

# Tank Documentation

## Member

1. Ditthapong	Lakagul	6310545965
2. Tawan	Boonma	6310545272

## Use Case

UC1: Player chooses map.

UC2: User chooses mode.

UC3: Tank moves.

UC4: Tank fire.

UC5: Tank destroys other tank.

**Use Case:** UC1: Player chooses map.

**Primary Actor & Goal:** Map has been chosen.

**Pre-condition:** Game window has started.

**Post-condition:** Map has been chosen.

**Main Success Scenario:**

1. Player opens the game window.
2. System display game window.
3. Player clicks to choose one of three maps.
4. System set up map that has been chosen.

**Use Case:** UC2: User chooses mode.

**Primary Actor & Goal:** Mode has been chosen.

**Pre-condition:** Game window has started.

**Post-condition:** Mode has been chosen.

**Main Success Scenario:**

1. Player opens the game window.
2. System display game window.
3. Player clicks to choose one of two modes.
4. System set up mode that has been chosen.

**Use Case:** UC3: Tank moves.

**Primary Actor & Goal:** Tank moving.

**Pre-condition:** Game map and mode has been chosen.

**Post-condition:** Tank moving.

**Main Success Scenario:**

1. User presses on the key.
2. System displays the next move of the tank.
3. User controls the tank to move in the bush.
4. System displays the invisible tank.
5. User try move through the brick and steel
6. System displays the tank which cannot move through it.

**Use Case:** UC4: Tank fire.

**Primary Actor & Goal:** Tank fire bullet.

**Pre-condition:** Game map and mode has been chosen.

**Post-condition:** Tank fire bullet.

**Main Success Scenario:**

1. User presses on the key.
2. System displays the bullet.
3. User fire bullet through brick
4. System displays the broken brick.
5. User tries to fire through the steel.
6. System displays steel that hasn't broken.

**Use Case:** UC5: Tank destroys other tank.

**Primary Actor & Goal:** Tank has been destroyed.

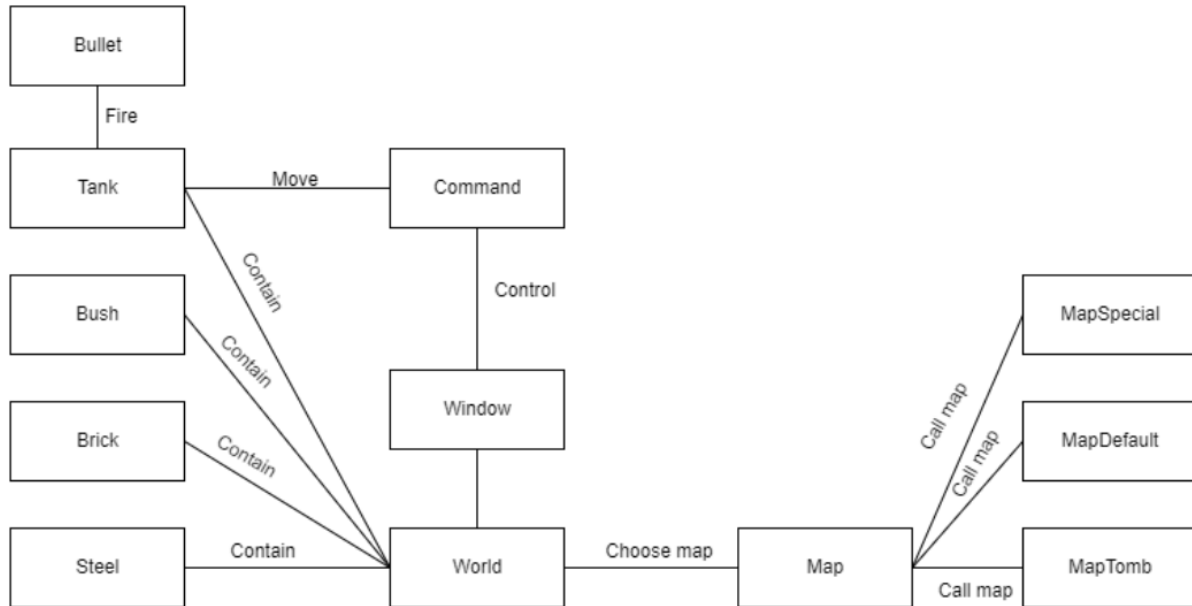
**Pre-condition:** Tank has been fired.

**Post-condition:** Tank has been destroyed.

**Main Success Scenario:**

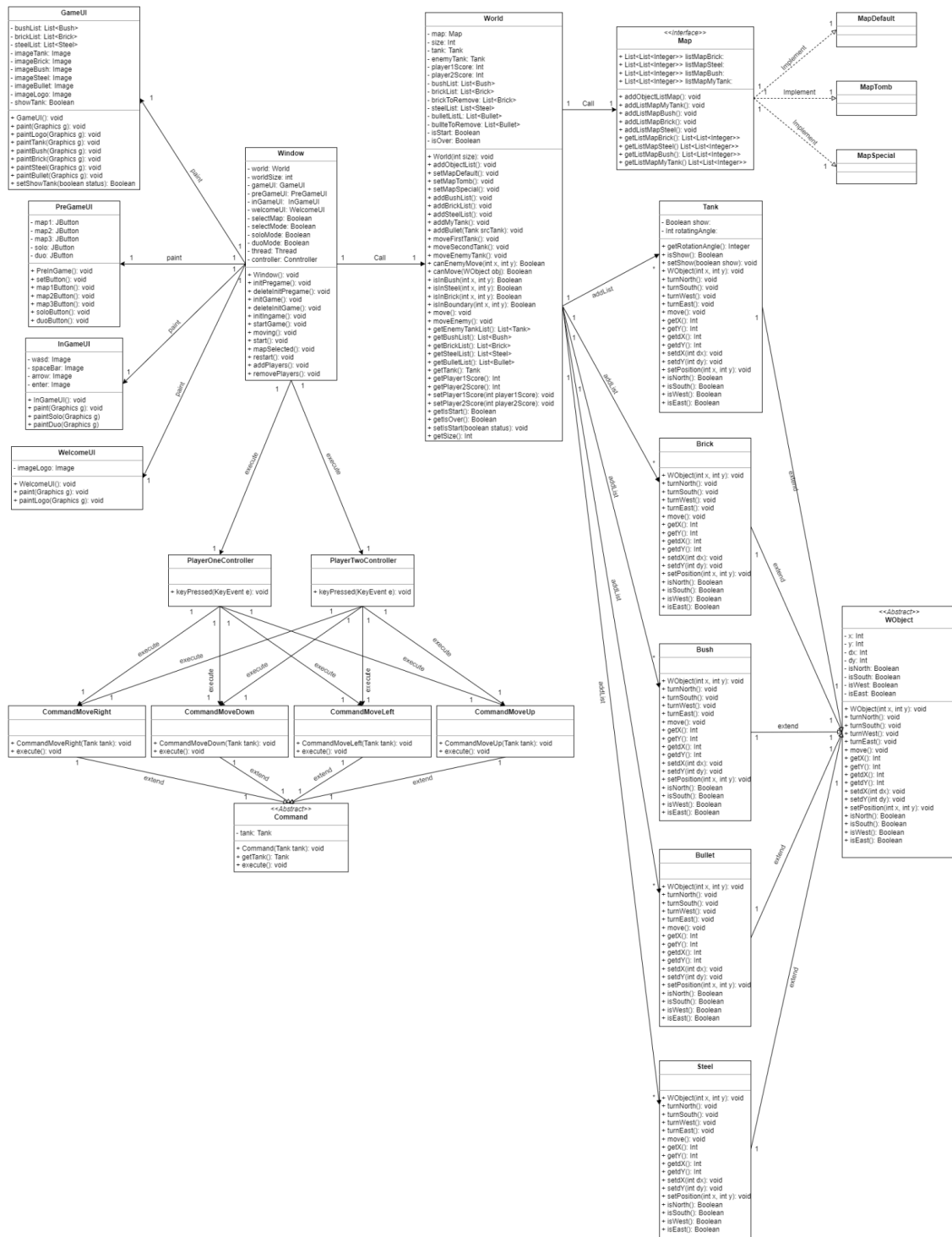
1. User presses on the key.
2. System displays the bullet.
3. Bullet hit tank.
4. System display destroyed tank.

# Domain Model Diagram



[https://drive.google.com/file/d/1Xo5mv4pR4sel\\_g-9QnjbOWDu4w4Jj3l3/view?usp=sharing](https://drive.google.com/file/d/1Xo5mv4pR4sel_g-9QnjbOWDu4w4Jj3l3/view?usp=sharing)

# UML Class Diagram



Note: Please take a look at this link if you want to see Class diagrams clearly.

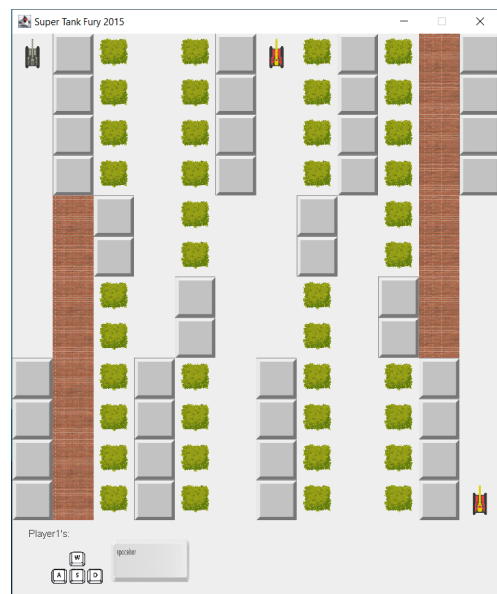
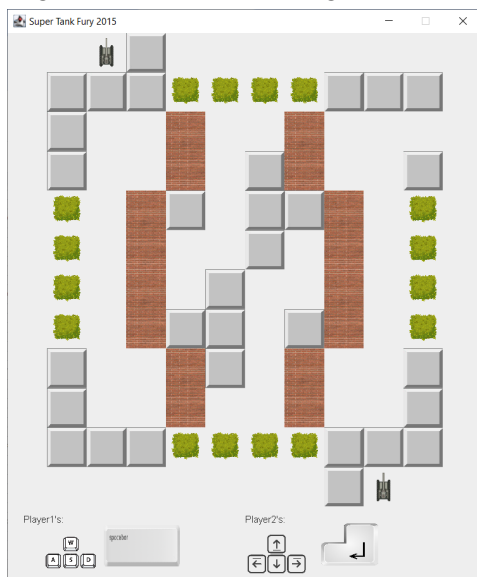
<https://drive.google.com/file/d/1AX1E3zxaW-tABDwwZLEym7iLS4tQGTII/view?usp=sharing>

# Creativity

- Map selection: consist of 3 maps for the user to select.



- In game interface: when game starts.



- Restart the game when the game finishes(Player wins or loses).  
(Try in our game)

# Supporting documents describing knowledge used in the project

## Design pattern

1. **Command design pattern:** This pattern has been used in the control part.
2. **Flyweight design pattern:** This pattern has been used in the image part.
3. **Adapter design pattern:** This pattern has been used in map selection.

## GRASP principle

1. **Polymorphism:** Create a stable interface for choosing maps.
2. **Protected variable:** Support changing in every class.
3. **Low coupling:** Every class doesn't depend on each other.
4. **High cohesion:** Every class and method has only one job and purpose.
5. **Indirect:** Create World.java another class to use for supporting low coupling and high cohesion .