

Basic Programming Project 2018-2019: Dance with Stars

Yi Zhao-r0685182

Zhou Wang-r0689176

Game Introduction

This is the story of our game Dance with Stars.

At the beginning of the game, you will see the start window:

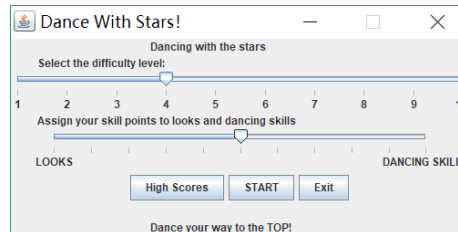


Figure 1: Start Window

You choose the difficulty level (*diffLevel*, from 1 to 10 with the default value 4) and assign the distribution of looks and dancing skills (*tradeOff*, from 0 to 10 with default 5). There are four attributes: *energy*, *scoreDance*, *scoreLook* and *scoreTotal*. At the very beginning, the choices of *diffLevel* and *tradeOff* will give you original values of energy and scores:

$$energy = 200 - 10 * diffLevel$$

$$scoreLook = 20 * tradeOff$$

$$scoreDance = 200 - 20 * tradeOff$$

$$scoreTotal = scoreLook + scoreDance$$

When you click the “START” button, you are at the stage randomly on a grid of a 10*10 board. For example:

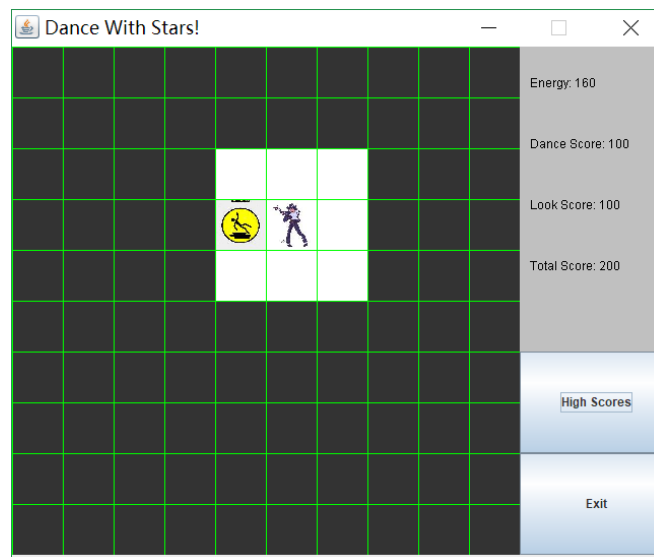















Figure 2: Game Window

The grids around you are white. It means you can move to any of these grids by clicking the mouse. However, those black grids are beyond your reach. Among all these grids around you, there is one grid with meaning. The symbols and corresponding meanings and effects are shown in Table 1.

Table 1: Symbols, Meanings and Effects

Symbol	Meaning	Effects
	Player	For any step you move, the energy will decrease by 20.
	Juror 1	When you click the grid, if your <i>scoreDance</i> is higher than the threshold set by Juror 1 (100), your <i>scoreDance</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Juror 2	When you click the grid, if your <i>scoreDance</i> is higher than the threshold set by Juror 2 (200), your <i>scoreDance</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Juror 3	When you click the grid, if your <i>scoreDance</i> is higher than the threshold set by Juror 3 (300), your <i>scoreDance</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Star 1	When you click the grid, if your <i>scoreLook</i> is higher than the threshold set by Star 1 (100), you impress the star and your <i>scoreLook</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Star 2	When you click the grid, if your <i>scoreLook</i> is higher than the threshold set by Star 2 (200), you impress the star and your <i>scoreLook</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Competitor 1	When you click the grid, if your <i>scoreTotal</i> is higher than the threshold set by Competitor 1 (300), you win the competitor and your <i>scoreLook</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Competitor 2	When you click the grid, if your <i>scoreTotal</i> is higher than the threshold set by Competitor 2 (500), you win the competitor and your <i>scoreDance</i> will increase by 50. The <i>scoreTotal</i> updates, too.
	Award for Look	It is the award for your look. Your <i>scoreLook</i> will increase by 150. The <i>scoreTotal</i> updates, too.
	Award for Dance	It is the award for your dance. Your <i>scoreDance</i> will increase by 150. The <i>scoreTotal</i> updates, too.
	Blind Light	When you click the grid, you are going through an area with blind light. Your energy will decrease by 30 but <i>scoreDance</i> will increase by 100. The <i>scoreTotal</i> updates, too.
	Slippery Floor	When you click the grid, you are going through an area with slippery floor. Your energy will decrease by 40 but <i>scoreDance</i> will increase by 100. The <i>scoreTotal</i> updates, too.
	Trophy	If your <i>scoreTotal</i> is higher than the threshold value, the trophy will appear. Then you can claim it if your energy doesn't run out.

If your energy is over, or you claim the trophy, the game is over. Then the *HighScores* record will be updated. As is shown in Figure 3, the record includes the highest three *scoreTotal* values in history.

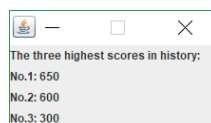


Figure 3: *HighScores* Record Example

Class Description

The codes are in Java. All classes including fields and methods as well as their relationship are shown in Figure 4.

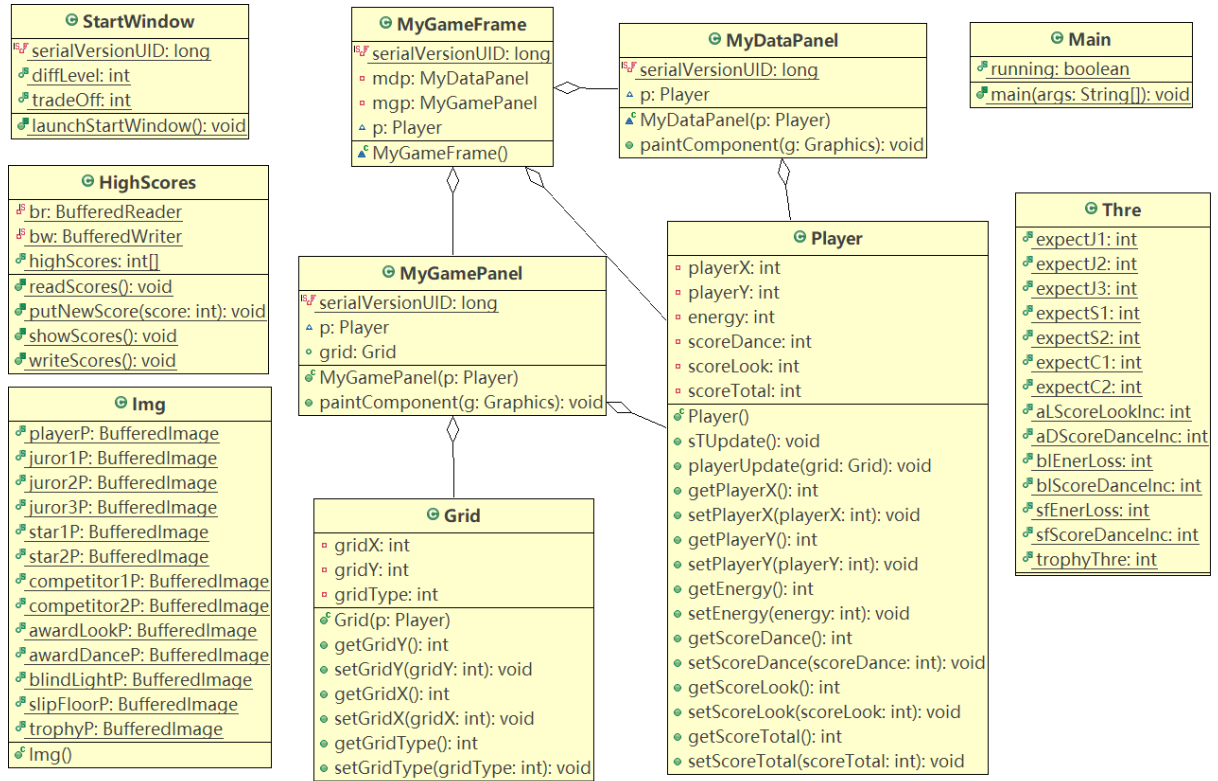


Figure 4: UML

When the codes start to run, the function *launchStartWindow()* is activated. And we can see the start window (Figure 1). If we click on the “High Scores” button, the function *showScores()* of class HighScores is activated, and we can see the highest three scores in history (Figure 3). If we click on the “START” button, a new instance of *MyGameFrame* is created and initialized (Figure 2). It includes *mdp*, *mgp*, *p*. The player *p* is an instance of the class *Player*, when it moves, depending on its surroundings, *p* updates via the function *playerUpdate(Grid)*, and its *scoreTotal* updates via *sTUpdate()*. When *p* moves, *mgp*, which is an instance of the class *MyGamePanel* repaints. It is also the case for *mdp*. The state update is achieved by the function *paintComponent()* (*repaint()*). When the game is over, we click “Exit” button to exit.

Strengths, Weaknesses and Difficulties

The main strengths of our project are: it is runnable, and we applied lot of knowledge we learnt from class, which including OOP, I/O processing, Exceptions, GUI and so on. The main weakness of this project is: it might be a very simple game and not interesting enough.

During the process, we faced 3 main difficulties. The first one is deciding how the player interacts with other elements (i.e. how the player’s energy and scores change). The second one is how to achieve the state changes (i.e. by the *paintComponent()* and *repaint()* functions). The third one is how to solve bugs with patience and insist on the project until it works.