Jacky, Huanyu

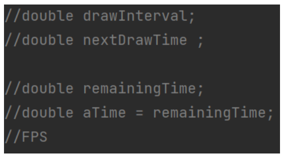
Code Review - Group 6

KeyHandler.Java:

- ***Changed confusing variable names:***
  - Old code: public Boolean uP, dP, lP, rP;
  - Modified: public Boolean upPressed, downPressed, leftPressed, rightPressed;
    Commit: 84e38b4659fe0675abfdf36e01828768086ac1de
- ***Changed ambiguous variable name:***
  - Old code: int code;
  - Modified: int keyCode;
    Commit: 84e38b4659fe0675abfdf36e01828768086ac1de

GameScreen.Java:

- ***Removed unused variables:***
  - Static Thread enemyThread;
  - Double drawInterval
  - Double nextDrawTime;
  - Double remainingTime;
  - Double aTime = remainingTime;

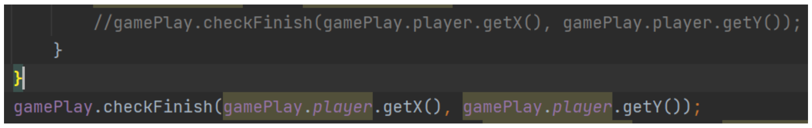


```
//double drawInterval;
//double nextDrawTime ;

//double remainingTime;
//double aTime = remainingTime;
//FPS
```

Commit: 84e38b4659fe0675abfdf36e01828768086ac1de

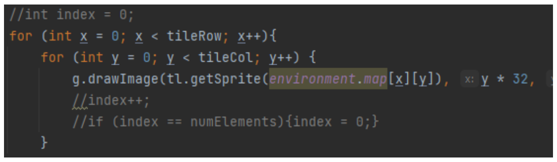
- ***Moved the function call outside the if statement to avoid unnecessary duplication***
  - Instead of checking whether the game finished in every if statement(to check which key was pressed), we move the code out and merge the four sentences into one. So that after no matter which statement was executed, they will do the same checkFinish function.



```
            //gamePlay.checkFinish(gamePlay.player.getX(), gamePlay.player.getY());
        }
}
gamePlay.checkFinish(gamePlay.player.getX(), gamePlay.player.getY());
```

Commit: 0551ccc774596804cfe764faf903689f8ca1f44e

- ***Unnecessary use of unsound construct:***
  - In `paintBackground` method and `paintTiles` method we have a variable called index to record how many times we call `g.drawImage` function, however, we never use this static in other places and it’s unnecessary to record that.



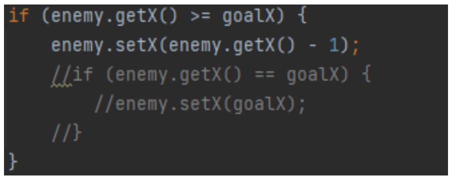
```
//int index = 0;
for (int x = 0; x < tileRow; x++){
    for (int y = 0; y < tileCol; y++) {
        g.drawImage(tl.getSprite(environment.map[x][y]), x: y * 32,
        //index++;
        //if (index == numElements){index = 0;}
    }
```

Commit: 0551ccc774596804cfe764faf903689f8ca1f44e

GamePlay.java:

- ***Code duplication:***

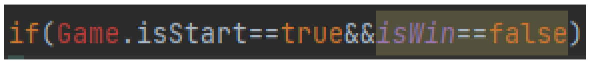
- In `setRandEnemyMovement` method, what the function does is to detect the enemy's position relative to the player's. I deleted the original code's method whose effect is to check if (enemy's position == player's position) then set enemy's location as its original location, it's unnecessary.



```
if (enemy.getX() >= goalX) {
    enemy.setX(enemy.getX() - 1);
    //if (enemy.getX() == goalX) {
        //enemy.setX(goalX);
    //}
}
```

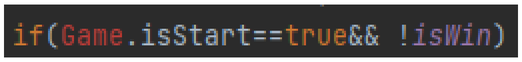
Commit: 0551ccc774596804cfe764faf903689f8ca1f44e

- ***Simplified if statement inside setRandEnemyMovement method:***
  - In `setRandEnemyMovement` method, the if statement is over complicated and instead of writing equals false, we can say "!isWin".



```
if(Game.isStart==true&&isWin==false)
```

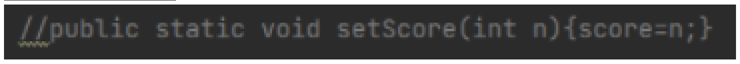
Modified into:



```
if(Game.isStart==true&& !isWin)
```

Player.java:

- ***Removed unused Function:***
  - I delete the unused function `setScore()` because the features are replaced by `winReward()`

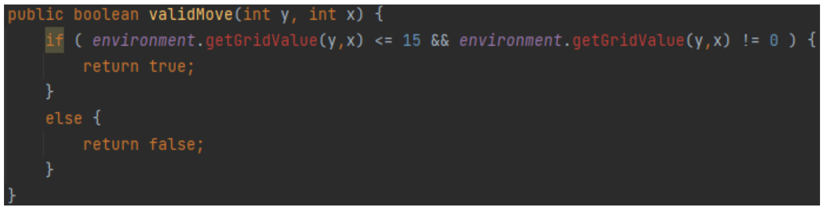


```
//public static void setScore(int n){score=n;}
```

Commit: 0551ccc774596804cfe764faf903689f8ca1f44e

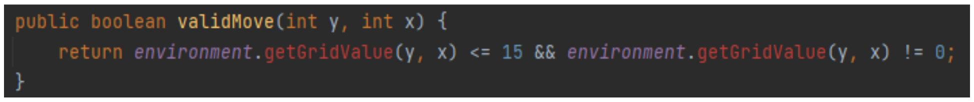
Bonus.java:

- ***Methods that are too long and that could benefit from being refactored :***
  - Simplified Boolean function inside valid move method:



```
public boolean validMove(int y, int x) {
    if ( environment.getGridValue(y,x) <= 15 && environment.getGridValue(y,x) != 0 ) {
        return true;
    }
    else {
        return false;
    }
}
```

Modified into simplified return for the given values:



```
public boolean validMove(int y, int x) {
    return environment.getGridValue(y, x) <= 15 && environment.getGridValue(y, x) != 0;
}
```

- ***Refactored code for validMove Method replaced by abstract parent class:***
  - Bonus.java and Reward.java both had validMove. We removed this and put it inside the gameObject Class which is the parent class of both Bonus and Reward. By doing this we are reducing coupling between the classes depending on each other. All objects in the game now extend GameObject which includes the methods validMove, randPosition, getters and setters for positions.
  - Commit: cc5de64e0ffcb67af159c34f9a920049e368a6cc