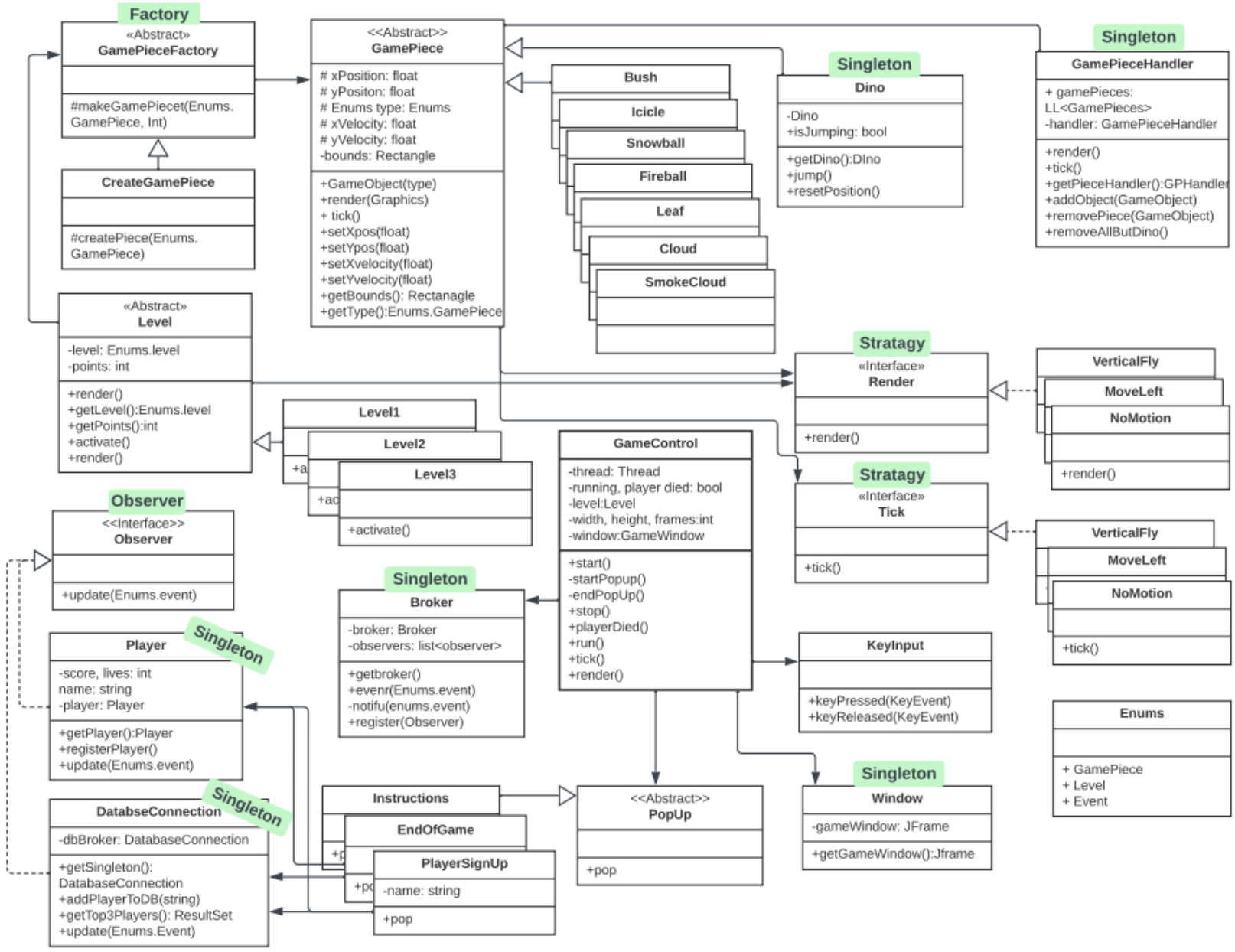
Project 6
The Last Dino
Priya Patel and Heather Monteson

1. **Status Summary**
    a. Work Done: Majority of the initial setup completed including player sign up, instructions, level 1 game background and game pieces, and some user key input to control the game pieces.
        i. Backend game setup (Heather)
        ii. Adding graphics to game (Priya)
        iii. Key Input (Priya and Heather)
        iv. Database setup (Heather)

    b. Changes and Issues
        i. Getting things to run across both our systems has been challenging but we have, as of now, resolved all issues.
        ii. Working with threading and understanding how to jump from one level to the next has been challenging but we should be able to resolve issues by the next iteration.
        iii. No major changes

    c. Patterns
        i. Factory: We are using the factory pattern for the creation of all of our game pieces (clouds, leaves, fireballs, snowballs, bushes and icicles) and using it as a means for also encapsulating an algorithm to place the game pieces at random locations with minimum set spacing between them as they are created.
        ii. Singleton: There are a number of singletons in use so that we can easily access data and shared objects and avoid accidental duplications of objects within the thread.
        iii. Observer: Observer is being used to track key events that are affecting data, particularly player lives left and score. A Player object and DatabaseConnection object wait for updates from the systems Broker object and adjust the player data and database values accordingly.
        iv. Strategy: Implementing in the next iteration to isolate tick and render behavior that is used across classes.

2. Class diagram



3. **Plan for next iteration: detailed list of tasks to complete broken down by class**

-Tick

-implement strategy pattern

-Render

-implement strategy pattern

-DatabaseConnection:

- Add update SQL statements to update values in the database as player gets points/loses lives

-Must store both the name and the id generated to up

-GameController:

-Handel stopping game if player dies

-Switch levels (currently handled in the stop method of GC but need to determine how/when to call. There is an operation set up but unsure if it will work.

-GamePiece:

-All: graphics, tweaking tick/motion from key input

-Obstacles/leaves:

- Randomize positions and when they appear in the window
- Set up rectangle bounds for collisions
- Verify velocity/movement once everything is updated

-Dino:

- Code jumping: image jumps, trigger though key input
- Code ducking: image ducks, trigger though key input
- Code standing: imagestands after ducking, trigger though key input back to running.
- Rectangle bounds for collisions

-Background pieces:

- Randomize positions as they move across the screen

-Player:

- create player score and lives lost graphics

-EndOFGame:

-Create Display

-Get player info from player to display

-Get top 3 players info from database to display (query set and working)

-Make is so that the window will still display player stats even if there is not a connection