Name: Mean Diamand

Neptun Code: KSG25Z

Flappy Bird Game

Specification

For Java Project

Description of the Game

This project is a low budget replicated version of the mobile game called Flappy Bird that was so popular many years ago on both App store and Play store.

This game development is written in Java and based on an object-oriented programming style.

The application allows the user to control the small cartoon bird figure that flies over the pipes, which will be the obstacles for the game. After passing each pipe, the user will gain one point, and the game will keep track of those points after the player loses the game by saving it on the High score board, and it will also ask the username of the player and shows it with the score on the high score board. There will be many types of birds that you can pick to play as.

me

The game will have two levels of difficulty:

1. Easy mode
2. Hard mode (which the speed and jumping height of the bird will be increased)

The game will be over if the bird falls off the screen or hits the pipe. After that it will ask the user to enter the name and the user can choose to play the game again or exit the game.

Game Layout

1. Main Menu Screen
    - Components: Start Game button, High Score button, and Exit button.

- Description: The initial display of the game after opening, it displays the picture of the flappy bird game and the buttons which the user can start, check the high score, or exit the game.
- Access: This is the default screen on game launch.

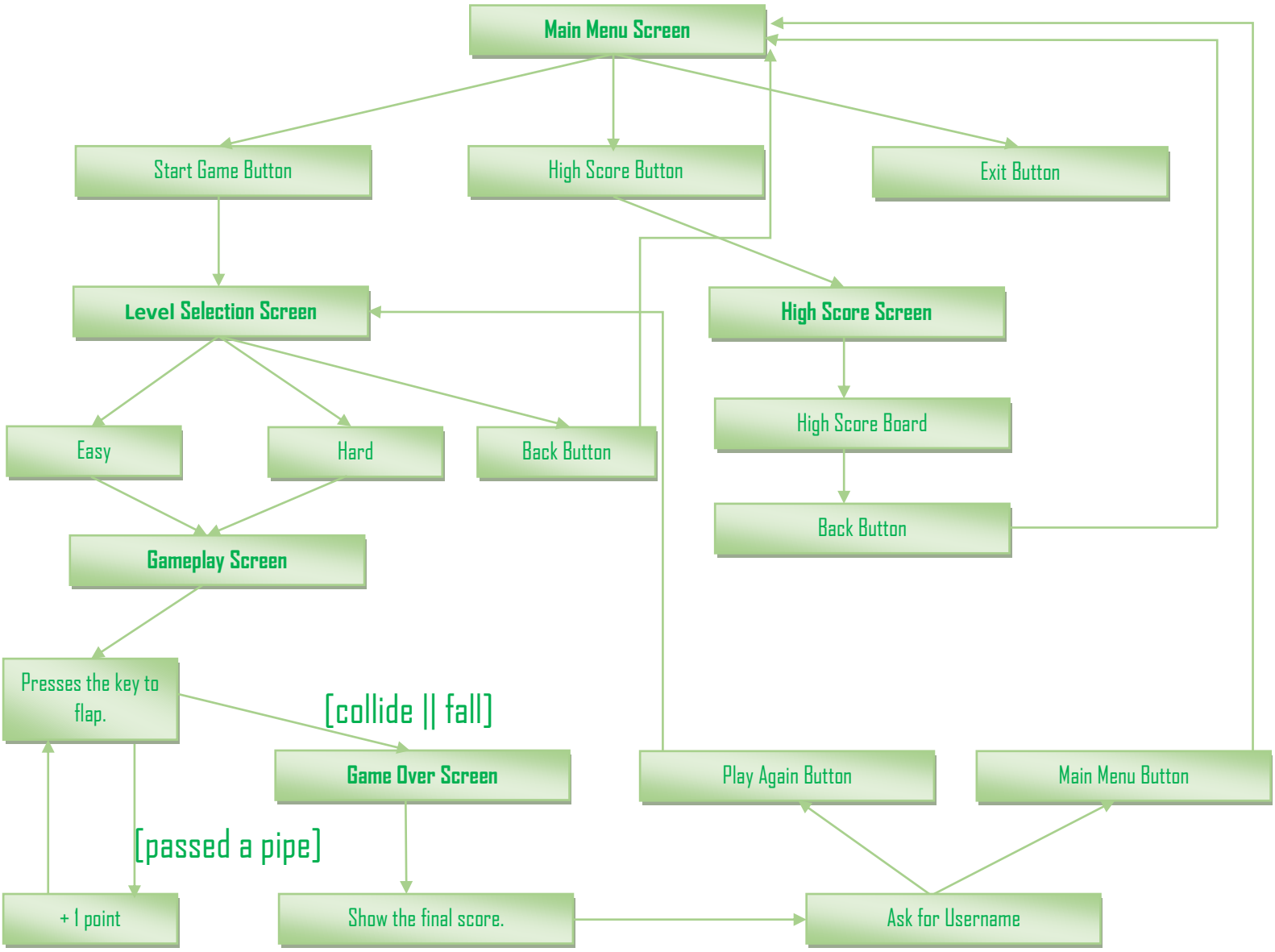
2. Level Selection Screen
    - Components: Easy Mode button, Hard Mode button, and Back button.
    - Description: Lets the user select their level of the game.
    - Access: Clicking on the Start Game button in the Main Menu will direct the user here.
3. Gameplay Screen
    - Components: A dynamic background, cartoon bird figure, pipes, current score.
    - Description: The primary game interface where the user controls the bird.
    - Access: After choosing the level of difficulty, the user will come to this screen.
4. Game Over Screen
    - Components: Final score display, Username input box, Play Again Button, Main Menu button.
    - Description: Displays after the game ends, showing the user's score and asks them to input their name for saving with the high score.
    - Access: Appears after Game Over which happens when the bird collides with the pipe or fall.
5. High Score Screen
    - Components: List of high scores with the usernames, and Back button.
    - Description: Showing the user's achievement.
    - Access: can be accessed from the Main Menu.

Name: Mean Diamand

Basic of Programming 3

Neptun Code: KSG25Z

Use-case Diagram



List of all use-cases in the game:

1. Main Menu Access
    - Description: The user launches the game, and a simple interface will be shown with a primary game option.
    - Flow: Launch game -> Main Menu.
2. Starting the game

- Description: The option to start the game.
- Flow: Main Menu -> Level Selection.

3. Choose level (Easy or Hard)
    - Description: After starting the game, the user can select between Easy and Hard modes of the game.
    - Flow: Level Selection -> Gameplay.
4. Play the Game
    - Description: The user controls the bird, trying to pass the pipe without falling and colliding with the pipes to gain 1 point every time he or she passed a pipe.
    - Flow: Bird flaps with a key presses -> +1 point for each pass from the pipes -> Game Over if there is any collision or fall.
5. Recording High Score
    - Description: When Game Over, the user's score is saved into the high score board and the username will be asked.
    - Flow: Game Over Screen -> Input username -> save score
6. Viewing the High Score
    - Description: The user can review the high score board.
    - Flow: Main Menu -> High Score Screen.
7. Exiting the Game:
    - Description: The user can exit the game application.
    - Flow: Main Menu -> Exit button -> Exit Game.

Technical Addendum

I'm planning to use Swing and Abstract Window Toolkit(awt) libraries which will help me to create the user interface for my project application such as:

- Rendering Capabilities: This ensures that the game components, including the bird, pipes, and background, are drawn efficiently on the screen.
- Event Handling: The game will react to user input like key presses (for instance, to make the bird flap its wings) and button interactions (e.g., starting the game, viewing the high score, or exiting).
- Image Management: AWT provides classes that help image handling. These classes are essential for loading, manipulating, and displaying various game graphics on the window.

- Window and Panel Management: I will employ classes like JFrame and JPanel to create the main game window and various panels within the game.

Additionally, to save player usernames and their corresponding high scores, I plan on using the Serializable interface. This will allow me to serialize the game data and store it in a file. Later, when I need that information to load back, I can deserialize the file contents, effectively retrieving and displaying saved scores and player names in the application.

As the development progresses and I might gain more insights from the ongoing lectures through the semester, I may integrate additional techniques and libraries to enhance the game's functionality and user experience later in the future.