

Compétences devant être maîtrisées le 28 mars 2021

DOCUMENTATION

Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert.		
Je sais concevoir et décrire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application.		
Je sais concevoir un diagramme UML de qualité représentant mon application.		
Je sais décrire mon diagramme UML en mettant en valeur et en justifiant les éléments essentiels.		

CODE

Je sais utiliser les Intent pour faire communiquer deux activités.		
Je sais développer en utilisant le SDK le plus bas possible.		
Je sais distinguer mes ressources en utilisant les qualifier		
Je sais faire des vues xml en utilisant layouts et composants adéquats		
Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les événements		
Je sais coder une application en ayant un véritable métier		
Je sais parfaitement séparer vue et modèle		
Je maîtrise le cycle de vie de mon application		
Je sais utiliser le findViewById à bon escient		
Je sais gérer les permissions dynamiques de mon application		
Je sais gérer la persistance légère de mon application		
Je sais gérer la persistance profonde de mon application		
Je sais afficher une collection de données		
Je sais coder mon propre adaptateur		
Je maîtrise l'usage des fragments		
Je maîtrise l'utilisation de Git		

APPLICATION

Je sais développer une application sans utiliser de librairies externe		
Je sais développer une application publiable sur le store.		
et Je sais utiliser le bluetooth et faire communiquer mon application avec un autre téléphone		
ou Je sais utiliser la caméra		
ou Je sais utiliser Firebase		
ou Je sais consommer un service web		
ou Je sais utiliser le GPS		
ou Je sais utiliser l'accéléromètre et ou le gyroscope		

ou	Je sais utiliser le détecteur de proximité		
ou	Je sais utiliser toute autre contrainte validée par l'enseignant		

Je sais utiliser les Intent pour faire communiquer deux activités.

```

62         Intent intent = new Intent( packageContext: this, JoinActivity.class);
63         intent.putExtra( name: "playerName",name);
64
65         updatePreferencesForName(name);
66
67         startActivity(intent);
68     }

```

Je sais distinguer mes ressources en utilisant les qualifier

```

<? string.xml (fr)
<? strings.xml

```

Je maîtrise le cycle de vie de mon application

```

34     @Override
35     protected void onStop() {
36         super.onStop();
37         //FirebaseDatabaseHelper.removePlayer(player, player.getCurrentRoom().getRoomCode());
38         Intent intent = new Intent(getApplicationContext(), ActivitePrincipale.class);
39         startActivity(intent);
40     }

```

(la ligne en commentaire sera remplacée par les méthodes nécessaires à déconnecter un joueur du salon, ça n'a pas été fait par manque de temps)

Je sais gérer la persistance légère de mon application -> Non traité (on a bloqué la rotation)

Je sais gérer la persistance profonde de mon application

```
69 @ private void updatePreferencesForName(String name){
70     preferences.edit().putString("fr.iut.cours.name", name.isEmpty()?"Incognito":name).apply();
71 }
```

Je sais afficher une collection de données/ Je sais coder mon propre adaptateur

```
1 package iut.projet.view;
2
3 import ...
4
14 public class AdaptateurRoomList extends RecyclerView.Adapter {
15     private List<Player> playerList;
16
17     public AdaptateurRoomList(List<Player> liste) { this.playerList = liste; }
18
21
22     @NonNull
23     @Override
24     public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
25         LinearLayout ll = (LinearLayout) LayoutInflater.from(parent.getContext()).inflate(R.layout.player_list_cell, parent, attachToRoot: false);
26         return new RoomListViewHolder(ll);
27     }
28
29     @Override
30     public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
31         Player p = playerList.get(position);
32         ((RoomListViewHolder)holder).changeReadyDot(p.isReady());
33         ((RoomListViewHolder)holder).getName().setText(p.getPlayerName());
34     }
35
36
37     @Override
38     public int getItemCount() { return playerList.size(); }
39
41
42     public void updatePlayers(List<Player> players){
43         this.playerList = players;
44         notifyDataSetChanged();
45     }
46 }
47
```

Je maîtrise l'usage des fragments

```
if(getIntent().getStringExtra( name: "roomCode")!=null){
    hostFragment = new HostFragment(getIntent().getStringExtra( name: "playerName"),getIntent().getStringExtra( name: "roomCode"));
}
else {
    hostFragment = new HostFragment(getIntent().getStringExtra( name: "playerName"));
}
if(fragment == null){
    getSupportFragmentManager().beginTransaction().replace(R.id.fragmentContainer, hostFragment, tag: null).commit();
}
}
```

