

# 객체지향 설계패턴

## *F Team 2nd Iteration*

• 01 분반    • 김기석 교수님    • Due Day: 2019 년 5 월 13 일 까지



21500375 신지은  
21500643 정선일  
21700034 곽영혜  
21700398 신유진

## 1. Function List to be Finally Implemented

Function	Implementation
<i>First Setting:</i> Team Name, Time per Quarter, The number of Quarter	1st Iteration
<i>Total Timer:</i> Display the timer of Total Play	1st Iteration
<i>Add Score:</i> If the user pushes the button of score, the score increases	2nd Iteration
<i>Display Score:</i> Display each team's score	2nd Iteration
<i>Display Attack:</i> Show the who has to attack	2nd Iteration
<i>Attack Timer:</i> Display the Attack Timer (24 or 14 seconds)	2nd Iteration
<i>Count Penalty:</i> The user can record the Penalty count	3rd Iteration
<i>Change the Court:</i> When the Quarter finishes, the location of score board is changed	3rd Iteration
<i>Break Timer:</i> When the Quarter finishes, only the break timer starts	Final Iteration
<i>Strategy Timer:</i> If one team want Strategy time, only the Strategy timer starts.	Final Iteration
<i>Load / Save the data:</i> Load or Save the data of play result and show them to the user	Final Iteration

## 2. List of Implemented Functions in 1st Iteration

Function	Implementation
<i>First Setting:</i> Team Name, Time per Quarter, The number of Quarter	1st Iteration
<i>Total Timer:</i> Display the timer of Total Play	1st Iteration

## 3. List of Implemented Functions in 2nd Iteration

Function	Implementation
<i>Add Score:</i> If the user pushes the button of score, the score increases	2nd Iteration
<i>Display Score:</i> Display each team's score	2nd Iteration
<i>Display Attack:</i> Show the who has to attack	2nd Iteration
<i>Attack Timer:</i> Display the Attack Timer (24 or 14 seconds)	2nd Iteration

## 4. Brief description about important classes and methods

playGame	firstSetting
+width: int +height: int +nextButton: JButton +p1: JPanel +p2: JPanel +card: CardLayout +fs: firstSetting +mb: mainBoard	+panel: JPanel +TN1: JTextField +TN2: JTextField +TPQ: JTextField +TNQ: JTextField +nextButton: JButton
+playGame() +actionPerformed(ActionEvent): void +main(String[]): void	+firstSetting(int, int) +placeComponents(JPanel, int, int): void +getPanel(): JPanel +getTN1(): JTextField +getTN2(): JTextField +getTPQ(): JTextField +getTNQ(): JTextField +getNextButton(): JButton

**playGame:** 이 프로그램의 메인 함수가 있다. CardLayout 을 통해 firstSetting 프레임에서 mainboard 프레임으로 전환해주는 역할을 한다. nextButton 을 통해서 프레임이 전환될 때 유저가 입력했던 값들을 메소드를 통해 다음 프레임에게 넘겨준다.

**firstSetting:** 경기를 시작하기 전에 필요한 설정을 위한 프레임을 생성할 때에 필요한 각종 Label 과 Button 을 포함하고 있는 하나의 panel 을 만들어 주는 클래스이다.

mainBoard	displayAttack
+width: int +height: int +panel: JPanel +gt: gameTimer +sb: scoreBoard +at: Attack_Timer +da: displayAttack	+width: int +height: int +A: JButton +B: JButton
+mainBoard() +getPanel(): JPanel	+displayAttack() +setTeamName(String, String): void +getTeam1Attack(): JButton +getTeam2Attack(): JButton +actionPerformed(ActionEvent): void

**mainboard:** firstSetting 클래스와 같이 뒤에 있는 displayAttack, gameTimer, Attack\_Timer, scoreBoard 의 클래스를 통해 하나의 panel 을 생성해주는 클래스이다

**displayAttack:** firstSetting 에서 입력을 받았던 팀 이름과 보여주고 동시에 색을 통해 공격 차례를 나타낼 수 있는 Button 을 생성해주는 클래스이다.

gameTimer	Attack_Timer
+width: int +height: int +storedTPQ: int +MaxQN: int +QN: int +minute: int +second: int +millisecond: int +thread: Thread +display: JLabel +quarterLabel: JLabel +start: JButton +stop: JButton +disp: String +on: boolean	+width: int +height: int +counter: int +second: int +millisecond: int +thread: Thread +display: JLabel +start: JButton +stop: JButton +reset: JButton +reset14: JButton +disp: String +on: boolean
+gameTimer() +setMinute(int): void +setQuarterNum(int): void +getQuarterLabel(): JLabel +getLabel(): JLabel +getStartButton(): JButton +getStopButton(): JButton +updateQuarterNum(): void +changeLabel(): void +updateI(): void +run(): void +actionPerformed(ActionEvent): void	+Attack_Timer() +getLabel(): JLabel +getStartButton(): JButton +getStopButton(): JButton +getResetButton(): JButton +getReset14Button(): JButton +reset(): void +reset14(): void +changeLabel(): void +update(): void +run(): void +actionPerformed(ActionEvent): void

**gameTimer:** thread 를 통해 firstSetting 에서 입력 받았던 ‘한 쿼터 당 시간’을 기준으로 countdown timer 를 생성해주는 클래스이다. 추가적으로 firstSetting 에서 입력 받았던 ‘총 쿼터 수’ 만큼 전체 경기 타이머를 reset 해주고 현재 쿼터 수를 보여줄 수 있도록 Label 과 Button 들을 생성해주는 클래스이다.

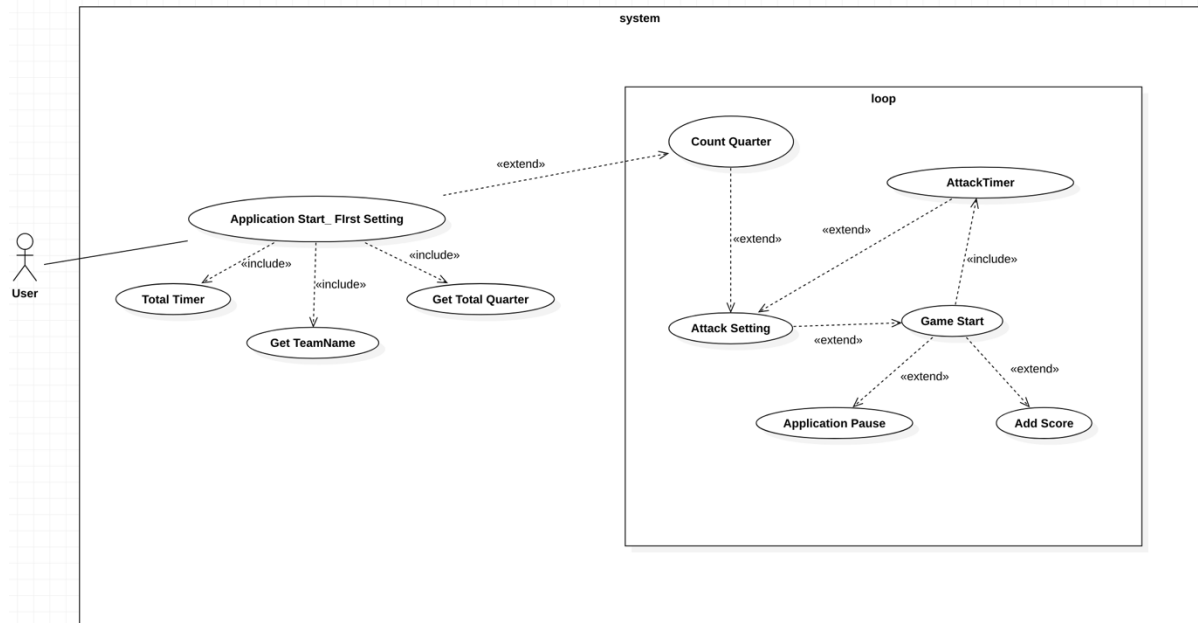
**Attack\_Timer:** gameTimer 와 같은 방법으로 countdown timer 를 생성하나 24 초나 14 초로 timer 를 reset 해주는 Label 과 Button 을 생성해주는 클래스이다.

*# 두 개의 timer 를 살펴보았을 때 많은 부분이 겹치고 특정 부분만 다르므로 Template Method Design Pattern 을 적용할 수 있을 것 같다.*

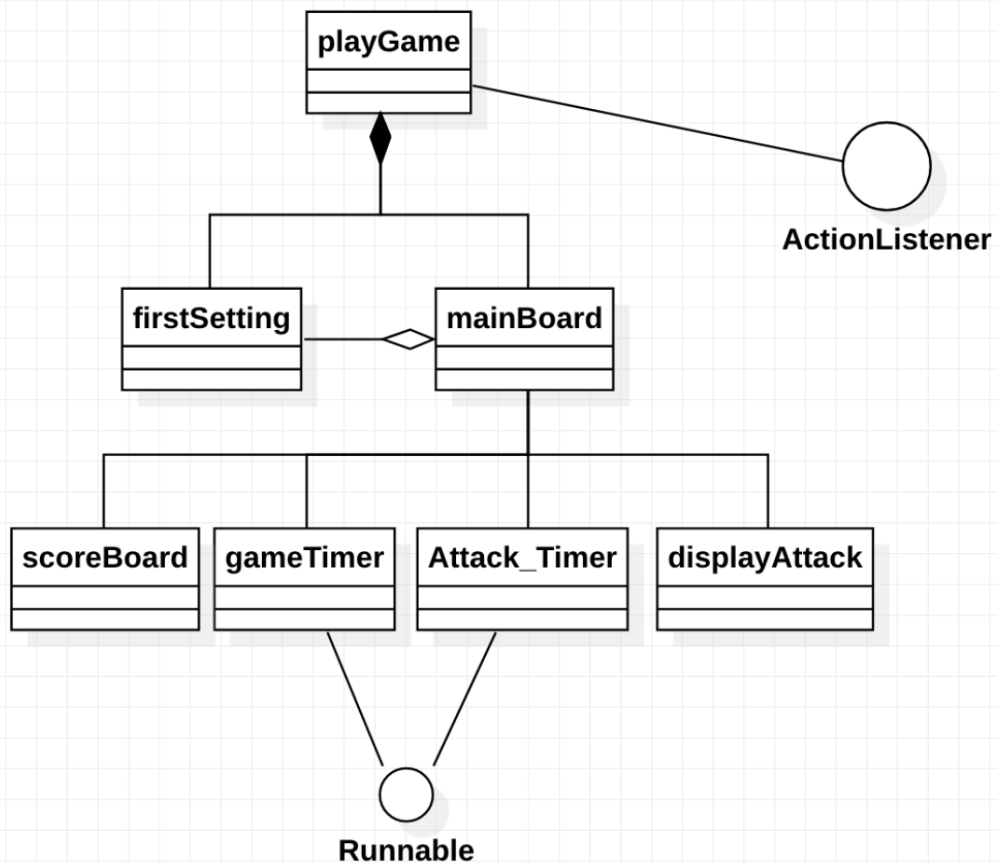
scoreBoard
+width: int +height: int +redScoreAmount: int +blueScoreAmount: int +redScore: JLabel +blueScore: JLabel +redUpButton: JButton +redDownButton: JButton +blueUpButton: JButton +blueDownButton: JButton +resetButton: JButton
+scoreBoard() +getRedScore(): JLabel +getBlueScore(): JLabel +getRedUp(): JButton +getRedDown(): JButton +getBlueUp(): JButton +getBlueDown(): JButton +actionPerformed(ActionEvent): void

scoreboard: 해당하는 팀의 점수와 그 점수를 증가시키거나 감소시킬 수 있는 버튼들을 생성해주는 클래스이다.

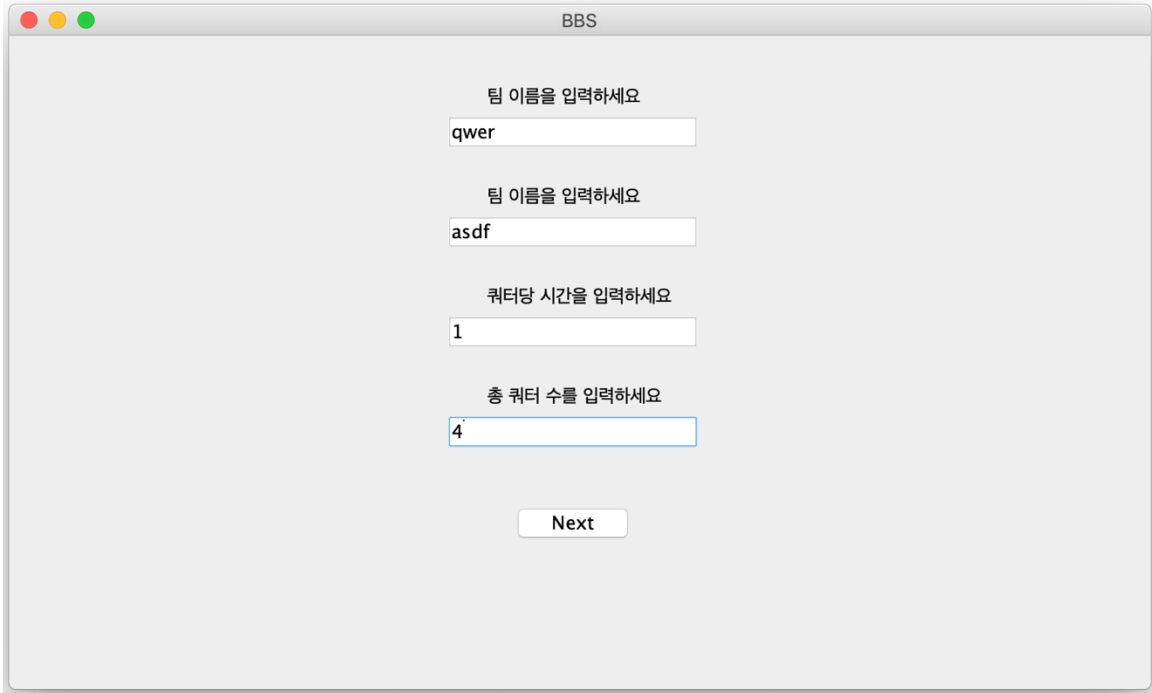
## 5. Use Case Diagram



## 6. Class Diagram



## 7. Main Screen Dump



A screenshot of a web browser window titled "BBS". The page contains four input fields for configuration, each with a label above it. The first two fields are for team names, the third for quarters per game, and the fourth for the total number of quarters. A "Next" button is at the bottom.

팀 이름을 입력하세요  
qwer

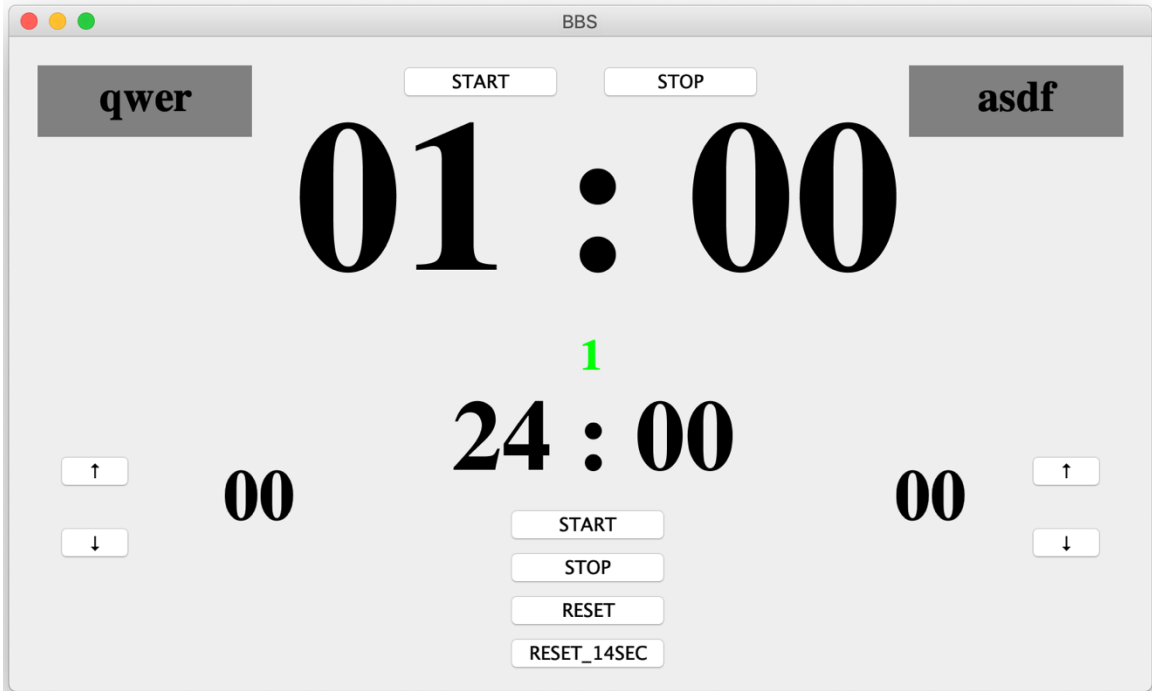
팀 이름을 입력하세요  
asdf

쿼터당 시간을 입력하세요  
1

총 쿼터 수를 입력하세요  
4

Next

팀 이름, 쿼터당 시간과 총 쿼터 수를 입력하는 화면



A screenshot of the "BBS" main screen after configuration. It features a large central timer showing "01 : 00". Above the timer are "START" and "STOP" buttons. To the left and right of the timer are team names "qwer" and "asdf" in grey boxes. Below the main timer is a green number "1" and a smaller timer showing "24 : 00". At the bottom are four buttons: "START", "STOP", "RESET", and "RESET\_14SEC". On the far left and right are vertical controls with up/down arrows and a "00" score display.

qwer

START STOP

01 : 00

1

24 : 00

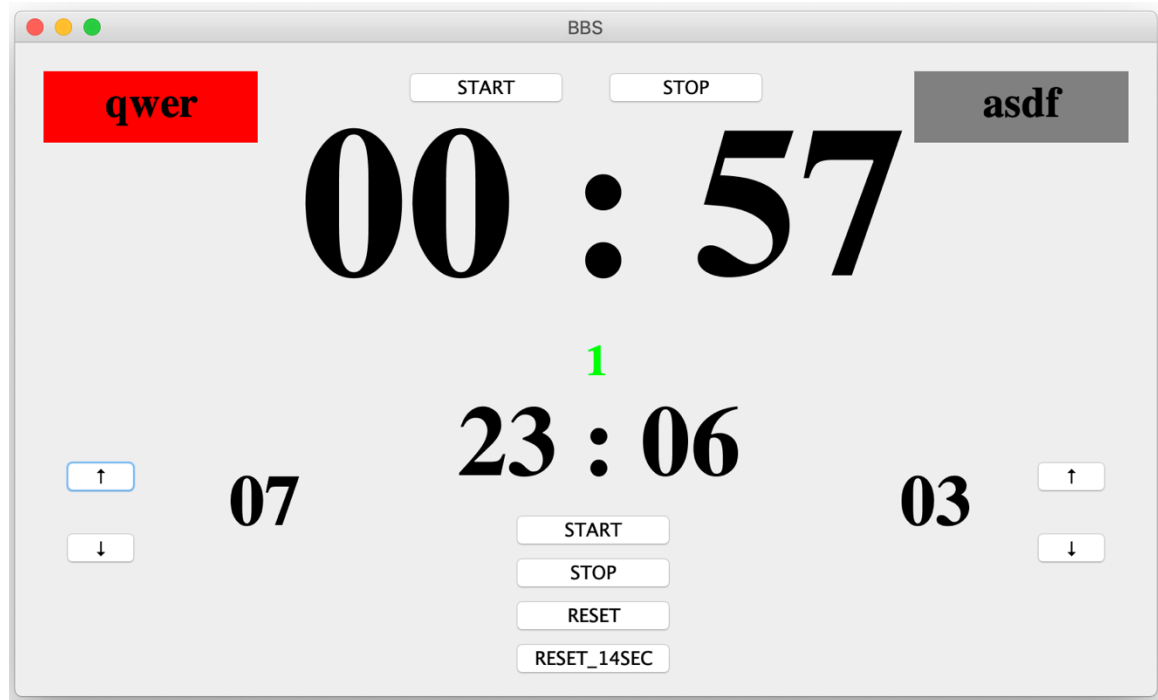
00

START STOP RESET RESET\_14SEC

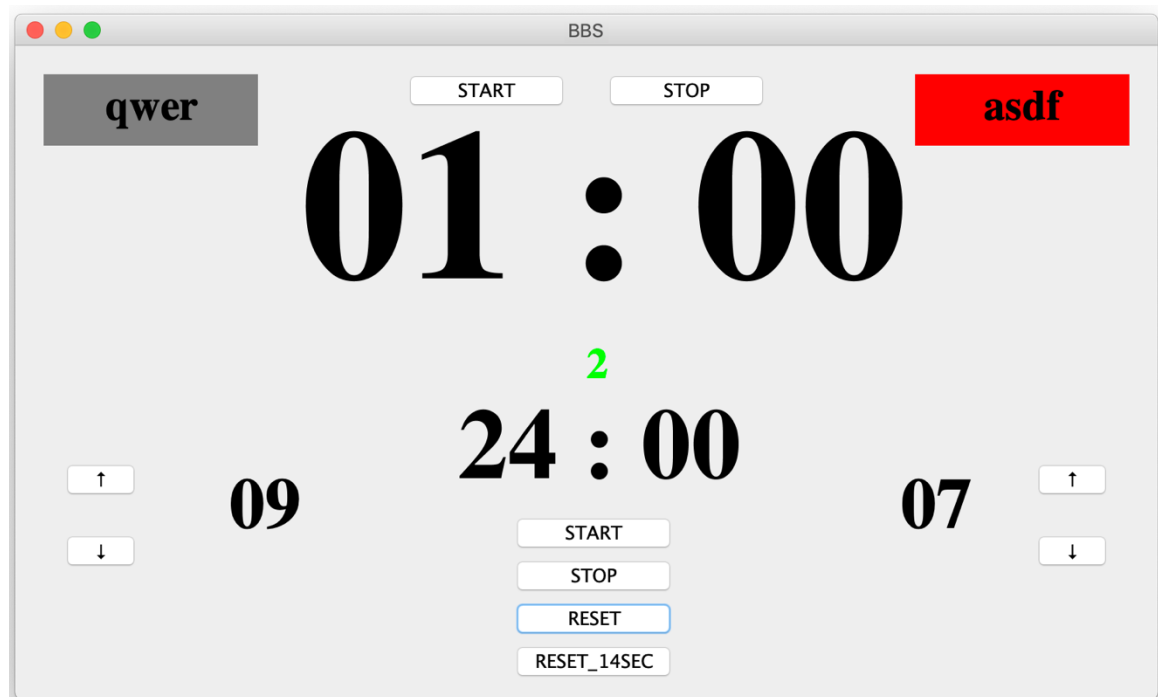
00

첫 번째 세팅을 마치고 Next 버튼을 눌렀을 때의 화면





‘qwer’ 팀의 공격 차례일 때의 화면



전체 경기 타이머가 끝났을 때 현재 쿼터 수를 증가해주면서 초기화 된 화면

## 8. Iteration 별 Refactoring / Design Pattern

Iteration	Sub-Iteration	Refactoring / Design Pattern	적용 / 내용요약 / 기술된 페이지
1st	1	기능 구현	
2nd	2	기능 구현 Refactoring#1 Refactoring#2	DP#1(Template Method) 적용의 가능성 확인 (보고서 p. 4)  Refactoring#1 전체 경기 타이머를 구현하는 방법을 바꿈. (Thread 로 구현)  Refactoring#2 프레임의 연결 방법(CardLayout)
3rd	3	기능 구현 DP#1(Template Method) DP#2(Strategy) DP#3(State)	DP#1(Template Method)를 위한 Interface 정의 DP#2(Template Method)를 위한 Interface 정의 DP#3(State)를 위한 Interface 정의 DP#4(Factory Method) 적용의 가능성 확인
Final	4	기능 구현 DP#4(Factory Method)	

## 9. 팀원 구성 및 팀 역할분담

21500375 신지은:

일정 관리, 발표자료 만들기, Feedback, 기능 구현

21500643 정선일:

기능 구현, Feedback,

21700034 곽영혜:

기능 구현, Feedback

21700398 신유진:

Report 통합 및 수정, Feedback, 기능 구현 및 코드 통합