VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

PROJECT REPORT

Candy Crush

Class: **Object-Oriented Programming – ITIT19IU31**

Instructors: **MSc. Le Thanh Son**

MSc. Nguyen Quang Phu

Members: **Bui Thi Xuan Lan - ITDSIU19007**

Nguyen Duc Minh - ITITIU19030

Ho Chi Minh City, 8 January, 2021

Table of Contents

I. Introduction

Candy Crush is a free-to-play match-three puzzle video game released by King on April 12/2012 for Facebook. Other versions for iOS, Android, Windows Phone, and Windows 10 followed after. Right now, the game is still alive with 9.2 million players and has become most successful on mobile devices.

Our project builds on the match-three principle on a simpler level with our own designs. It can be accessed on Github here.

The project was written in Java for Windows.

II. Terminology

Tile: a square carrying an icon that can be clicked to make a move.

Figure 2.1.: Five different tiles with the first one selected.

Grid: multiple tiles forming a square or a rectangle.

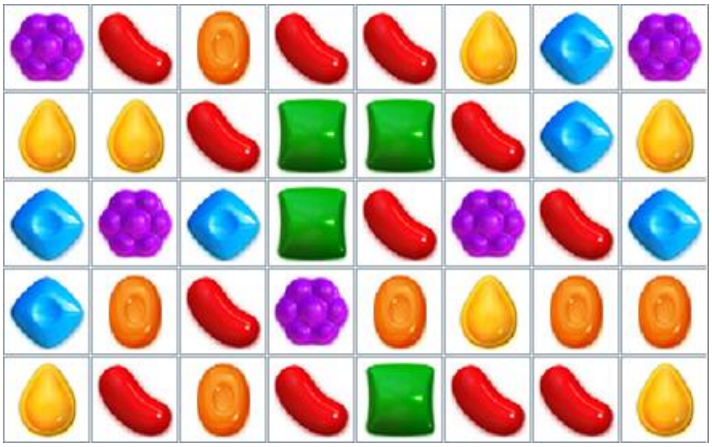


Figure 2.2.: A 5-by-8 grid.

Row: the height of the grid (in number of tiles). In Figure 2.2., row = 5;

Column: the width of the grid (in number of tiles). In Figure 2.2., column = 8.

Match: three or more tiles forming a vertical or horizontal line.

Move: when two adjacent tiles are selected.

III. Game Rules

The objective is to match tiles of the same variations and make them disappear to earn points, then the tiles on top of them will drop down and new tiles will appear.

The player makes a move by clicking two adjacent tiles to switch them. If a match occurs, the grid clears the matched tiles and randomizes new tiles which are dropped down.

- If there are more than one match after a move, the game updates the GUI after all those matches are processed.
- The total matches are tallied and displayed at the top.
- Each tile is given a score, and the total score is calculated accordingly and displayed at the top.

The player is able to play until there is no more possible move. Then, the game is over.

IV. Features

1. *Main features*

Figure 4.1.: The home screen.

The game starts with the home screen (Figure 4.1.) where there are three buttons:

- “Play!”: start a new game,
- “Exit”: exit the program,
- “Leader Board”: see the leader board.

a. After “Play!” is clicked, the game tells the player to choose a play mode:

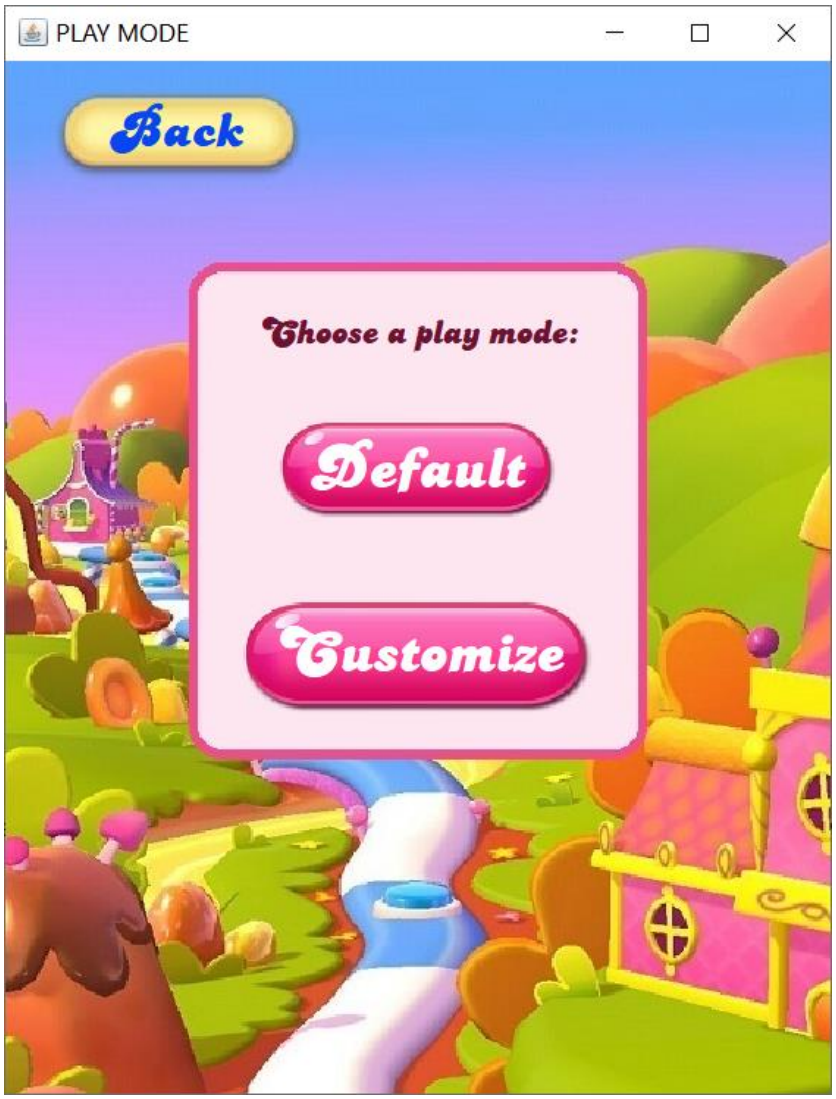


Figure 4.2.: The play mode screen.

- “Default”: play a default game with preset tiles and character icon, and play grid of size 5 by 5 (Figure 4.4.),
- “Customize”: open the customize screen, where the player can choose a tile theme and a character icon, and set the play grid’s dimension (minimum of 5 and maximum of 10, both width and height) (Figures 4.3. and 4.5.).

Figure 4.3.: The customize screen.

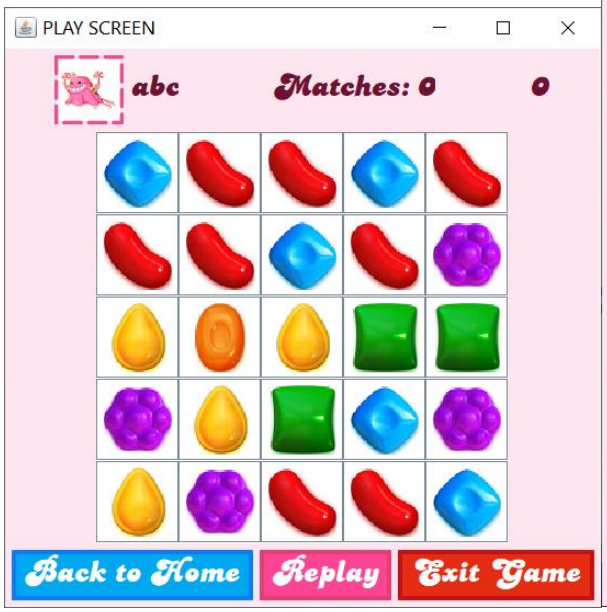


Figure 4.4.: The play screen from option “Default”.

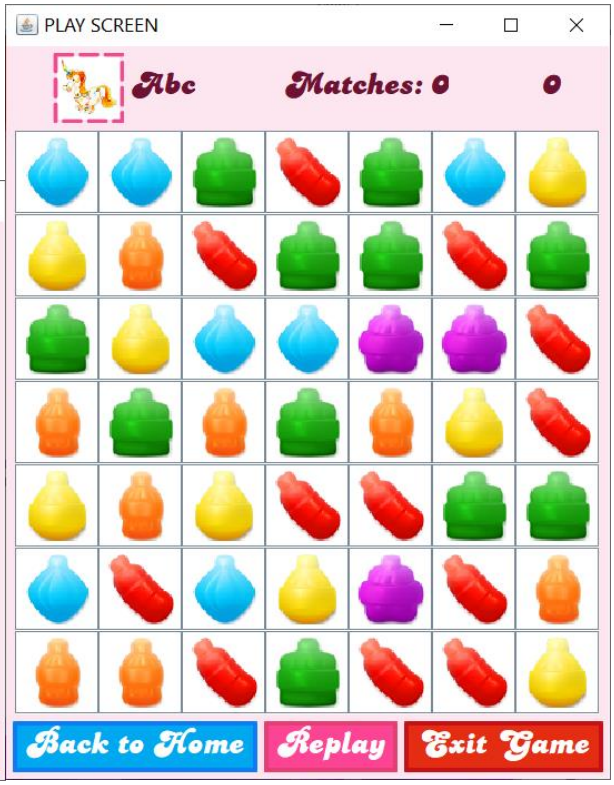


Figure 4.5.: The play screen from option “Customize”.

When the game is over, the grid is disabled and the player is shown the results.

Figure 4.6.: The game over screen.

The player's icon, name, and final score are displayed on the top left.

The buttons to proceed are displayed on the bottom left.

The leader board is displayed on the right.

b. "Leader Board" is clicked:

If the program is started for the first time, there is no data to display the leader board.

Figure 4.7.: The empty leader board.

Otherwise, the leader board is displayed normally.

Figure 4.8.: The leader board.

2. *Extra features*

Leader board: shows a list of play records (Figures 4.6. and 4.8.). In details,

- Position: where players rank,
- Date and Time: when players finished their games,
- Name: players' names,
- Score: players' score.

Sound: a sound plays when a match is made.

Pop-ups:

- Information: where the player inputs their name (after Figure 4.2.).

Figure 4.9.: The name input pop-up.

- Warnings:
  - When the player clicks “Go” in Figure 4.9. without an input.

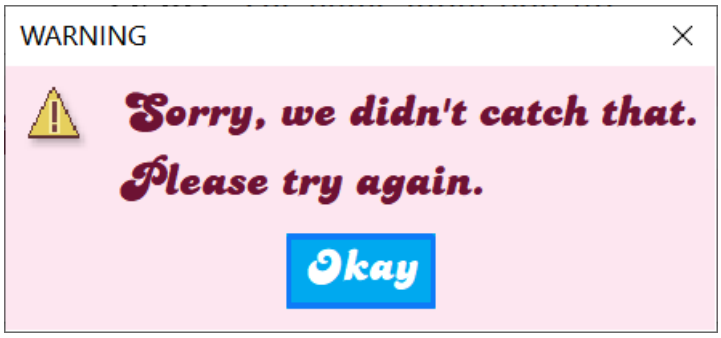


Figure 4.10.: The no input warning.

  - When the player clicks the buttons in Figures 4.4. and 4.5. before finishing their game.

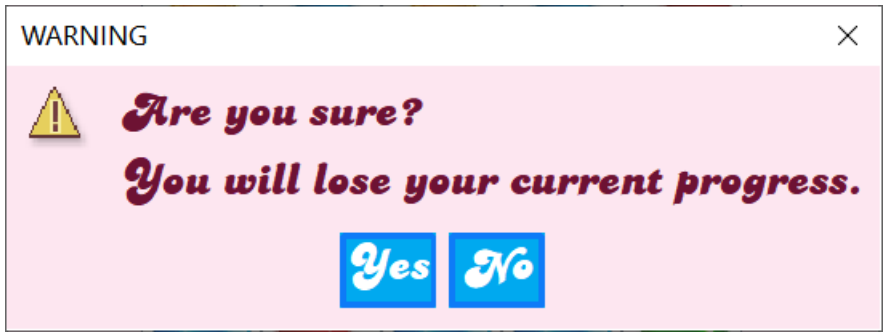


Figure 4.11.: The progress warning.

- Messages:
  - When the player is added in the leader board (on Figure 4.6.).

Figure 4.12.: The new record message.

- When the player sets a new record in the top three in the leader board (on Figure 4.6.).

Figure 4.13.: The top three message.

Packages: load images into the game (Figure 4.3.).

- Tile packages: different themes for the tiles (nine themes, six variations each).
- Character package: the character icon for the player (15 characters).

V. Workflows

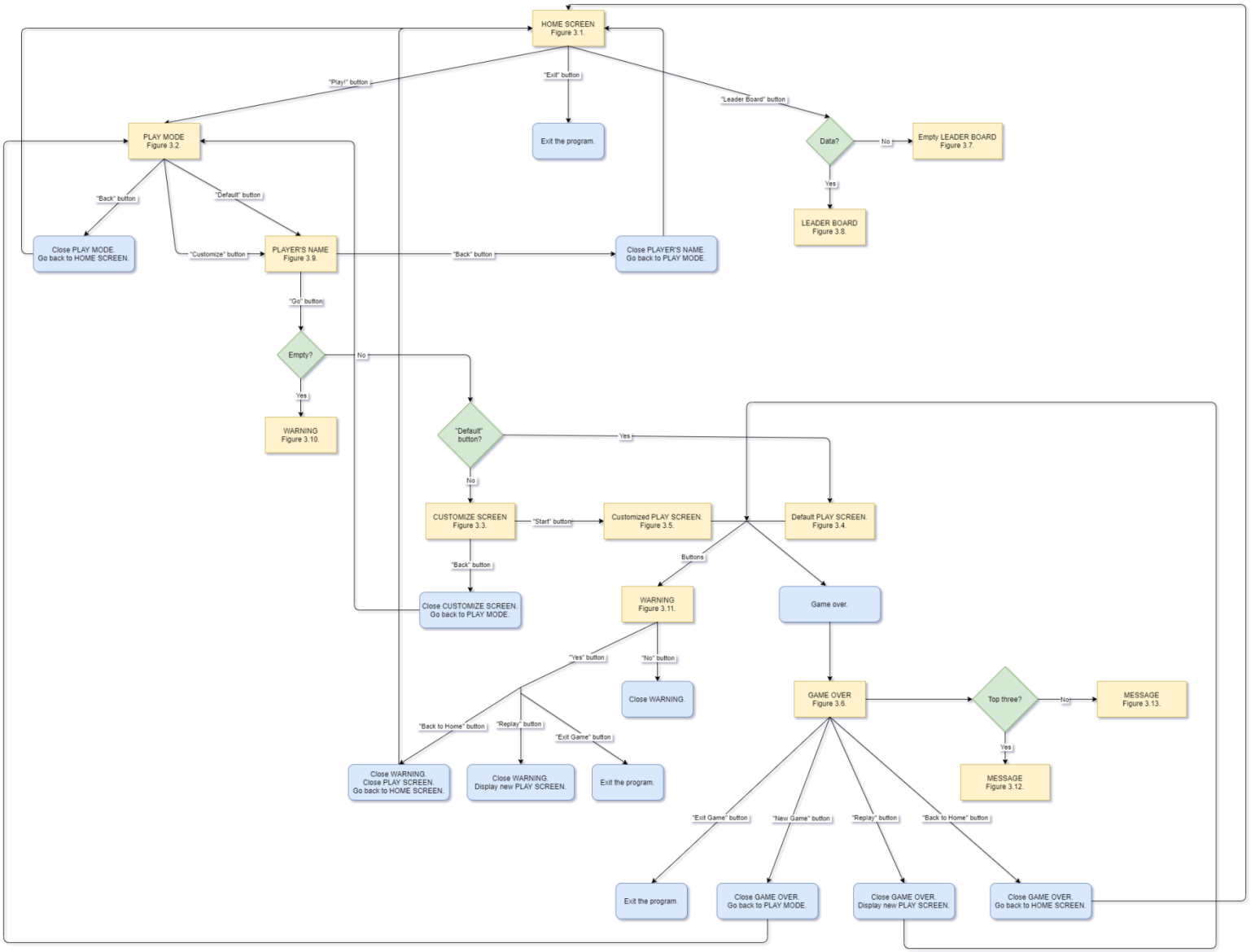


Figure 5.1.: Workflow diagram.

This is a diagram of all events that can happen in the game.

Notes:

- Screens: yellow color,
- Buttons: written on arrows,
- Actions: blue color,
- Forked events: green color.

VI. UML Diagrams

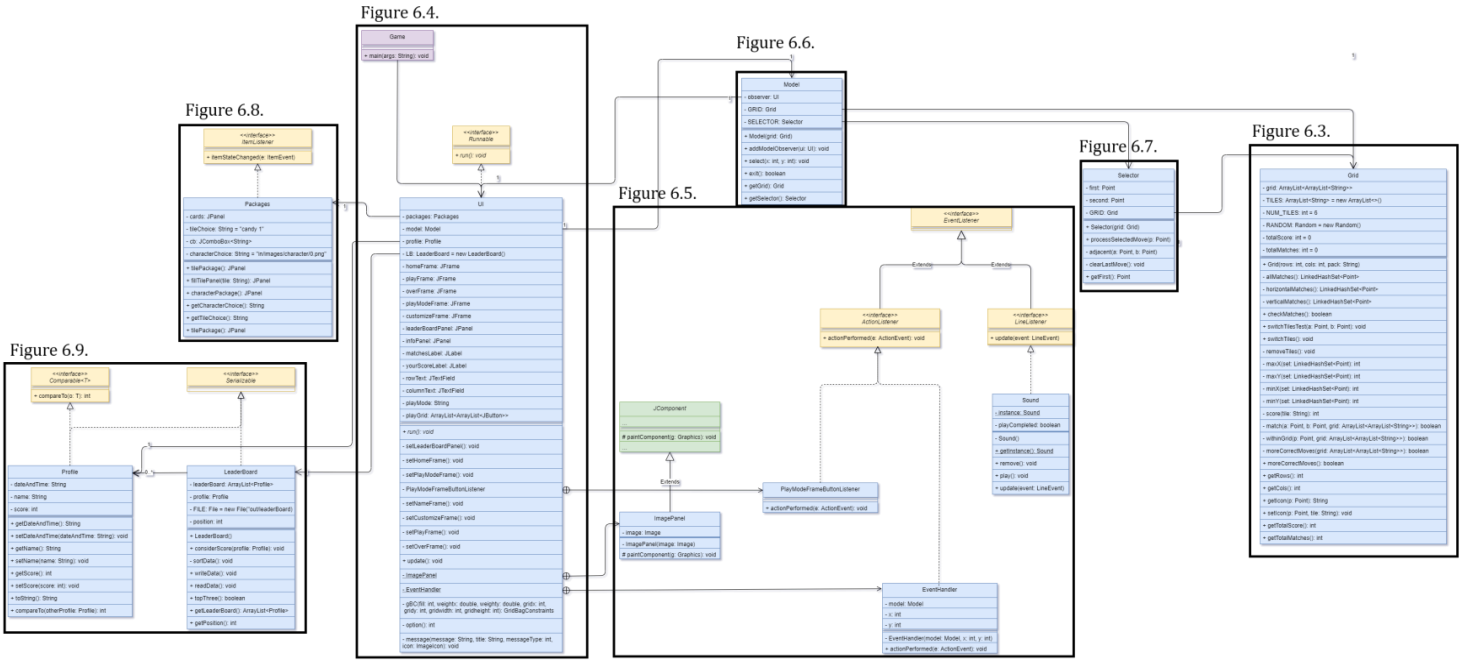


Figure 6.1.: UML diagram.

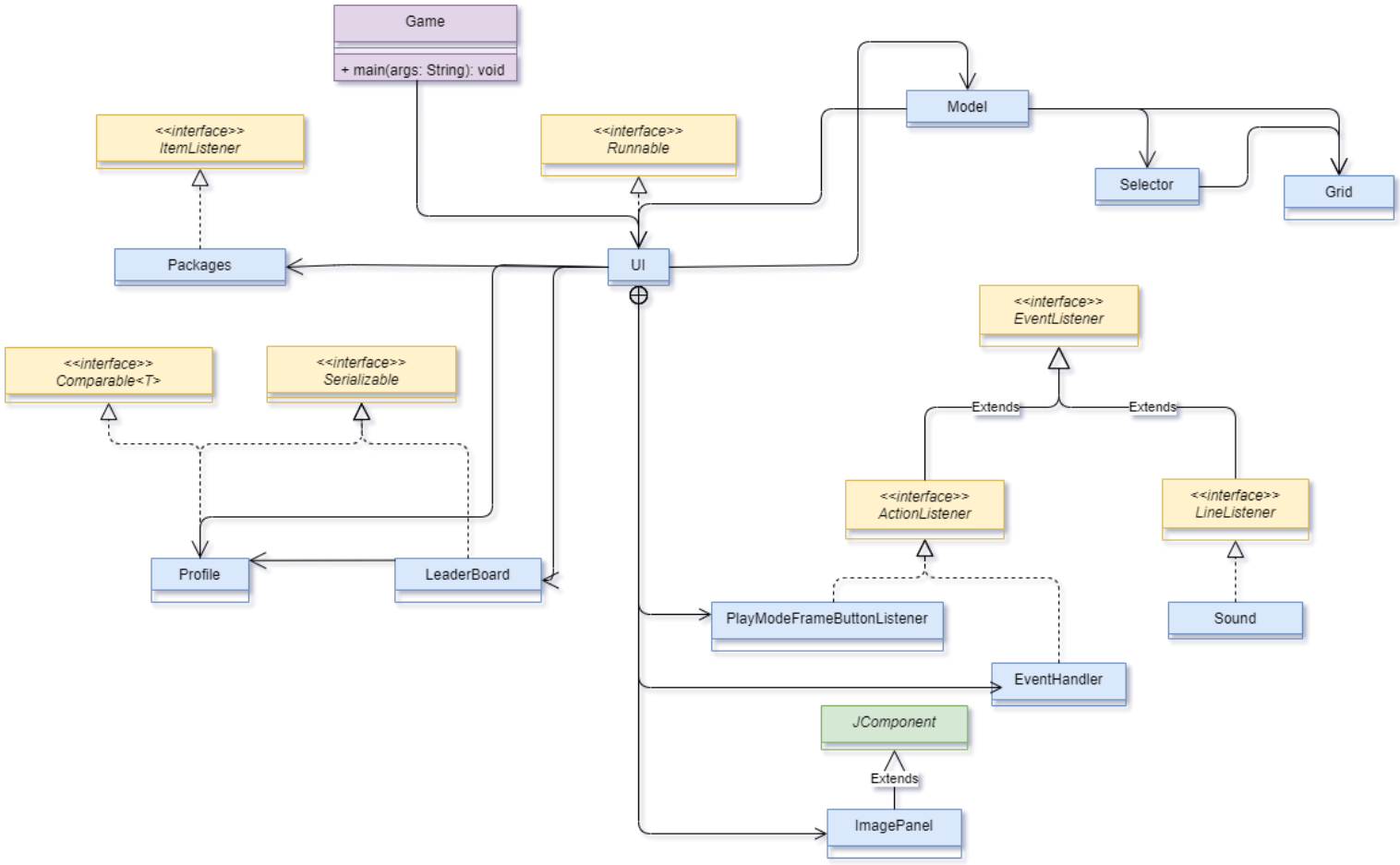


Figure 6.2.: Collapsed UML diagram.

Figure 6.3. (right): Class Grid.

Figure 6.4. (left):

Classes Game and UI.

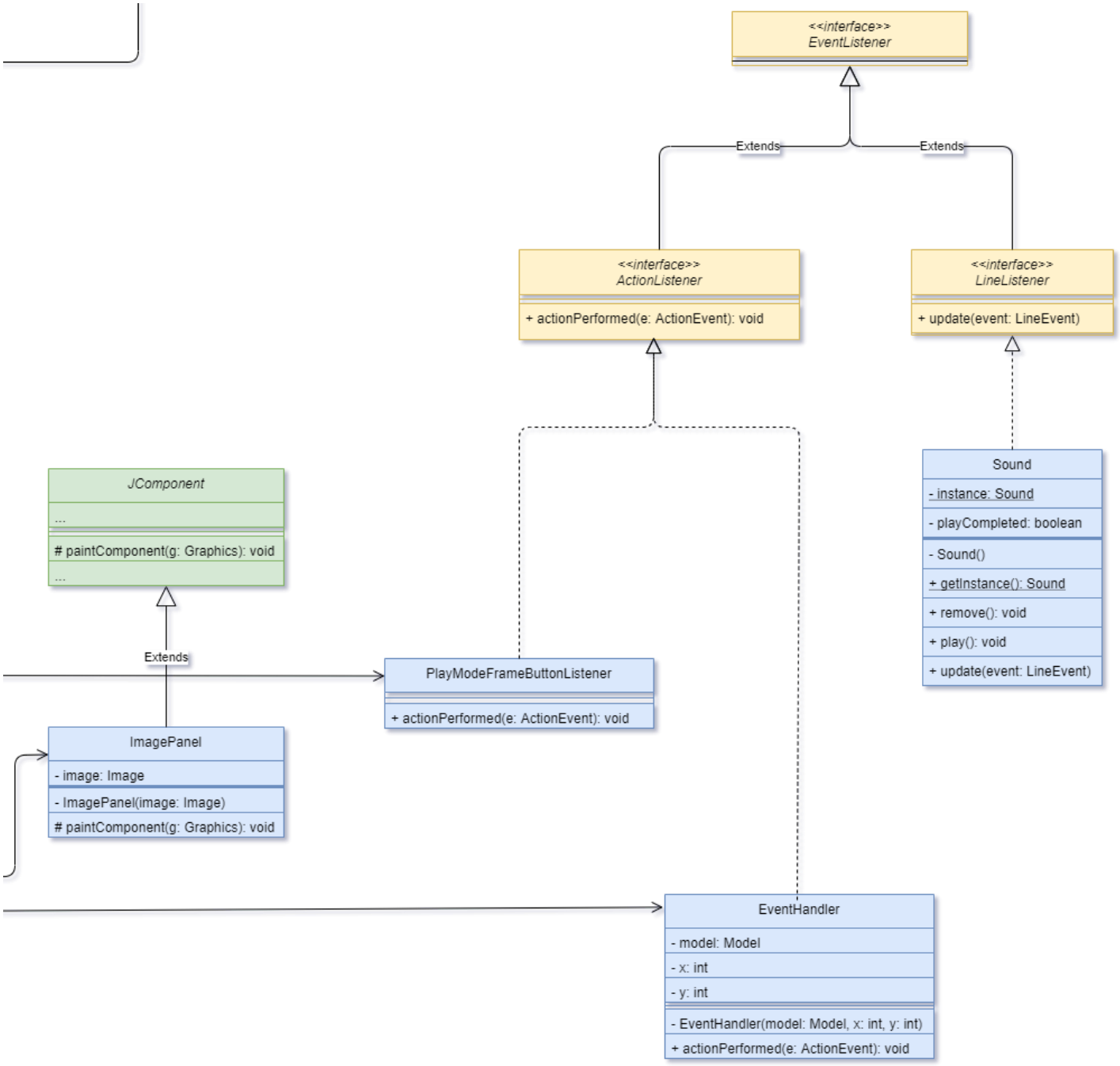
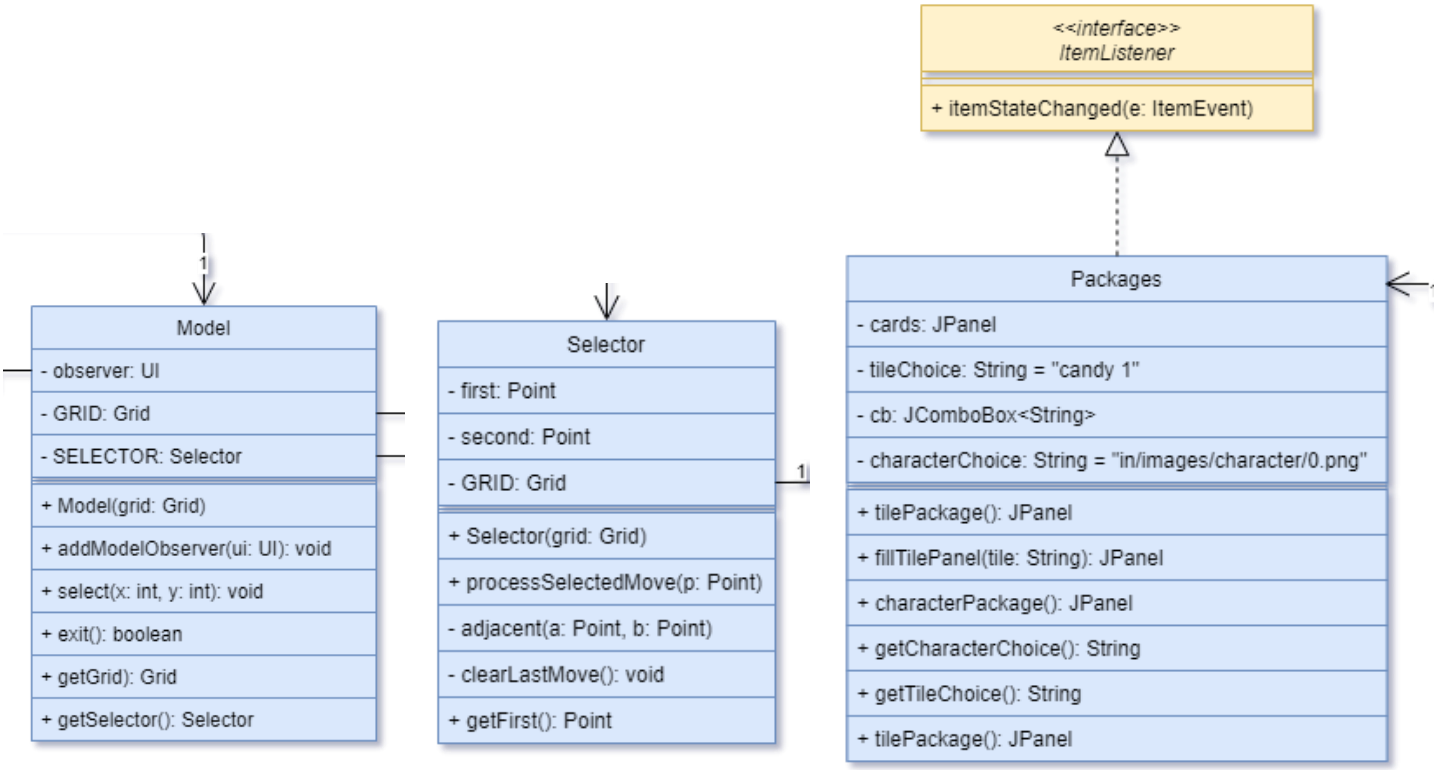


Figure 6.5.: Class Sound and nested classes (in class UI) ImagePanel, PlayModeFrameButtonListener, and EventHandler.



Figures 6.6. to 6.8. (left to right): Classes Model, Selector, and Packages.

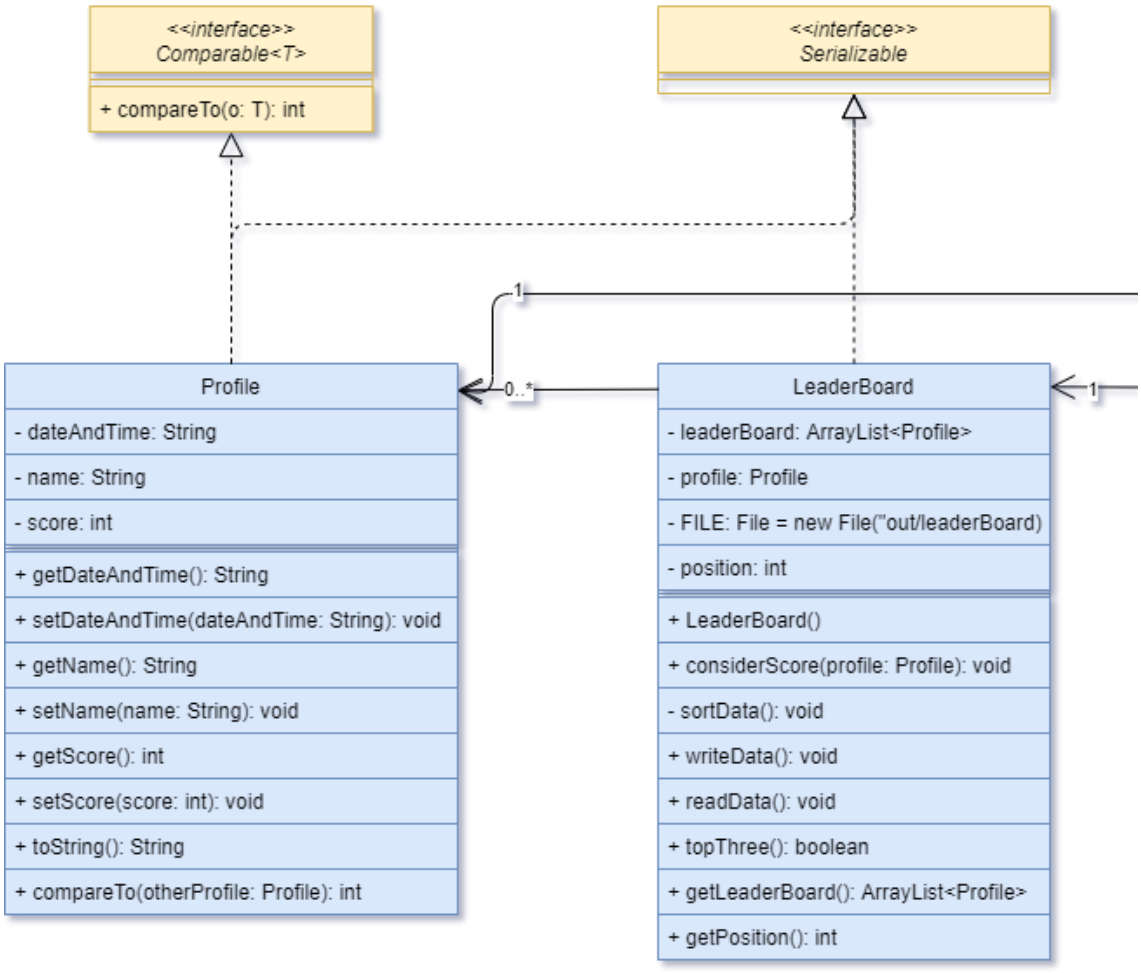


Figure 6.9. (left to right): Classes Profile and LeaderBoard.

Figure 6.1. shows the full UML diagram. As it is too large to view comfortably, Figure 6.2. shows a UML diagram with all the classes collapsed.

Figures 6.3. to 6.9. show the classes in detail.

Notes:

- Interfaces: yellow color,
- Abstract classes: green color,
- Concrete classes: blue color.

VII. Classes

- *Package model:*

  Class Grid (Figure 6.3.) handles functionalities in the play grid.

  Class Model (Figure 6.6.) handles changes in the UI and the play grid.

  Class Selector (Figure 6.7.) handles selections in a play grid.

- *Package ui:*

  Package leaderBoard (Figure 6.9.):

  - Class LeaderBoard creates the leaderboard for the game.
  - Class Profile saves every player's date and time, name, and score.

  Class Packages (Figure 6.8.) packs images into JPanels for use in UI.

  Class Sound (Figure 6.5.) plays a sound using Singleton Pattern.

  Class UI (Figures 6.4. and 6.5. (nested classes in UI)) displays the game.

- *Class Game (Figure 6.4.):* main class to run the game.

See code for documentation details.

VIII. Design Patterns

Class Sound uses Singleton Pattern.

Class UI uses Command Pattern.

Class Model uses Observer Pattern.

Appendix

Link to Github: https://github.com/btxl246/OOPCandyCrush
Link to reference: https://github.com/sriniketh923/Candy-Crush

Project started on Oct 12, 2020.
Project finished and reported on Jan 08, 2021.

Roles:
Xuan Lan: UI, Model, Grid, Seletor, Packages, Sound, documentation.
Duc Minh: UI, Profile, LeaderBoard, documentation.