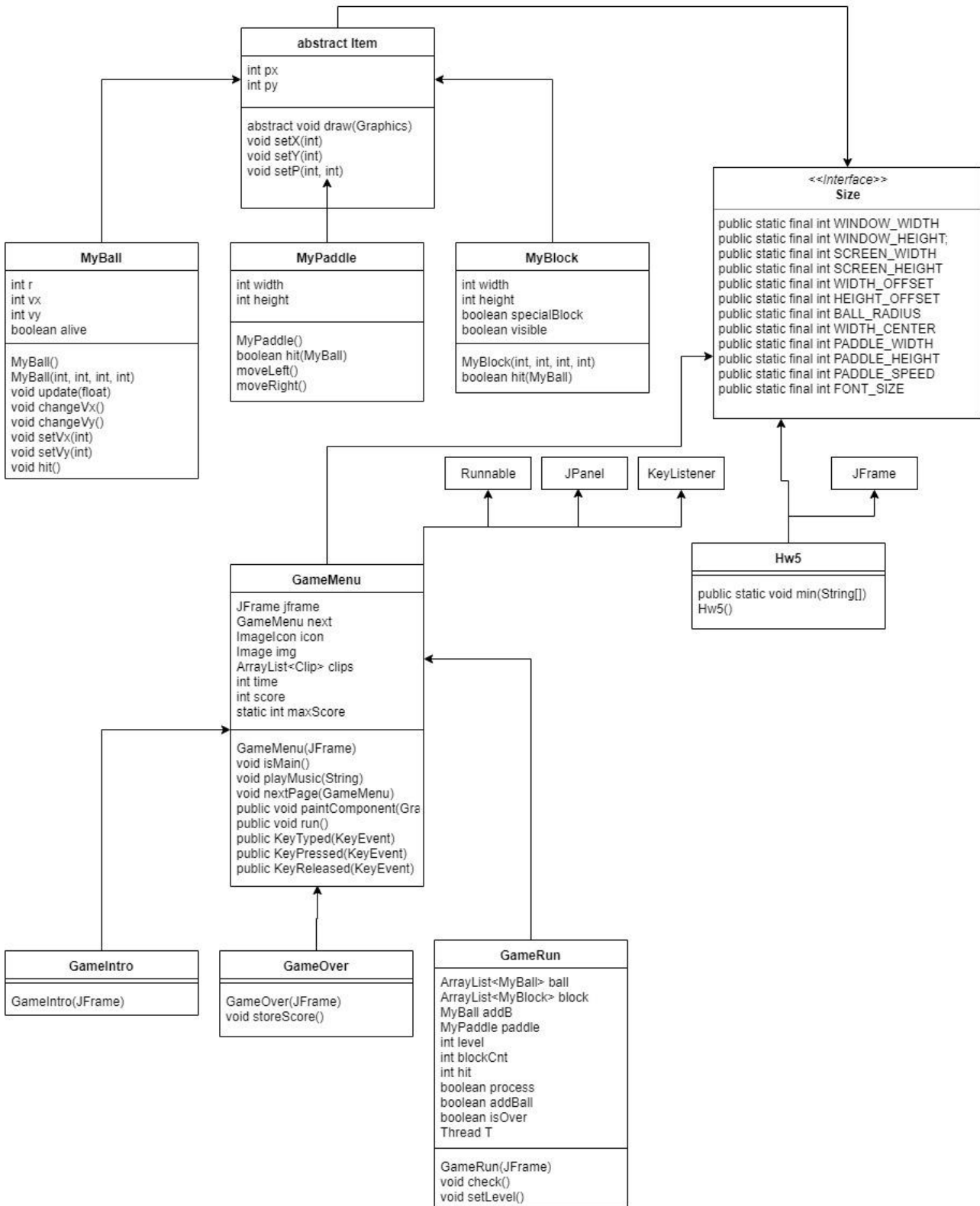


<Homework 5 – 벽돌깨기 게임>

컴퓨터공학과 18011529 김효경



<전체적인 클래스의 모습>

공, 블록, 패들과 같이 게임에서 필요한 아이템들을 abstract class Item을 만들어 상속받도록 했다.

-좌표 px, py를 필드로 가지고 draw와 같은 메소드를 가진다.

게임화면클래스들은 abstract class GameMenu 를 만들어 상속받았다.

- run, playMusic, nextPage, paintComponent 등의 메소드를 가지고 JFrame 객체, image 객체, 오디오 clip 리스트, maxScore등을 필드로 가진다.

<구현 방법>

1. 화면 전환 :

Game화면 클래스들은 JFrame 객체를 필드로 가지고 있다. 이는 현재 자신이 add되어있는 frame 이다.

각각의 화면은 JPanel을 상속받았기 때문에 Frame에 add되어야 보여진다. 이를 위해 GameMenu 클래스에 nextPage라는 함수를 두었다.

자기자신을 Frame에서 지운 후 다음에 올 화면을 같은 프레임에 add한다. 화면이 바뀌어도 점수를 저장해야하기 때문에 score를 복사해 넣는다. 또한 화면이 전환될 때마다 열렸던 음악파일을 모두 닫아야 하므로 clip리스트를 돌며 다 닫아준 후 리스트를 clear해 주었다.

2. 음악 재생 :

GameMenu 의 playMusic 이란 함수를 이용했다.

- 불러올 음악파일의 이름을 인자로 넘겨주면 그에 해당하는 파일을 열고 현재 화면에 재생되고 있는 clip 리스트에 add 해주었다.

3. 충돌 검사 :

충돌이 일어나는 공, 블록, 패들에는 각각 hit이란 함수를 두어 서로가 충돌할 때를 처리 해주었다. 블록과 패들의 hit함수에는 공 객체를 보내 자신의 좌표와 비교하게 한 후 충돌 시 return true, 충돌이 없을 시 return false 를 하도록 했다. 리턴 값을 이용해 충돌이 일어났을 경우 블록을 없애거나 소리를 발생시키는 등의 실행을 했다. 공은 화면과의 충돌을 고려해 벽에 부딪힌 경우 속도를 수정했다.

충돌이 일어난 블록은 visible 을 false로 바꾸어준 후 충돌검사도 하지않고 그림도 그리지 않도록 한다. 충돌이 블록개수만큼 일어나면 level을 바꾸고 맵을 새로 그린다.

공이 화면 밖으로 나가면 공의 alive를 false로 바꾸고 false가 된 공의 수가 ball리스트 사이즈와 같으면 게임오버가 된 것이므로 GameOver 화면으로 넘어간다.

4. 그 외 :

스페셜 블록 - 충돌하면 공이 3개 생성되는 블록이다. 블록 리스트를 생성할 때 각 블록은 1/5 확률로 스페셜 블록이 된다. (rand 함수 이용) Boolean 타입의 specialBlock을 두어 구별한다.

패들 움직임, 스페이스바로 시작 - keyListener 를 구현 해 keyPressed 에서 키 검사를 한 후 그에 맞게 움직인다. 패들에는 moveLeft, moveRight 함수를 두었다.

Thread - 각 패널들은 Runnable을 구현해 run함수에서 여러가지 일을 한다.

GameIntro -> run 하는 동안 while 문을 돌며 time을 증가시킨다. 이 time을 가지고 게임타이틀 이미지를 차례로 불러온다. 또한 Press spacebar 안내문을 time을 가지고 보였다 안보였다 하게 하였다. (time 을 2로 나눈 나머지 0, 1을 가지고 branch)

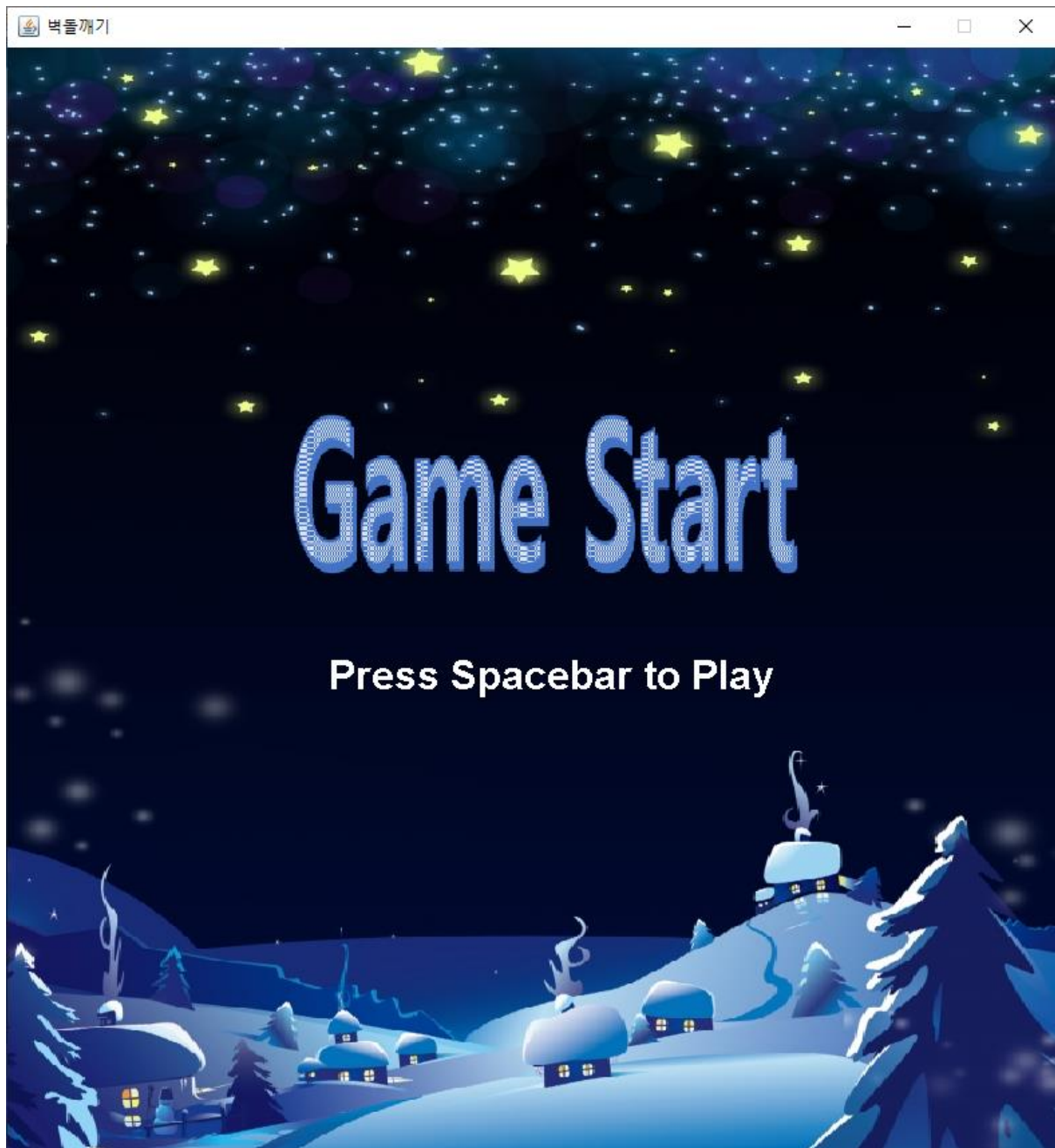
GameRun -> run하는 동안 이중 while문을 도는데 내부의 while문은 한 레벨을 시행하는 동안 도는 while문이다. 따라서 blockCnt와 hit수를 비교하며 process 가 true 인 동안 실행된다. 외부 while문은 gameover를 검사하는 while문으로 모든 공이 alive = false가 되면 isOver = true가 되어 모든 while문을 나오게 된다. 모든 while문이 끝나면 nextPage를 시행해 gameOver화면으로 넘어간다.

GameOver -> GameIntro와 같이 time을 증가시키며 이에 맞춰 화면에 글씨를 띄운다.

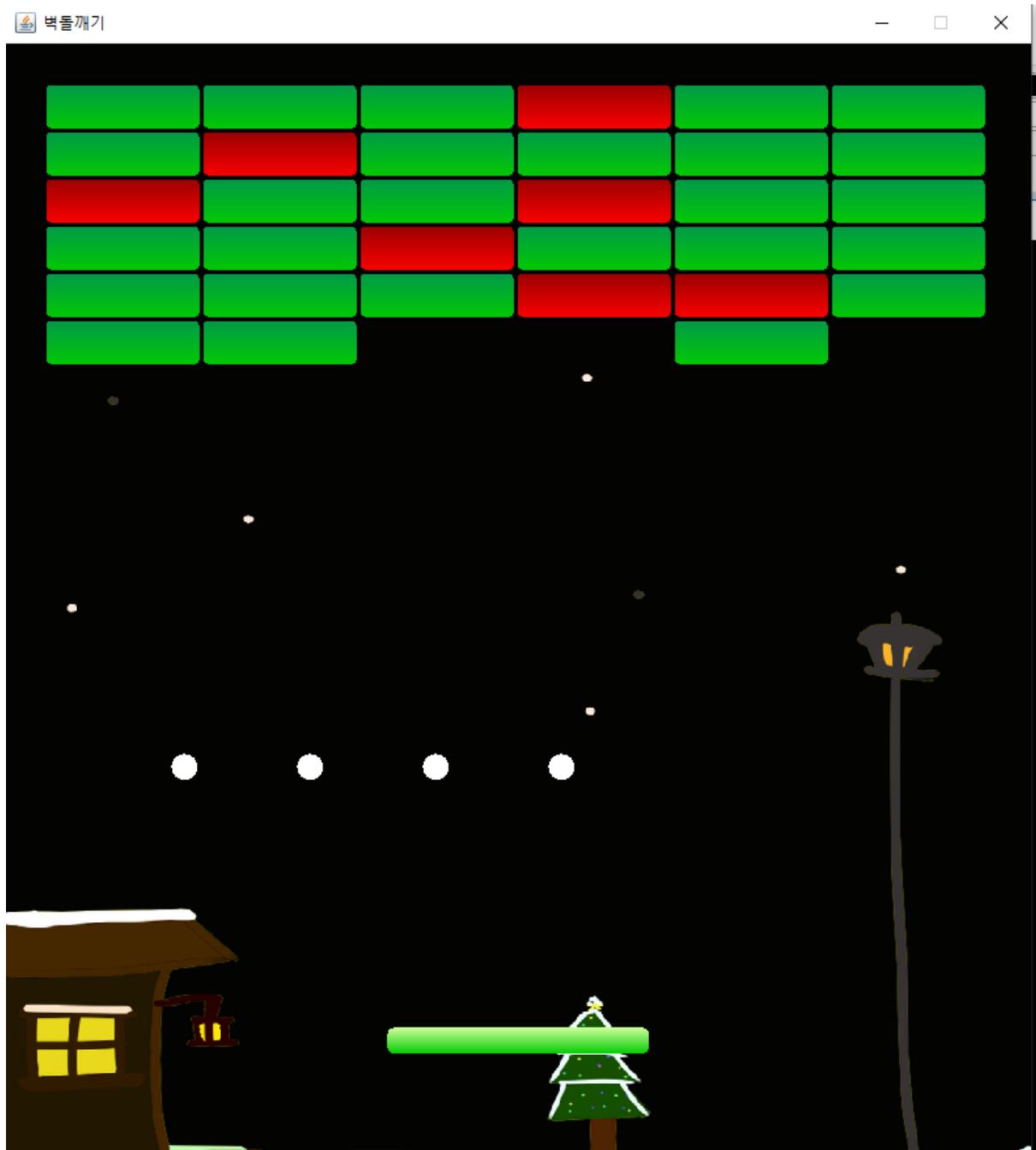
Size interface - 창크기를 고정시키고 요소들의 크기를 공유하면서 쉽게 구현하기 위해 필요한 값들을 멤버로 가지게 하였다.

Static int maxScore - GameMenu에서 maxScore를 static으로 둔 이유는 모든 자식클래스들이 같은 값으로 가지고 있으면서 함께 변하길 바라서이다. 이번 게임의 점수는 score값으로 가지고 GameOver까지 복사해 간다. GameOver에서는 이를 maxScore와 비교해 맞는 값을 maxScore로 가지고 있도록 storeScore함수를 갖고 run 함수가 시행 될 때 이를 수행한다.

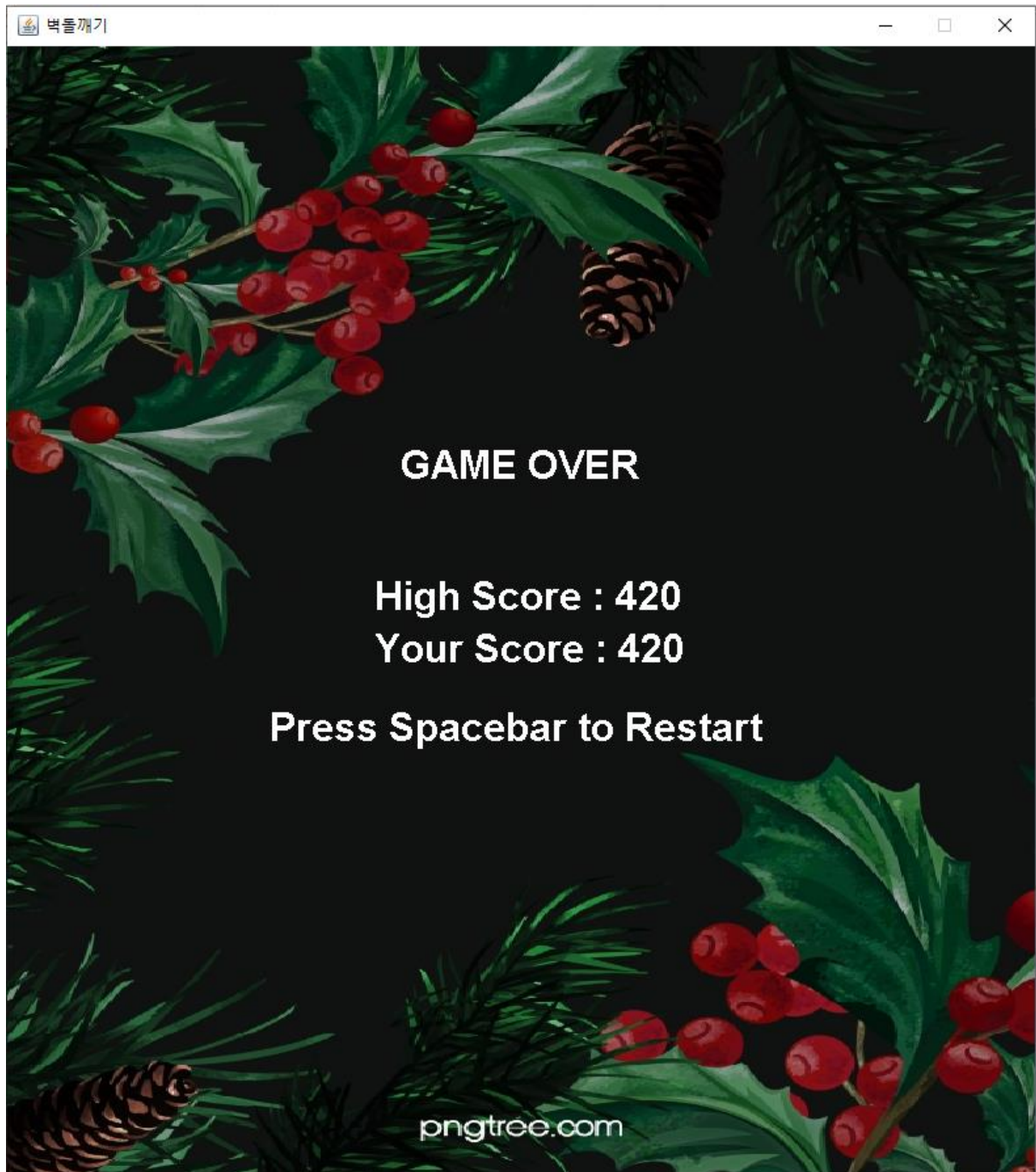
<실행화면>



- Start 화면



- 게임화면



- GameOver 화면

<느낀 점>

- 시간이 많이 부족해 원하는 걸 다 구현할 수 없어 아쉬웠다. 구현 시 어려운 점이 많았는데 특히 focus 와 audio가 힘들었다. 그 외에도 자잘한 버그들이 많아 계속해서 실행해보며 고쳐야 했다. 그래도 혼자 게임을 설계하고 코딩하여 결과가 나오는 걸 보니 매우 뿌듯했다. 팀플이 아닌 개인으로 프로젝트를 진행 한 느낌이라 나 자신에게 칭찬해주고 싶었다.

내용은 어려웠지만 자바를 좀 더 다양하게 활용해 볼 수 있는 시간이 되어 유익했고 다양한 구현을 공부할 수 있어 좋았다.