

ClanBomber Game Logic

Franz Xaver Steinbacher

18.12.2018

UNIVERSITY OF APPLIED SCIENCES UPPER AUSTRIA
CAMPUS HAGENBERG
DEPARTMENT EMBEDDED SYSTEMS DESIGN
REVISION: 1.0

1 Revision History

REVISION	DATE	AUTHOR(S)	DESCRIPTION
0.1	17.12.2018	Franz Xaver Steinbacher	First draft of the document.
1.0	18.12.2018	Franz Xaver Steinbacher	First release of the document.

2 Basic

2.1 Abstract

The following document will describe the game logic of the Bomberman clone ClanBomber.

2.2 Game States

The Game has following states:

- PRE-GAME
- IN-GAME
- POST-GAME

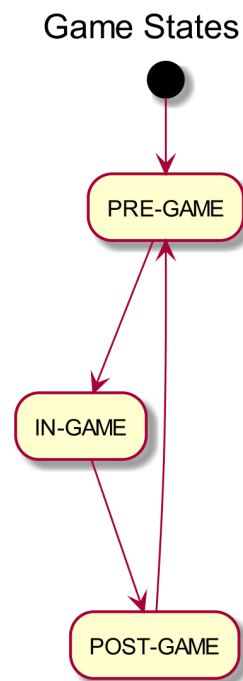


Figure 1: Game States

The State Diagram shows the three different game states. After the start up the game enters the pre-game state. Now the server waits until enough players are connected, then the server enters the in-game state.

In the in-game state the server runs the game logic and sends the information to the clients. When the game is over the server enters the post-game state and sends the ranking to the clients. Then the server goes back to the pre-game state and starts a new game iteration.

2.3 Game Sequence

The Sequence Chart shows the communication between the main classes inside the server.

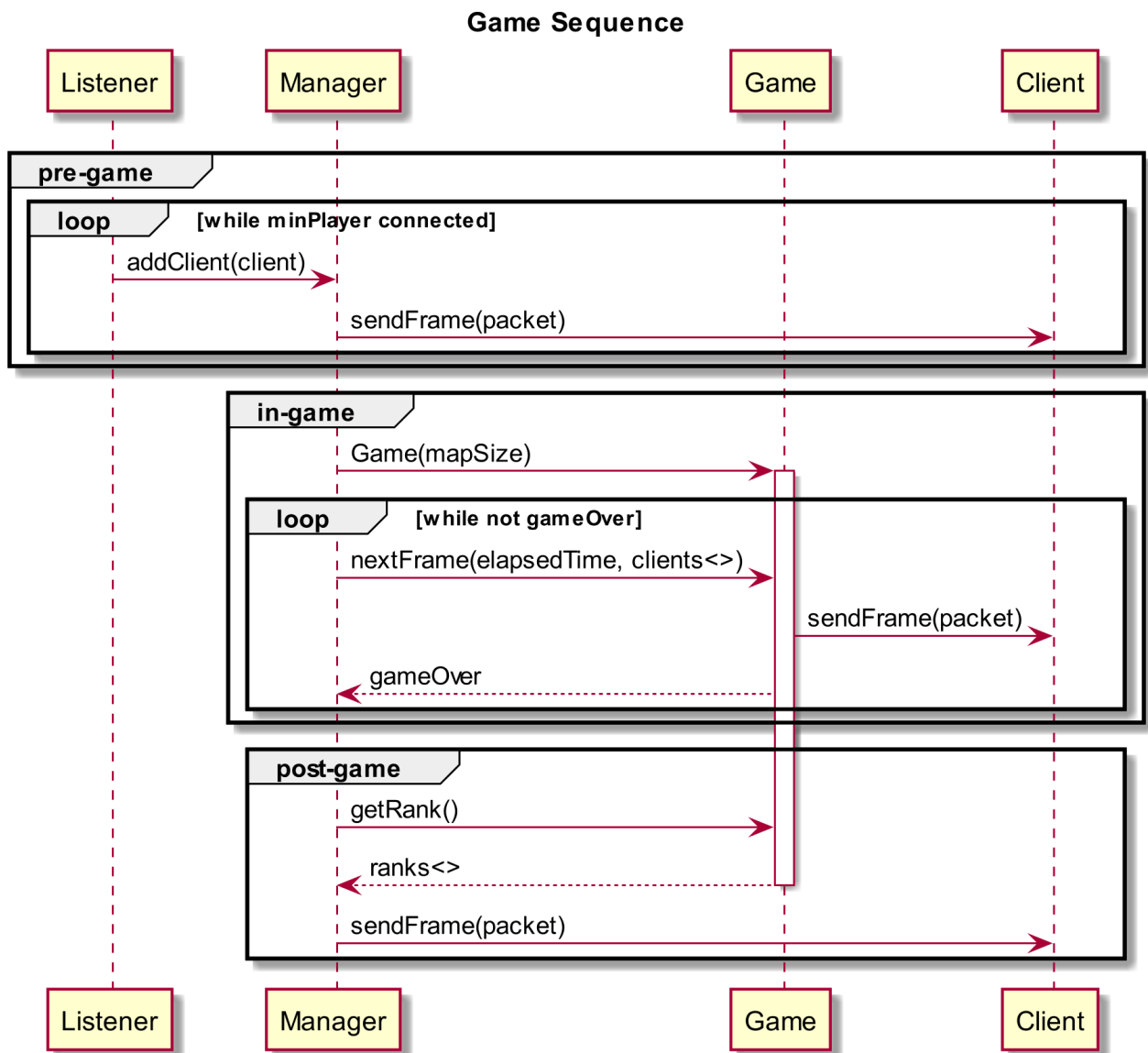


Figure 2: Game Sequence

The Manager Class is the main part of the server, it manage the game states, the connection and disconnection of clients. When enough player have connected the manager create a object of the class Game.

Class Game represents the Game Logic, with collision detection, map, player management and the win condition.

The Client Class represents the Connection to the client, with the interfaces getInput and sendFrame the logic can access the clients.

2.4 Game Logic

The main Game Logic is hosted in the nextFrame function. The function is called with the elapsed time till the last call and a vector with the connected clients.

With the elapsed time the logic can calculate the movement and the time when a bomb explodes.

NextFrame operates the following steps:

1. If a clients connect during the game, add the client to the game with 0 lives.
2. Remove all disconnected players.
3. Calculate the flame timeout, if the flame timeout is lower than 0, remove the flame.
4. Check if a bomb is inside a flame, if yes the bomb explodes.
5. Calculate the bomb timeout, if the timeout is lower than 0, let the bomb explode.
6. Create the flames for the exploding bombs.
7. Reduce the immortal time of the players.
8. Check if a player is inside a flame, if yes check if the player is immortal, if no decrement the live count.
9. Check if a player is dead, if yes add the player to the dead list.
10. Check the input and move the players and set the bombs.
11. Check if a player picks up a item.
12. Send the result to the clients.
13. Check if the game is finished.
14. Return the game state(true if game is over, else false).

2.4.1 Map

The Map is a square with a minimum of 8x8 and four different type of fields:

- Empty: a field where the player can move and place bombs.
- Fixed: a undestroyable block that can't be destroyed or the player can't move or place bombs at this field.

- Breakable: a destroyable block, which can be destroyed with a bomb.
- Item more bombs: the player which moves first to this field get a additional bomb.

Field	Move	Place Bomb	Destroy
Empty	Yes	Yes	No
Fixed	No	No	No
Breakable	No	No	Yes
Item	Yes	Yes	Yes

For the detail map definition see the Network Protocol Specification.

2.4.2 Player

A player has the following variables:

- ID: the unique player id
- Bombs: number of bombs the player can place
- Lives: number of remaining lives, if 0 the player is dead
- Direction: the moving direction of the player
- Position X: the x coordinate as float
- Position Y: the y coordinate as float
- ImmortalTime: the remaining time span in which the player is immortal and can't be killed

To kill a player, the player have to touch a flame. When this happens the player lose a life and his immortal time is set a defined timeout. While this time the player can't be killed.

If a player is killed and he has no remaining lives, he can't do anything more. He has to wait until the game ends.

2.4.3 Movement

The movement of a player is calculated with the inputs from the client and the elapsed time till the last call.

The movement constant is defined with 4 fields per second.

A player can operate following move commands:

- up
- down
- left
- right
- idle

When the command idle is received the logic does nothing and the player remains at his position.

When a moving command (up, down, left or right) is received the logic checks if the player can move in this direction. For this purpose the logic has to check the neighboring fields. If the field is empty or a item the player can move to this field. If the field is a block the player only can move to the middle of the current field.

When a place bomb command is received from the client the logic checks the ticking bombs of the player on the field. If the player has bombs left he can place the bomb. A player can only place one bomb at a field.

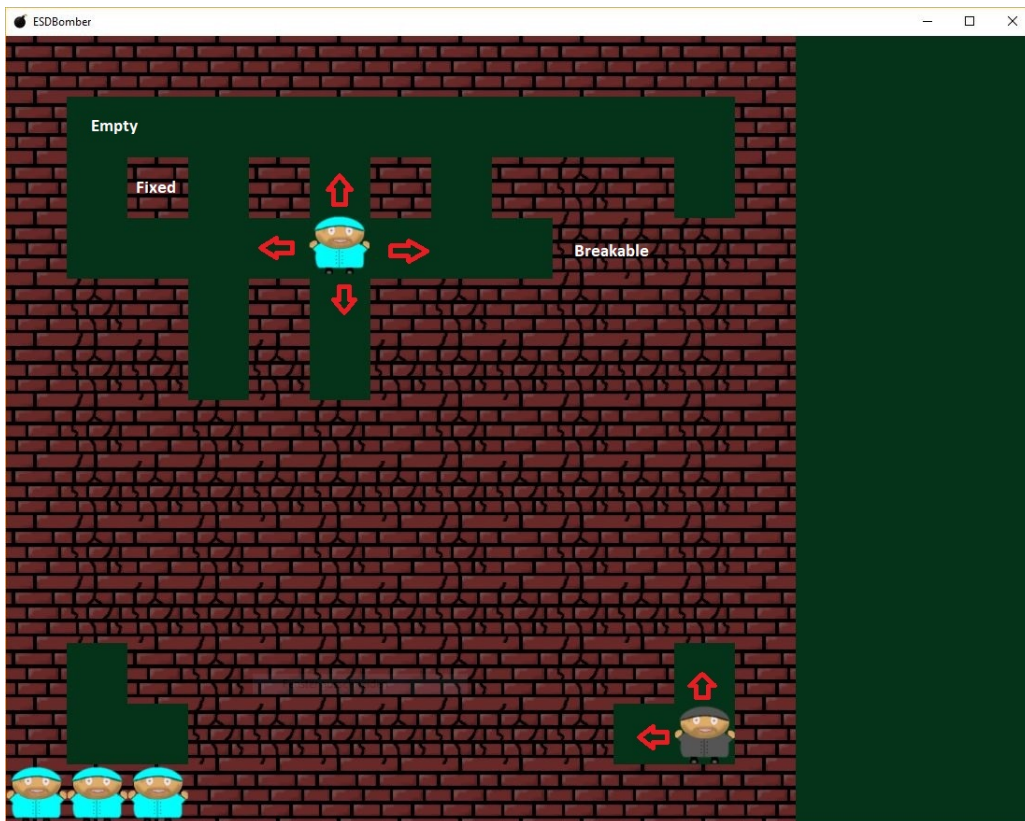


Figure 3: Move Logic

The image shows the movement option for two different players.

2.4.4 Bomb

A bomb has the following variables:

- ID: the player id
- Position X: the x coordinate as float
- Position Y: the y coordinate as float
- Timeout: the remaining time until the bomb explodes

When the timeout of a bomb is less than 0 or the bomb is hit by a flame the bomb explodes and is removed from the bomb vector.

2.4.5 Flame

A flame has the following variables:

- ID: the player id
- Position X: the x coordinate as float
- Position Y: the y coordinate as float
- Timeout: the remaining time until the flame is removed

When the timeout of a flame is less than 0 the flame is removed from the flame vector.

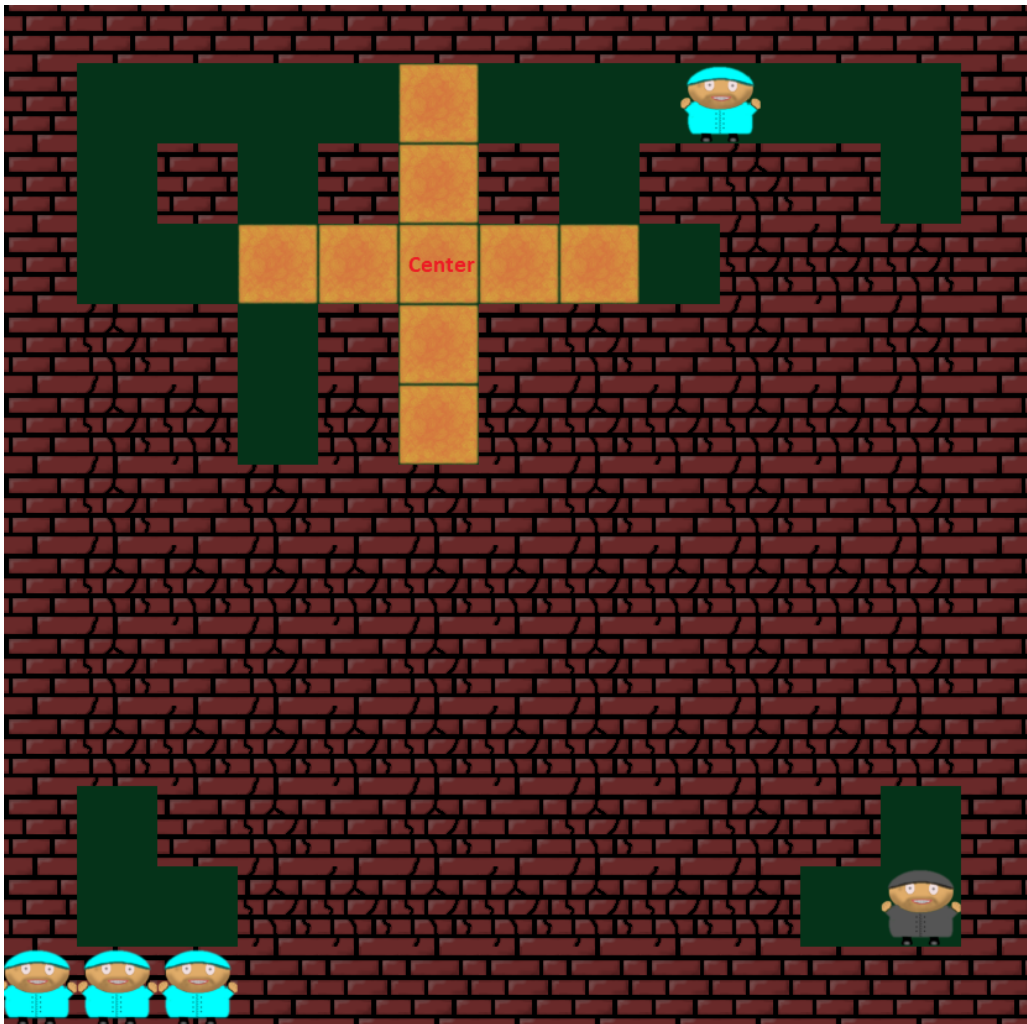


Figure 4: Flame Logic

The picture shows the spreading of a flame. With the center the flame has a maximum spread of 3 fields.

Each time when a bomb explodes the flames for the bomb are calculated. If a flame hits a destructible block the block is destroyed and the flame is blocked at this position.

2.4.6 Item

When a destroyable block is destroyed there is the chance that randomly a item spawns at the position of the block.

If a item is hit by a flame the item is destroyed. If a player moves to the position of the item he will get the item and it will be removed from the map.

The more bombs item, adds a additional bomb to the player, who picks the item up.

There is no limit how much items a player can pick up.