Ping Pong Summary

* We took the base class of a Bluetooth card game and from a chat application and modified to meet our functionality.
* We used the same game logic from the initial libGDX version and a lot of trial and error method to achieve the non-libGDX version.
* In fact, libGDX was easier to work with.

Regarding the classes

* Ball
* Size
* Position, Velocity getters and setters.
* Collision handlers
* Paddle
* Height and width
* Position getters and setters
* Move handler
* Bluetooth Fragment
* Handles the Bluetooth connection user interface part and discovery & connection of devices
* ArrayList – is a dynamic data structure in which you can add or remove any number of elements. **Important Note:**In ArrayList, the items to be added are only of [object](https://abhiandroid.com/java/class-objects) type. No [primitive data types](https://abhiandroid.com/java/primitive-data-types-in-java) (i.e. int, char etc) are allowed to insert in ArrayList.
* ArrayAdapter - In android, An [adapter](https://abhiandroid.com/ui/adapter/) is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to [adapter](https://abhiandroid.com/ui/adapter/) view then view can takes the data from the [adapter](https://abhiandroid.com/ui/adapter/) view and shows the data on different views like [listview](https://abhiandroid.com/ui/listview/" \o "ListView" \t "_self), [gridview](https://abhiandroid.com/ui/gridview/" \o "GridView" \t "_self), [spinner](https://abhiandroid.com/ui/spinner/) etc. ArrayAdapter is more simple and commonly used Adapter in android. Whenever you have a list of single type of items which is backed by an array, you can use ArrayAdapter. For instance, list of phone contacts, countries or names. By default, ArrayAdapter expects a Layout with a single [TextView](https://abhiandroid.com/ui/textview/" \o "TextView" \t "_self), If you want to use more complex views means more customization in grid items or list items, please avoid ArrayAdapter and use custom adapters.
* BroadcastReciever - Base class for code that receives and handles broadcast intents sent by [Context.sendBroadcast(Intent)](https://developer.android.com/reference/android/content/Context#sendBroadcast(android.content.Intent)).
* AyncTask - <https://developer.android.com/reference/android/os/AsyncTask>
* BluetoothSocketListener
* Implements the runnable interface, all the instances are intended to be executed by a thread
* <https://developer.android.com/reference/android/bluetooth/BluetoothSocket>
* GameState
* Go through every function, most importantly the draw and run ones
* GameThread
* Runs concurrently with other threads
* <https://developer.android.com/reference/android/view/SurfaceHolder>