

Building Security Simulator

Software Requirements Specification

Version 1.1

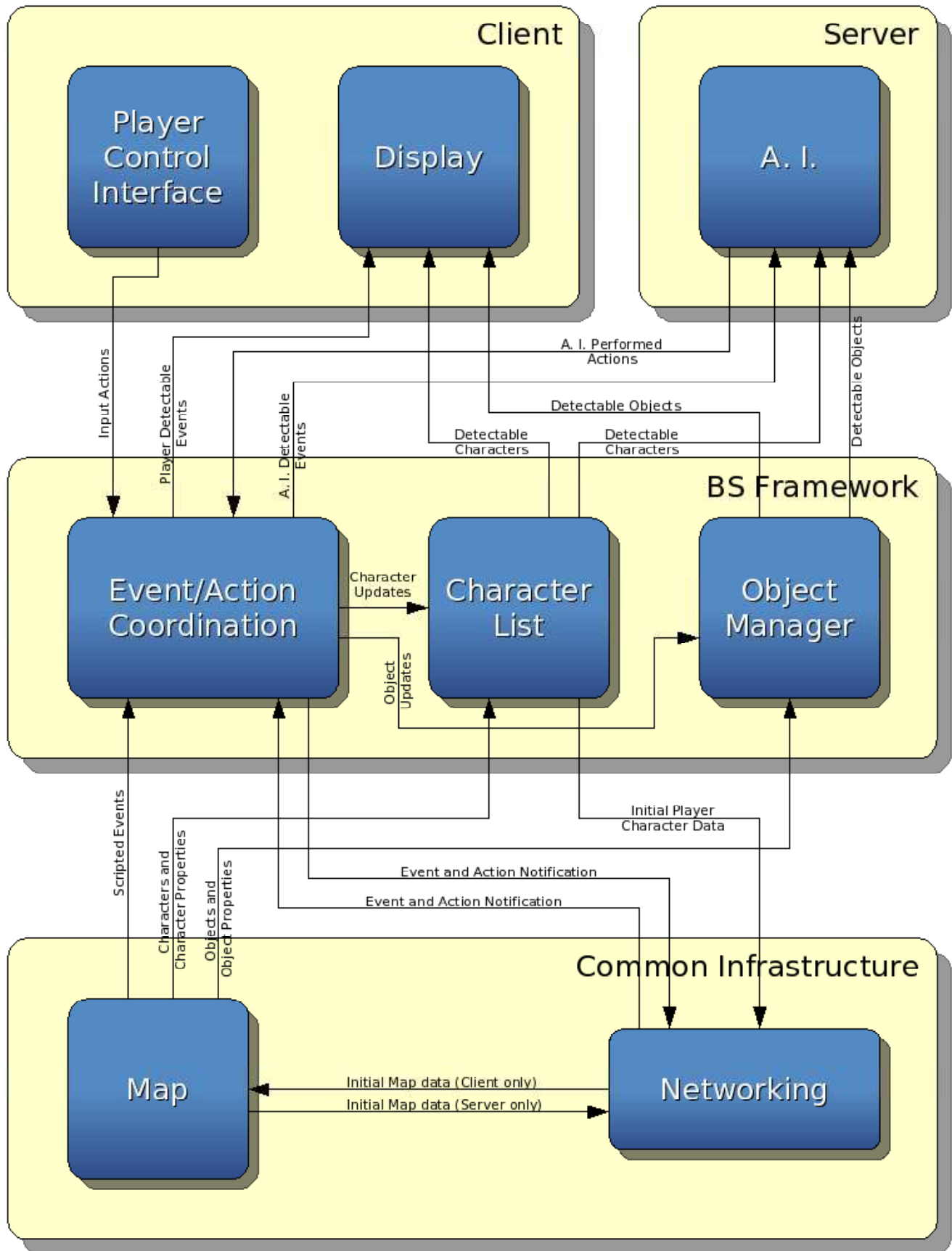
Project Members

- Matthew Fulgo
- Lane Haury
- Craig Szczesiul
- Tal Ron

Table of Contents

- I. Architectural Overview
- II. Software Requirements
 - A. Server
 1. Artificial Intelligence
 - B. Client
 1. Player Control Interface
 2. Display
 - C. BS Framework
 1. Event/Action Coordinator
 2. Character List
 3. Object Manager
 - D. Common Infrastructure
 1. Map
 2. Networking
- III. RDD Reference

Architectural Overview



Software Requirements

A. Server

1. Artificial Intelligence System

- a. Some characters may be controlled by A.I. (RDD C.8).
- b. All A.I. controlled character's information must be relayed in the same way as player characters (RDD A.1, RDD C.9).
- c. Minimally, the A.I. character must have the ability to move through the map in a realistic fashion and perform actions on objects in the environment (RDD A.13, RDD C.9).
- d. The A.I. can input the same set of actions as can players (RDD A.12, RDD C.9).
- e. A.I. characters will peripherally be able to send short messages describing events.

B. Client

1. Player Control Interface

- a. The client will facilitate the player's control of his/her character.
- b. The client will prevent the player from performing illegal actions such as opening locked doors and breaking the laws of the physics model (RDD A.8).
- c. Players will input their character's actions through input devices (e.g. mouse or keyboard) (RDD A.12).
- d. Player characters will be able to communicate through a chat window (RDD A.16).
- e. Methods of input must allow full range of proper character movement (RDD A.13).
 - i. Characters must be able to rotate left and right.
 - ii. Characters must be able to move forward, backward, left, and right.
 - iii. Characters must be able to ascend and descend stairs, escalators, and elevators.

2. Display

- a. The client will provide each player a visual display of what resources are available to them and the environment, including their character and the visible objects around them (RDD A.22).
- b. All actions within a character's sphere of influence will be relayed to the client from the server (RDD A.9). A character's sphere of influence is the spherical radius within which the character can detect events (RDD E).

C. BS Framework

1. Event/Action Coordination

- a. The Event/Action Coordinator (EAC) will be in charge of handling and creating all events and actions.
- b. Each event will be either caused by an action, scheduled by the simulation, or randomly created (possibly weather, sensor malfunctions, or other peripheral additions).
- c. Some events will also be caused by characters, objects, and player input. The EAC will handle these events as necessary.
- d. The events will be communicated to both the server and the client through use of the network (RDD A.2, RDD A.5).
 - i. The Server EAC will receive and process actions and send events to the appropriate characters.
 - ii. The Client EAC will receive and process events and send actions to the Server.
- e. Communication between characters will be handled as actions and events. These actions include talking and peripherally shouting, radio, and whispering (RDD A.16).
- f. The EAC will be given updates by the server when police and fire alarm responses are created by a character (RDD A.17).
- g. The Server's EAC will receive information about sounds and movement and will send the necessary updates to the characters (RDD A.20).
- h. All characters will be able to perform actions that will change the simulation's environment. These include manipulating doors, moving objects, acquiring resources, watching or listening to surroundings, walking, and communicating (RDD A.6).
 - i. Some actions will cause events such as sound (location, magnitude), environment alteration, and location changes in the simulation (RDD A.7).
 - ii. Other actions, such as monitor viewing, will not trigger events.
- i. Monitor viewing will grant players a view of an area (RDD A.11).

- j. A sound event shall encapsulate its source location and magnitude. The intensity of the sound event to the detecting character/sensor shall be calculated based on the distance between the source and the detecting character/sensor and the magnitude of the sound (RDD A.20).
- k. Movement events shall encapsulate location and velocity information. This will allow for varying sensitivities in motion detection based on the magnitude of the velocity, size, and location of the moving object (RDD A.20).
- l. As characters acquire resources, their access and abilities are altered by the EAC (RDD A.15).

2. Character List

- a. The Character List will store a reference to each character object within the simulation (RDD A.1).
- b. Each character object will be supplemented by its location, orientation, status, sphere of influence, resources, and capabilities (RDD A.1).
 - i. This includes the objective for each character, if one exists (RDD A.19).
 - ii. This also includes the base set of resources with which the teams begin each round (RDD A.21).
- c. A character will potentially come in two forms: Player-controlled and A.I.-controlled (RDD C.8).
- d. Characters should also be able to move through the environment (RDD A.13).
- e. The Character List will store the resources carried by each character (RDD A.15).

3. Object Manager

- a. The Object Manager will store a reference to each object within the simulation (RDD A.1).
- b. Each object will be supplemented by its location and status (RDD A.1).
- c. An object will be defined as any physical entity within the simulation.
- d. Objects may not move through other objects (RDD A.14).
- e. Some objects may move or alter other objects.
 - i. Only character objects may alter objects.
 - ii. Any object can move another if they come into contact. This movement will be controlled by the physics model.

D. Common Infrastructure

1. Map

- a. The Map must contain the layout of the environment, including all immovable objects and characters' starting locations.
- b. The Map shall contain the location of sensors and their orientations at the start of each round (RDD A.18).

2. Networking

- a. The network will have to send information between the server and client.
 - i. The network must send actions from clients to the server (RDD A.2).
 - ii. The network must send events from the server to the clients as determined by the Server's EAC.
- b. At the beginning of each round, the network will have to send information about the map and objects from the server to the clients (RDD A.4).
- c. The network will continuously send updates regarding the characters, sensors, and events from the server to the clients (RDD A.3, A.10).

RDD Reference

A. Functional Requirements

- 1. The simulation must maintain a list of characters and objects and data specific to each of them. This includes:
 - a. character status
 - b. location in map coordinates
 - c. environmental information the character can currently access
- 2. The following data must be transferred from the client to the server over a network protocol:
 - a. location updates

- b. actions
3. The following data must be transferred continuously (streamed) from the server to the client over a network protocol:
 - a. location and status updates of objects and characters within the character's sphere of influence.
 - b. sensor data to which the character has access.
 - c. actions marked as required for the given client
4. The following data must be transferred once from the server to the client at the initialization of each round:
 - a. map information
 - b. object list
 - c. properties for each object and character
5. Actions must be declared as either required or not-required to be sent to the client. Public actions are defined as those which have a sensory effect on the game environment. Private actions, such as viewing a camera monitor, do not potentially affect other characters or the game environment. Universal actions are marked as required for distribution to every character.
6. Actions will include:
 - a. opening, closing, locking, and unlocking doors
 - b. changing the location of movable objects
 - c. acquiring keys (including the stealing of keys from guards)
 - d. watching camera monitors
 - e. walking
 - f. communicating
7. Actions can cause
 - a. sounds, which have a location and magnitude and may be represented visually
 - b. visual cues
 - c. changes in character status
 - d. changes in object properties (e.g. opening doors, moving boxes, blocking hallways)
8. Actions should also have limitations (e.g. can't open a door without an access card)
9. All actions within a character's sphere of influence must be relayed to that client.
10. The server will process sensor data in order to send that data to clients designated for receiving that information.
11. Characters must be able to access sensor data by using monitors.
 - a. Use of monitors will be limited by a characters proximity to the monitor
 - b. Camera monitors will grant players a view of an area.
 - c. Monitors for motion sensors and infrared beams will convey a map location if triggered
 - d. Monitors will be configured to determine which sensors' data they will show.
12. Players must input control of character actions to the game. Players will control character movement, communication, and interactions with the environment and each other. This control will be achieved through human input devices such as a keyboard and mouse.
13. Characters must be able to move throughout the environment: they must be able to rotate left and right and move forward, backward, left, and right. Characters must be able to ascend and descend stairs, escalators, and elevators.
14. Objects may not move through other objects.
15. Characters must be able to acquire core resources. These include:
 - a. keys / access cards
 - b. radios (peripheral feature)

16. Characters will be able to communicate with each other through a chat mechanism, subject to constraints on the mode of communication used. These include:
 - a. talking (default)
 - b. text chat (peripheral)
 - c. shouting (peripheral)
 - d. radios (peripheral)
 - e. whisper to nearby characters (peripheral)
17. Include support for time-triggered events. Characters can initiate a one-shot timer for the following events:
 - a. police arriving – the level of response/number of police characters will be determined by the properties of the map
 - b. fire alarms (peripheral)
18. Maps shall contain sensor locations and orientations at the start of each round.
19. Intruders will have a set of objectives in each round. "Victory" is obtained upon the fulfillment of those objectives.
20. Sensors and characters must be able to detect the following events:
 - a. sounds
 - b. movement
21. Each character starts a round with a base set of objects. These objects will be defined by the map. These include:
 - a. keys / access cards
 - b. flashlight (peripheral)
 - c. radio (peripheral)
22. Each player's client display will indicate/list what objects their character currently carries.
23. The server shall be an instance of the platform that contains all of the data in the simulation and does not allow direct control by players.

B. Non-Functional Requirements

1. Passage of time uses a 1:alpha mapping
2. Maps shall have realistic sensor quantities and placement.
3. Sensors shall model real-life, having limitations on use/abilities, failures, sensitivity levels, and the potential for bypass.
4. Guards shall not know the intruders' objectives.
5. Characters' actions govern what information the character can access. This information will have realistic limitations based on:
 - a. field of view
 - b. sound propagation
 - c. map location
 - d. proximity to objects

C. Peripheral Functional Requirements

1. Report statistics at the end of the round to help evaluate the guards' performance (high priority)
2. Sensors/objects must be placeable for a map/scenario
3. Aspects of players (health, movement speed, etc.) must be able to be changed within the game

4. An inventory system of what players can carry / are carrying
5. Players can trade inventory items with each other
6. Model night and day situations
7. Guards may have a prioritized list of things to guard and/or objectives
8. If any character in the game is not controlled by a player, it will be controlled by a component that handles artificial intelligence (A. I.).
9. The A. I. characters will have all the capabilities and limitations of player-controlled characters (e.g. an objective, sphere of influence, performing actions).

D. Peripheral Non-Functional Requirement

1. Limited funds should be given to the players (guards and intruders) to reflect real-world budgets for sensors (cameras, door alarms, etc.)

E. Glossary

Action

Happenings within the simulation that are directly caused by characters. Interactions with physical objects, the environment, or characters (e.g. walking, opening doors, and arresting intruders).

Character

Representation of a person within the game environment. A guard, intruder, etc.

Client

The program a player uses.

Event

Happenings within the simulation that are not directly caused by a character's action

Guard

Characters that attempt to detect, oppose, and capture or otherwise incapacitate the intruders.

Intruder

Characters whose objective(s) are given at the beginning of each round, usually involving destruction, theft, or other crimes.

Monitor

An in-game object used to relay sensor information to a character.

Objective

A goal presented by the game to a specific character or group of characters at the beginning of each round.

Player

A Real Life™ person.

Round

The period of game time beginning when objectives are given to the intruders and ending at the completion or failure of these objectives.

Sensor

An object within the game environment that captures information and is processed by the server.

Sphere of Influence

A three dimensional distance from the character in which a character or sensor can detect actions.

Status

An enumeration of a character's state, which defines their abilities and limitations (i.e. active/normal, incapacitated, captured, dead).

Attachments

- [Architecture.svg](#) (34.8 kB) - added by fulgo on 02/26/07 09:34:16.
- [Architecture.odg](#) (12.8 kB) - added by fulgo on 03/05/07 22:53:53.
- [Architecture.png](#) (95.3 kB) - added by fulgo on 03/05/07 23:00:06.