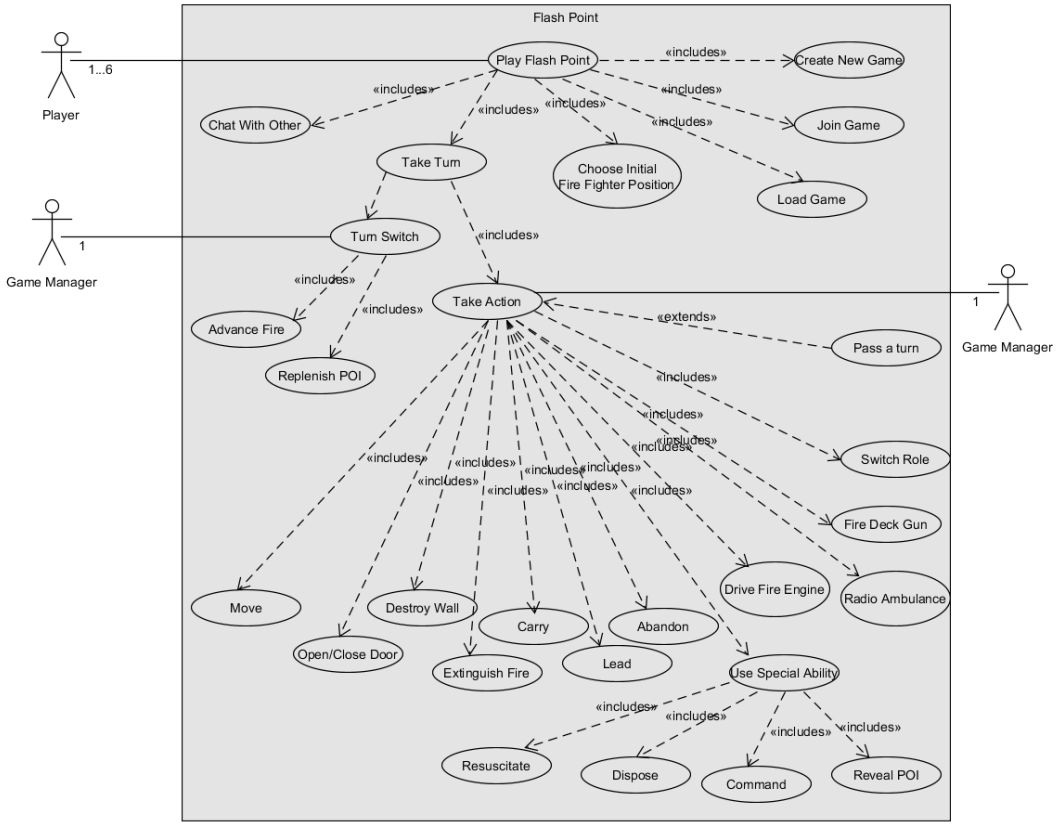

Play Flash Point

Use Case: Play Flash Point
Scope: Flash Point
Level: User Goal
Intention in Context: The intention of the *Player* is to play a game of flash point with his/her team mates.
Multiplicity: Multiple players can play flash point concurrently. A given player is not allowed to play multiple games simultaneously.
Primary Actor: *Player*
Secondary Actors: *Player (who play the roles of team mates).*
Main Success Scenario:

1. Player enters game menu.
2. Player chooses to either Join/Load/Create a game.
   Step 3 is executed after room owner starts the game.
3. Player chooses initial position for firefighters.
4. Take turns.
   Step 4 is repeated until players win/lose.
5. System informs player if they won/lost the game.

Extension:
2a. If the game is a saved game, step 4 is executed.
2b. If the player is not able to join, create, or load the game, repeat step 2.
3a. If the player is playing in experienced mode, player also choose together initial

position to place fire engine and ambulance.
4a. If enough players want to continue to get a perfect win, the game continues.

Join Existing Game

Use Case: Join Existing Game
Scope: Flash Point
Level: Subfunction
Intention in Context: The intension of the Player is to join a server that someone else created and is not started yet.
Primary Actor: Player
Secondary Actor: Other Players, Server Owner
Main Success Scenario:

1. Player informs System that he wish to join in a specific server.
2. System presents game lobby to Player.
3. Player chooses a character informs System that he is ready to begin the game.
4. Player waits all other players to ready and the server owner start the game.

Extensions:
2a. Player informs System that he wishes to exit this lobby. Use case ends in failure.
4a. If server owner exits the lobby, player exits as well. Use case end in failure.

Load Game

Use Case: Load game
Scope: Flash Point
Level: subfunction
Intention in Context: load the game that early saved
Multiplicity: Multiple players can load the same game concurrently. A single player can load different game.
Primary Actor: Player
Main Success Scenario:

1. System presents the Player a list of saved games.
2. Player chooses the desired game to load.
3. System shows player the game lobby.
4. System detects if the game is ready to be started.
5. Player informs the system to start the game.

Extensions:
2a. The saved game file is damaged; use case ends in failure.
(3-5)a. Player changes his mind and doesn't want to play anymore, he can exit; use case ends in failure.

Create New Game

Use Case: Create New Game
Scope: Flash Point

Level: Subfunction
Intention in context: The intention of the player is to start a new game from scratch.
Primary Actor: Player
Main Success Scenario:
1. Player informs the system to create a new game for x number of players.
2. System presents details of games for players to choose. (Map, difficulties, fire etc.).
3. Player informs System the detail of game settings.
4. System presents the player the created game lobby.
5. System detects if the game is ready to be started.
6. Player informs the system to start the game.

Extension:
2a: Time is limited for player to choose types. if player does not finish, system will automatically choose one mode.
3a: When ready bottom is not pressed within limited time, system will set it ready. Use case continues at step 2
(2-6)a : Player informs Systems that he/she wishes to cancel game creation. Use case ends in failure.

Place Starting Positions

Use Case: Place Starting Positions
Scope: Flash Point
Level: Subfunction
Intention in Context: The intention of the Players is to Place their starting firefighter and vehicle positions.
Primary Actor: Player
Secondary Actors: Other players (teammates).
Main Success Scenario:
6. System places initial game objects (fire, hotspot, etc).
7. System informs Player to place their starting position on any of the board spaces outside of the building.
8. Player informs System the position he wishes to place.
9. System informs players to vote for vehicles (ambulance, fire engine) initial positions.
10. System informs Player of new game states.

Extension:
3a. If the position Player chooses is not valid *(Occupied, Wrong Scope, etc.)*
    3a.1 System informs Player of invalid placement.
    3a.2 System informs Player to place a firefighter on a required position.

Communication

Use case: Communication

Scope: Flash point
Level: subfunction
Intention in context: Player to communicate with his teammates
Primary actor: Player
Secondary actors: Other players (Teammates)
Multiplicity: several users can communicate simultaneously
Main success scenario:
1. Current player performs one of the following:
- A text communication,
- A voice communication.
2. System publishes the message to all players

Text Communication

Use case: Text Communication
Scope: Flash point
Level: subfunction
Intention in context: Player to communicate with his teammates in text message
Primary actor: Player
Main success scenario:
1. Player enters the text message
2. System receives the message

Extensions:
1a. Player enters empty string, use case ends.
1b. Player enters mature text, the text is replaced by "*" (extra feature)

Voice Communication

Use case: Voice Communication
Scope: Flash point
Level: subfunction
Intention in context: Player to communicate with his teammates in voice.
Primary actor: Player
Main success scenario:
1. Player speaks to the microphone.
2. System receives the voice

Extensions:
1a. If no recording device is detected, use case ends

Take-Turn

Use Case: Take-Turn
Scope: Flash Point
Level: Subfunction

Intention in Context: The intention of the player is to play his turn.
Primary Actor: *Player*
Secondary Actor: Other player (player's team mates).
Main Success Scenario:

1. System informs the *Player* that it is his turn.
2. The player takes action.
   Step 2 can be repeated as many times as the player wants.
3. System informs player the updated game state after the player's action.
4. Current *Player* informs the system to ends his/her turn.
5. Use case continues at Turn Switch.

Extension:
2a. If the turn time is up, then step 5 is executed regardless of the player's remaining action points.
2b. If the player chooses to Pass a Turn (saving his/her action point), step 5 is executed.
2c. If after taking action the game is over, the system pause/terminates the game and informs players if they won/lose.

Pass a Turn

Use Case: Pass a Turn
Scope: Flash Point
Level: Subfunction
Intention in Context: Player wants to pass a turn
Primary Actor: Player
Main Success Scenario:

1. Player informs System that he/she wants to pass a turn.
2. Use case continues at Turn Switch.

Take Action

Use Case: Take Action
Scope: Flash Point
Level: Subfunction
Intention in Context: The intention of the player is to perform action to cooperate with his/her team mates.
Primary Actor: *Player*
Secondary Actor: Other player (player's team mates), *Game Manager*.
Main Success Scenario:

1. *Player* informs the *Game Manager* to perform the following action.
   - The player chooses to move to an adjacent position (data: the position to move to).

- The player chops a block of wall (data: which block of wall the player wants to chop).
- The player opens/closes a door (data: which door the player wants to open/close).
- The player extinguishes smoke/fire on a position (data: which fire the player wants to extinguish).
- The player carries/drops a victim/hazmat (data: which item the player wants to carry up).
- The player uses his/her special ability (data: the target to use the special ability on if any). (This only exists in experienced mode).
- The player uses the fire engine extinguish fire.
- The player drives the ambulance/fire engine (data: player to travel with and destination location).
- The player wants to Switch Role.

2. *Game Manager* updates the player's status and game status.
3. System informs players the updated *Player* status.
4. System informs players the updated game state.

Extension:

1a. *Game Manager* denied the request of performing the action.
    1a.1. System informs the *Player* about reason of failure.
    1a.2. The action is omitted; the use case ends in failure.

2a. If any victim is rescued after the action
    2a.1. *Game Manager* updates the list of POI.
    2a.2. System informs players the updated POI status.

(1-4)a. If the time runs up during any steps, the taken action is performed before turn goes to next player; use case continues on take turn of another player until all steps are done.

Move

Use Case: Move
Scope: Flash Point
Level: Subfunction
Intention in Context: The intension of the Player is to move to a new position
Primary Actor: Player
Secondary Actor: *Game Manager*
Main Success Scenario:

1. Player informs System that he wishes to move to a new position.
2. System highlights all reachable positions/grids according to the *Player's* current weight.
3. Player chooses a position.
4. System determines the cheapest path to the destination.
5. System updates the process of walking.
6. *Game Manager* determines if there is a POI need to be revealed.

Extensions:
3a. Player chooses an unreachable position, System warns the player. Use case ends in failure.
3b. Player informs System that he determines not to move. Use case ends in failure.
6a. If the POI is false alarm, *Game Manager* eliminates the POI and updates list of POI.

Extinguish fire

User case: Extinguish fire
Scope: Flash Point
Level: Subfunction
Intention in Context: *Player* wants to extinguish fire or smoke in a specific place.
Primary Actor: *Player*
Secondary Actor: *Game Manager*
Main Success Scenario:
1. The *Player* informs the system the position of fire/smoke to extinguish.
2. System requests *Player* to choose the final state after extinguishing.
3. *Game Manager* determines if the player's action point is enough to extinguish.
4. System extinguishes the fire/smoke on the specified position to the desired state.

Extension:
1a. The *Player* specified location is not valid to extinguish fire, use case ends in failure.
3a. If the *Player's* action point is not enough, use case ends in failure.

Carry

Use Case: Carry
Scope: Flash Point
Level: Subfunction
Intention in context: The intention of the player is to Carry the victim/hazmat.
Primary Actor: *Player*
Main Success Scenario:
1. *Player* informs system that he/she wants to Carry a victim/hazmat.
2. System detects if there is reachable victim/hazmat.
3. System asks player to choose which object to carry.
4. *Player* informs the system of their choice.
5. System updates the *Player's* weight according to the victim/hazmat's weight.

Extension:
1a. If the player is already carrying a victim/hazmat, use case ends in failure.
2a. If none, use case ends in failure.

Lead

Use Case: Lead
Scope: Flash Point
Level: Subfunction
Intention in context: The intention of the player is to Lead the victim.
Primary Actor: *Player*
Main Success Scenario:
1. *Player* informs system that he/she wants to Lead a victim.
2. System detects if there is reachable victim/hazmat.
3. System asks player to choose which object to lead.
4. *Player* informs the system of their choice.

Extension:
1a. If the player is already leading a victim use case ends in failure.
2a. If none, use case ends in failure.

Abandon

Use Case: Abandon
Scope: Flash Point
Level: Subfunction
Intention in context: The intention of the player is to abandon the victim/hazmat.
Primary Actor: *Player*
Main Success Scenario:
1. Player informs the system to abandon a victim/hazmat to the player current location.
2. System removes the victim/hazmat from the player's inventory.

Extension:
1a. If the player has no objects in inventory, use case ends in failure.
2a. If the victim/hazmat was carried by the *Player*, system updates the *Player's* weight.

Drive Fire Engine

Use Case: Drive Fire Engine
Scope: Flash Point
Level: Subfunction
Intention in context: The intention of the player is to drive the fire engine to desired location with/without other players riding it.
Primary Actor: *Player*
Secondary Actor: other players (team mates), *Game Manager*
Main Success Scenario:
1. Player informs the system where to drive the fire engine to.
2. *Game Manager* detects if the *Player* has enough AP to drive the fire engine to specified location.
3. System asks the players on parking position of the fire engine if they want to ride.

4. System moves the fire engine and the players riding it to the desired position.

Extension:

1a. If the fire engine is placed in wrong place, use case ends in failure.
1b. If the *Player* is not at the parking spot of the fire engine, use case ends in failure.
2a. If the *Player* choose to drive the fire engine to opposite side, 4 AP is required.
2b. If the *Player* does not have enough AP, use case ends in failure.

Radio Ambulance

Use Case: Radio Ambulance
Scope: Flash Point
Level: Subfunction
Intention in context: The intention of the player is to radio the ambulance to specific destination with/without other players riding.
Primary Actor: *Player*
Secondary Actor: Other players (team mates), *Game Manager*.
Main Success Scenario:

1. *Player* informs the system where to radio the ambulance to.
2. *Game Manager* informs if the *Player* has enough AP to radio the ambulance to specified location.
3. System asks the players on parking position of the ambulance if they want to ride.
4. System moves the ambulance and players riding it to the specified location.

Extension:

1a. If ambulance is placed in wrong place, use case ends in failure.
2a. If the *Player* choose to radio the ambulance to opposite side, 4 AP is required.
2b. If the *Player* does not have enough AP, use case ends in failure.
4a. If the ambulance is moved to a location that has a victim.
    4a.1. The victim is moved to the rescued list.
    4a.2. *Game Manager* updates the list of POI.
    4a.3. System informs players the updated POI status.

Fire Deck Gun

Use Case: Fire Deck Gun
Scope: Flash Point
Level: Subfunction
Intention in Context: Player uses engine to extinguish fire.
Primary Actor: *Player*
Secondary Actor: *Game Manager*
Main Success Scenario:

1. *Player* informs the system to fire the deck gun.
2. *Game Manager* detects if Player has enough action point.
3. System detects if there are firefighters on the quadrant.

4. System randomly chooses a position to fire the deck gun within the regulated region.
5. System informs Player of updated game states.

Extension:

2a. If the player does not have enough AP use case ends in failure.

3a. If there are firefighters on the quadrant, use case ends in failure.

4a. If the *Player's* role is "Driver" he can reroll one or both dice.

4a.1. System asks the *Player* if he/she wants to reroll one/both dice.

4a.2. *Player* informs the system to either reroll one/both dice or not to reroll.

4a.3. System chooses another location to fire the deck gun according to *Player's* choice.

Switch Role

Use Case: Switch Role
Scope: Flash Point
Level: Subfunction
Intention in Context: The intention of the player is to switch to another role as firefighter to have different special abilities (this use case is only valid for advanced mode of the game).
Primary Actor: *Player*
Secondary Actor: *Game Manager*
Main Success Scenario:

1. Player informs the system to switch to another role.
2. *Game Manager* checks if the player has enough AP to switch role.
3. System presents a list of available roles to switch.
4. Player informs the system which role to switch to.
5. System updates the player's status and special ability.
6. System updates the list of available roles for another switch.

Extension:

1a. The player is not at the position of fire engine, use case ends in failure.

1b. The player has not enough AP to switch his/her role, use case ends in failure.

(1-2)a. The player can cancel role switching any time before informing the system which role to switch to, in this case, no AP will be consumed.

4a. When *Player* changes to a role with extra AP or something like "Movement AP" the *Player's* AP bar is also changed with the respect of the role's descriptions.

Interact Door

Use case: Interact Door
Scope: Flash point
Level: subfunction
Intention in context: Player to interact with door objects.
Primary actor: Player

Secondary Actor: *Game Manager*
Main success scenario:
1. Player chooses to interact with the door object.
2. *Game Manager* checks Player's AP.
3. System flips the open/closed status of the door object.

Extensions:
1a. If the door object is not adjacent to the player, use case ends in failure.
2a. If player's AP is not enough, use case ends in failure.

Destroy Wall

Use case: Destroy Wall
Scope: Flash point
Level: subfunction
Intention in context: Player to deal damage to a wall object.
Primary actor: Player
Secondary Actor: *Game Manager*
Main success scenario:
1. Player chooses to deal damage to a wall object.
2. *Game Manager* checks the remaining AP of the player.
3. System updates the state of the wall.

Extensions:
1a. If the wall object is not adjacent to the player, use case ends in failure.
2a. If player's AP is not enough, use case ends in failure.
3a. If the hit point of the wall is empty after chopping, the wall is destroyed.

Use Special Abilities

Use Case: Use Special Abilities
Scope: Flash Point
Level: Subfunction
Intention in Context: The intention of the player is perform the special ability of the role the player is currently playing. (This use case only exists in experienced mode.)
Primary Actor: *Player*
Secondary Actor: Other player (player's team mates), *Game Manager*.
Main Success Scenario:
1. The *Player* informs the system to perform special ability of the Player's current role. The performable special abilities are the following:
   - Resuscitate
   - Dispose
   - Command
   - Reveal POI
2. *Game Manager* checks if the player has enough action point.
3. System updates the game status after performing the special ability.

4. System updates the player's status after performing the special ability.

Extension:

(1-4)a. If the *Player's* ability is "passive" ability (like Generalist's extra AP, Rescue Specialist's extra Movement AP, less Chopping AP consumption and extra AP consumption to extinguish fire, etc.) The *Player* cannot actively perform it but the *Player* will have special properties (extra/less AP to perform actions, extra special AP bar, etc).

Resuscitate_Paramedic

Use case: Resuscitate_Paramedic
Scope: Flash point
Level: subfunction
Intention in context: Player to treat a victim.
Primary actor: Player
Secondary actor: GameManager
Main success scenario:

1. Player informs the System that she/he wants to treat the victim.
2. *Game Manager* checks Player's AP.
3. System sets the weight of the victim to 0.
4. System updates the status of the game.

Extensions:

1a. If the victim object is not at the same location as the player, use case ends in failure.
1b. If the Player decides to not treat the victim, use case ends.
2a. If the *Player* does not have enough AP, use case ends in failure.

Remove Hazmat_Hazmat Technician

Use Case: Remove Hazmat_Hazmat Technician
Scope: Flash Point
Level: Subfunction
Intention in Context: Remove a hazmat from the Firefighter's space
Primary Actor: *Player*
Secondary Actor: *Game Manger*
Main Success Scenario:

1. Player selects a Hazmat to remove.
2. *Game Manager* checks if the player has enough action point to remove the hazmat.
3. System removes the Hazmat.
4. System updates the game states

Extension:

1a. If the hazmat is not at the *Player's* position, use case ends in failure.
2a. if the player does not have enough AP, use case ends in failure.

Command_Fire Captain

Use Case: Command_fireCaptain
Scope: Flash Point
Level: Subfunction
Intention of context: the intention of the player is to command specific firefighter to do specific actions.
Primary actor: *Player*
Secondary actor: Other players, *Game Manager*
Main Success Scenario:

1. Player informs system to command other firefighters.
2. System asks Player to choose the firefighter to be commanded.
3. System shows the list of valid actions.
4. Player chooses one action from the list.
5. Game manager determines if there is enough command AP to finish this order.
6. System asks the commanded *Player* if they are willing to perform the action.
7. The commanded firefighter completes the action.

Extension:
1a. If the current *Player* is the only player in the game, use case ends in failure.
2a. If more than 1 command AP is spent on the CAFS firefighter in this turn, use case ends in failure.
5a. If there is not enough AP, use case ends in failure.
6a. If the commanded *Player* refused to do the action, use case ends in failure.

Reveal POI_Imaging Technician

Use case: Reveal POI_Imaging Technician
Scope: Flash Point
Level: Subfunction
Intention in Context: The *Player's* intention is to reveal POI directly.
Primary Actor: *Player*
Secondary Actor: *Game manager*
Main successful scenario:

1. The *Player* informs the system to reveal a POI
2. The *Player* chooses which POI he wants to reveal.
3. Game Manager determines if the player have enough AP to reveal
4. System reveal a POI

Extension:
3a. If the player doesn't have enough AP, user case fail
4a. If the POI is a false alarm, System eliminates the POI and updates POI list

Advance Fire

Use Case: Advance Fire
Scope: Flash Point
Level: Subfunction
Intention in Context: The system will proceed the game by adding more fire.
Primary Actor: *Game Manager*
Main Success Scenario:

1. The *Game Manager* randomly picks a position on map to advance fire. The following are the possible cases of advancing fire.
   - Advance fire on an empty tile adds a smoke to the space.
   - Advance fire on a smoke causes the smoke turn to fire.
   - Smoke adjacent to fire will be lighted up to fire.
   - An explosion occurs.
2. The *Game Manager* detects if any *Player* is affected by the advanced fire
3. The *Game Manager* detects if any POI is affected by the advanced fire and remove the POI.
4. The *Game Manager* detects the damage level of the house.

Extension:

1a. If the position of advanced fire has a hotspot, repeat step 1.

1b. If the position of advanced fire contains hazmat and fire, a new hotspot is placed; use case continues at step 2.

1c. When fire advanced on multiple hazmat.

1c.1. The *Game Manager* informs *Player* to choose order of hazmat explosion.

1c.2. The *Player* informs the system of their choice of order.

1c.3. *Game Manager* performs explosions in *Player* desired order; use case continues at step 2.

1c.4. *Game Manager* performs explosions in random order if the player does not pick order of explosion on time; use case continues at step2.

1d. If the explosion happens near walls/doors.

1d.1. The wall/door is destroyed by the system.

1d.2. The *Game Manager* increases damage level of house; use case continues at step 2.

1e. If the advance fire causes a shock wave.

1e.1. The *Game Manager* follows the shock wave to an empty space/smoke/door/wall.

1e.2. The *Game Manager either adds a fire to the empty space/smoke or adds a damage to the wall/door.*

1f. If the advance fire causes the house to collapse, the *Game Manager* informs players if they won/lose.

2a. Player(s) is on a location with fire on.

2a.1. The *Game Manager* asks the player to choose a valid location to respawn.

2a.2. The *Game Manager* respawns the player on the location they choose/ on a random valid location if the player does not choose location to respawn in a given time; use case continues at step 3.

3a. Removing POI will result in failure to rescue victims.

3a.1. *Game Manager* detects a victim involved in the fire and removing the victim will fail the rescue operation.

3a.2. *Game Manager* informs players they lost the game; use case ends in success (game over).

4a. The house is destroyed due to too much damage taken.

4a.1. *Game Manager* detects the level of damage is above maximum.

4a.2. *Game Manager* informs players if they won/lost the game; use case ends in success (game over).

Replenish POI

Use Case: Replenish POI
Scope: Flash Point
Level: Subfunction
Intention in Context: The *POI Manager* will proceed the game by replenishing POI.
Primary Actor: *Game Manager*
Main Success Scenario:

1. *Game Manager* randomly generates a position for placing the POI.
2. *Game Manager* places the POI to the location.
3. System informs all players the updated map and POI status (list of POI not yet put into the map).

Extension:

1a. If the randomly chosen position has fire, the POI is placed to a new location without fire following the trace on the map.

2a. The POI is placed on the same location of a *Player*.

2a.1. The POI is immediately revealed to players by the system.

2a.2. *Game Manager* discards the POI immediately if the POI is a false alarm and repeat step 1.

2a.3. *Game Manager* updates the list of POI.

2b. There is no more POI to be placed.

2a.1. *Game Manager* checks if there are POI that are not yet placed onto the map.

2a.2. When no remaining POI detected, *Game Manager* checks for existing POI on map.

2a.3. When no POI on map, *Game Manager* informs system to end game.

2a.4. System presents all players if they win/lose.