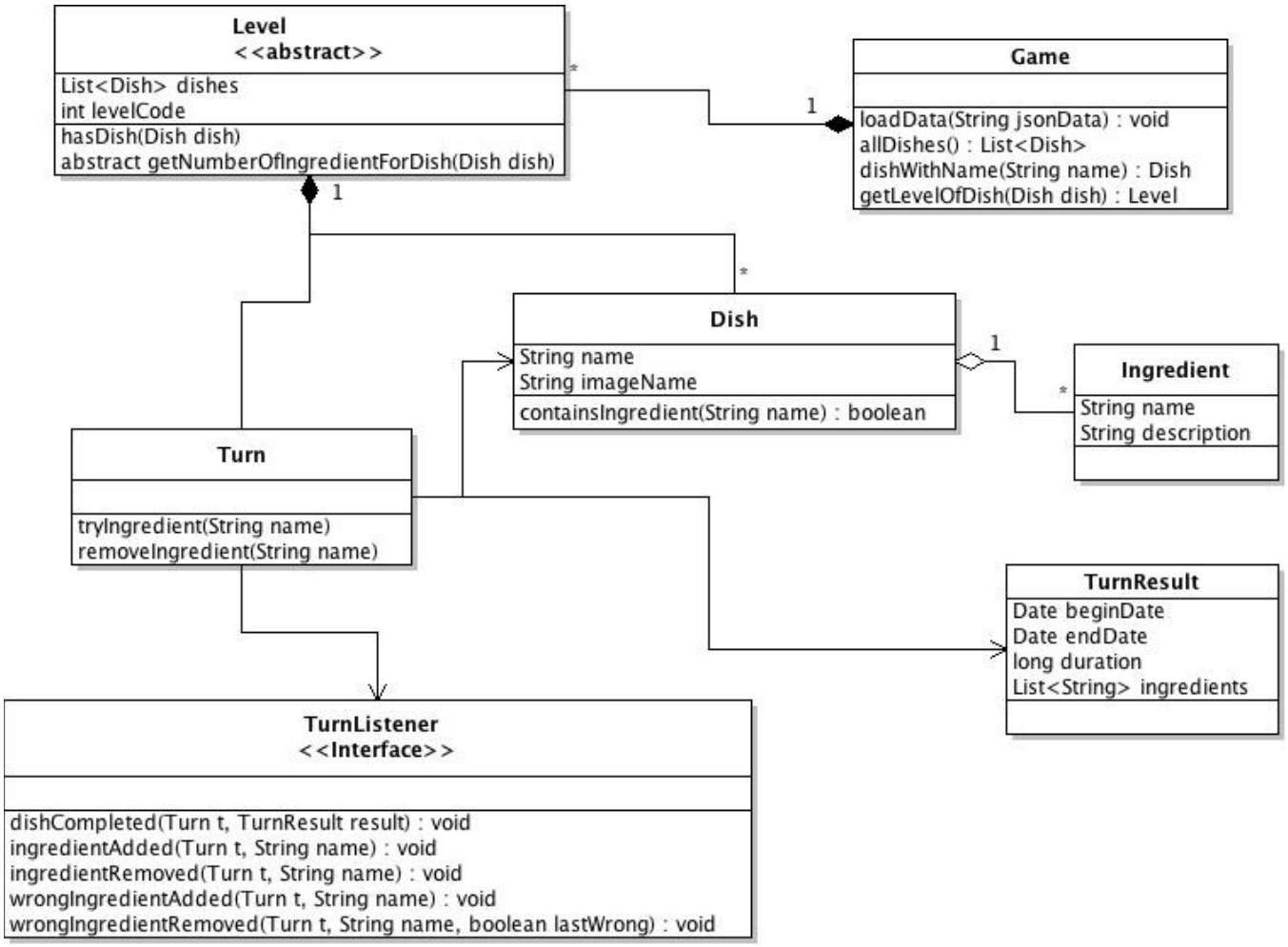
Design dell’applicazione

Diagramma UML del modello



Le classi del modello sono le seguenti:

- **Game**: classe statica che permette il caricamento dei dati di gioca da JSON e mantiene una referenza ai livelli e ai piatti
- **Level**: classe astratta, il metodo getNumberOfIngredientForDish ritorna il numero corretto di elementi solo per il Livello 1
- **Dish**: il piatto, un insieme di ingredienti
- **Ingredient**: l'ingrediente
- **Turn**: tiene traccia del completamento di un piatto, invia callback al relativo TurnListener. Al termine del piatto genera un TurnResult.
- **TurnListener**: interfaccia per la ricezioni di eventi relativi a un turno

- **TurnResult**: il risultato del turno

AllJoyn

La modalità multiplayer è stata realizzata attraverso AllJoyn. Qui si può trovare la documentazione completa: https://allseenalliance.org/developers/develop/api-guide/core/android.
L'applicazione espone un generico **MultiPlayerService** con il metodo *sendMessage().* Questo metodo permette di inviare semplici Signal con un parametro stringa.
Il formato di invio dei messaggi è CSV, il primo elemento è il tipo del messaggio.
Verranno inviati tre tipi di messaggi:

1. *dishes* seguito dai nomi del piatti, rappresenta l'inizio del gioco, l'host della partita invia al joiner della sessione i piatti selezionati
2. *stop* inviato quando uno dei due dispositivi abbondona il gioco (alla ricezione di questo messaggio si torna indietro alla schermata principale)
3. *dish* seguito dal nome del piatto, indica il completamento di un piatto

Gimbal

Nel gioco ciascun ingrediente è un trasmettitore Gimbal. Nell'applicazione si fa uso della versione 2 dell'API, in particolare si fa uso della parte di Beacon management. Qui la documentazione: https://gimbal.com/doc/android/v2/devguide.html#set_beacon
Tale API viene utilizzata nella MainActivity per orchestrare il flow del gioco. I beacon sono associati ad un ingrediente attraverso il *name.* La vicinanza di un beacon è determinata dal valore del suo RSSI. Il valore che discrimina la vicinanza di un beacon è -60.
Attenzione: potrebbe essere necessario cambiare il threshold di vicinanza (si trova nella classe Commons.java).

Activities

L'applicazione dispone di quattro Activity:

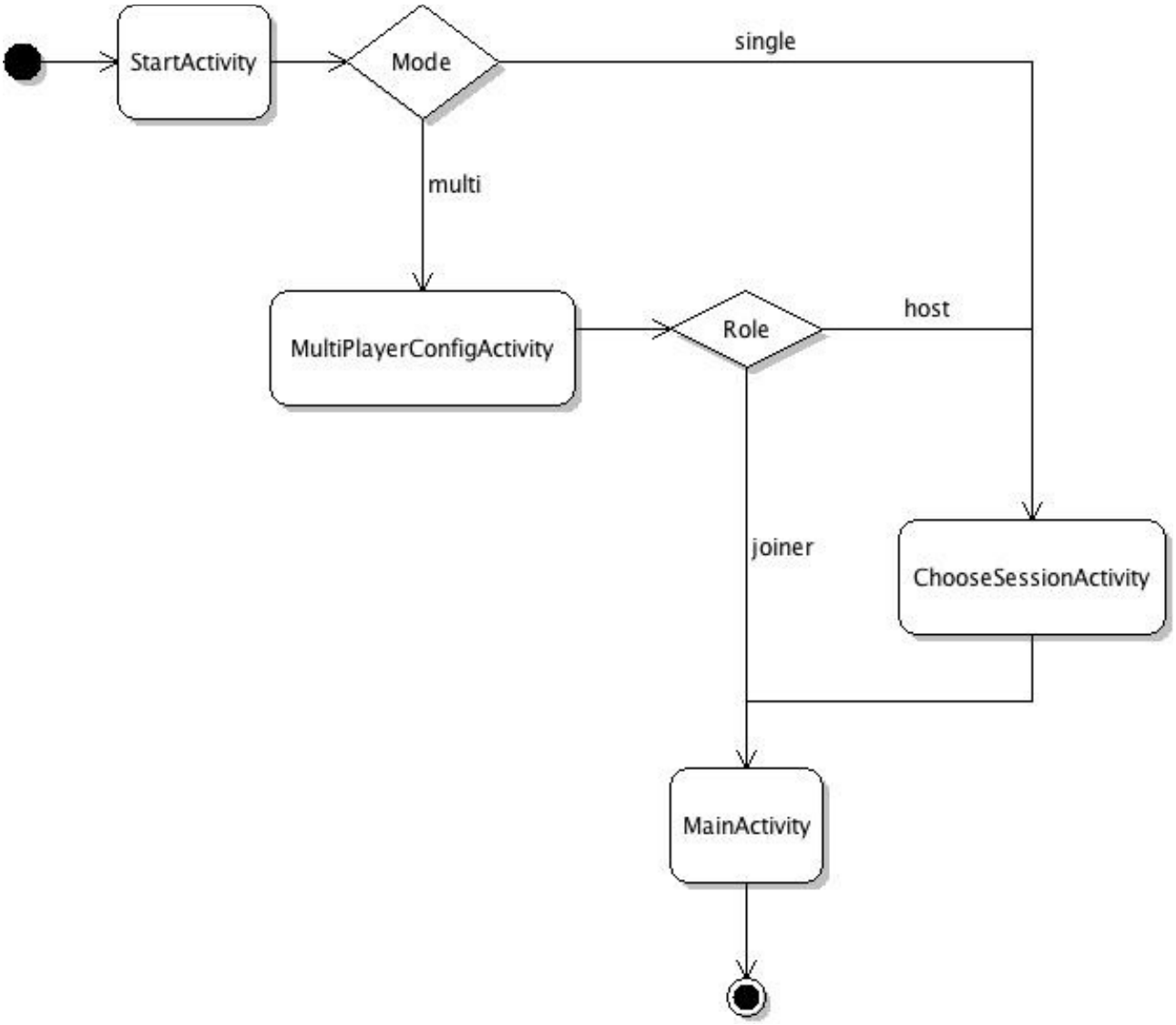
- **StartActivity**: mostra la schermata principale e carica i dati del gioco per evitare successivi caricamenti. Dispone di un bottone per l'inserimento della email per l'invio delle statistiche del gioco. La persistenza della email è affidata alle SharedPreferences di Android

- **ChooseSessionActivity**: mostra la schermata di scelta dei piatti, cerca nell'Intent un Extra "mode" che indica la modalità di gioco (singola o multiplayer). La selezione dei piatti può avvenire in due modi: selezionandoli da una GridView o premendo il tasto "Casuale"

- **MultiPlayerConfigActivity**: gestisce parte dall'inizializzazione della logica di AllJoyn, il comportamento varia a seconda se il giocatore seleziona di creare una partita (ruolo host) o di accedere ad una partita (ruolo joiner)

- **MainActivity**: orchestra la partita usando Gimbal API, AllJoyn Framework, la classe Turn e TurnListener. Cerca nell'Intent un Extra "mode" per la modalità e uno "dishes" per i piatti.

Di seguito viene riportato il flow di accesso alle diverse activity.



Durante una partita il tasto Back di Android riporta alla schermata iniziale dopo una conferma. Nel caso di Multiplayer se un dispositivo abbandona la partita anche l'altro torna alla schermata principale.

Adapters

Per visualizzare liste di oggetti sono stati usati i seguenti Adapter:

- **DishAdapter**: viene utilizzato nella schermata di selezione dei piatti, tiene traccia degli elementi selezionati
- **IngredientAdapter**: controlla la visualizzazione delle interazioni tra ingredienti e dispositivo. Mostra immagini diverse a seconda dell'assenza di ingredienti, ingredienti presenti o errati
- **TableAdapter**: controlla la visualizzazione della tavola con i sottopiatti vuoti o riempiti da quelli completati dalla squadra relativa al dispositivo e anche dalla squadra avversaria in caso di multiplayer.

Altre classi e file

- **Commons**: contiene costanti, della logica condivisa durante il gioco (AllJoyn) e metodi statici (invio mail)
- **GMailSender e JSSEProvider:** classi helper per l'invio di email
- **PrefUtils**: classe helper per la gestione delle SharedPreferences
- **/raw/data.json:** file contente i dati del gioco (ingredienti, livelli, piatti)
- **/raw/data_source.csv:** file originale in csv in cui erano contenuti i dati del gioco non ben strutturati
- **/raw/gen_json.py:** script python per trasformare *data_source.csv* in *data.json*