaming experience.