

NOM CUESTA

Prénom Quentin

Promo 2017 IL M1

Date 14/04/16



	1	3	5
--	---	---	---



CUESTA Quentin  
M1 - 2015

## MATIÈRE Développement Mobile

Android :

1,5/3

Android  
iOS

6,5/10  
7/10

- Question 1: Nous devons créer un broadcaster avec l'intent "com.twitter.tweet" pour pouvoir envoyer une paire (key, value) à l'application Twitter. "message" sera la clé de l'envoie. Nous supposons que Twitter possède un ~~listener~~ broadcast receiver pour attraper l'envoie de données avec la clé "message".

- Question 2:

< TextView :

```
< android:id = "lastTweet" />  
< android:content = "" />  
< android:width = "match_parent" />  
< android:height = "match_parent" />  
/>
```

filter.

0,8

```
String lastTweet = # receiveLastTweet();
```

```
TextView t = (TextView) findViewById(R.id.lastTweet);  
t.setText(lastTweet);
```

- Question 3: Nous ne pouvons pas faire cela car les tweets sont broadcasté dans le téléphone par l'application Twitter.

N'importe quelle application ayant un broadcast listener peut récupérer les tweets. C'est comme la radio.

- Question 4 :

3/5

```
for (int i=0; i<k; i++) {
```

```
    buffer = U0
```

A 

```
U1 = MyRule. computations (buffer U0, U1);
```

```
    U0 = buffer;
```

```
}
```

```
int result = U4;
```

- Question 5 :

~~Il y a une autre erreur pour ce qui est de la couleur mais il est quasi identique aux deux autres, il n'y a que l'id qui change.~~

< EditText :

```
< android:id = "U0" />
```

```
< android:width = "match-parent" />
```

```
< android:height = "match-parent" />
```

⇒ < android:hint = "U<sub>0</sub>" />

```
/>
```

A < EditText :

```
< android:id = "U1" />
```

```
< android:width = "match-parent" />
```

```
< android:height = "match-parent" />
```

```
< android:hint = "U1" />
```

```
/>
```

< Button :

```
< android:id = "compute" />
```

```
< android:width = "match-parent" />
```

```
< android:height = "match-parent" />
```

```
< android:name = "compute" />
```

/>

- Question 6: Nous ne pouvons pas le mettre dans un Button car l'algorithme s'exécute dans une tâche Asynchrone. Ce qui veut dire qu'à chaque fois que l'utilisateur va appuyer sur le bouton nous aurons une nouvelle tâche asynchrone. Déjà que les tâches peuvent être longues alors si nous en lancerons beaucoup notre application va se faire kill en mémoire car elle en utilisera trop.

- Question 7:

public class asyntCompute (~~int[3] Class~~, MyRule.computeUn(<sup>3</sup><sub>R</sub>)

extends

AsyncTask <

int k, U0, U1 = 0;  
<sup>3</sup>  
~~int [3] U1 = null;~~  
 int R = 0;

public void preExecute () {

public void update () {

public void OnExecute () {

int buffer = 0;

for (int i = 0; i < k; i++) {

buffer = U1;

U1 = MyRule.computeUn(U0, U1);

U0 = buffer;

}

R = U1

&

public void postExecute () {

removeResult(R);

public asyntCompute () {

k = U[0]; U0 = U[1]; U1 = U[2];

{ }

U : données passées en paramètres (données d'entrée)

V : action à effectuer ) type ??

W : variable à retourner.

Q. Question 8 : On peut lui envoyer une notification.  
~~

AP. Question 9 : C'est une application de réception de SMS. La classe étend `IntentReceiver` pour écouter grâce au `Intent-filter` la réception des SMS. Lorsque que cette application reçoit un SMS, elle reçoit un message grâce à un `buf` qui est un `StringBuilder` pour afficher le SMS à l'utilisateur.

Pos :

Q. Question 1 : Character et inventory seront des classes car nous pouvons trouver tout un tas de fonctions à leur ajouter. Les structures ne contiennent que des attributs.

class  
public character () {

var Character (

let pseudo, sex, age, type;

var HP, MP, location,

var inventory = new Inventory(); } ; 1

var Inventory (

var object: [int, Object]; );

NOM CUESTA

Prénom Quentin

Promo 2017 1L M1

Date 16/04/16

## MATIÈRE Développement Mobile

Question 2:

```
func initNM(ps: String, s: String, a: Int, t: String) {  
    let pseudo: String? = ps: "John Doe";  
    let sex: String? = s: "Unknown";  
    let age: Int? = a: 18;  
    let type: String? = t: "warrior";  
}
```

1,5

```
func initM(hp: Int, mp: Int, l: [Int]) {  
    func initM(l: [Int]) {  
        var HP = 20;  
        MP = 10;  
        n = l[0];  
        y = l[1];  
        inventory = "empty";  
    }  
}
```

• Question 3:

pseudo :	[ ]	[ ]
sex :	[ ]	[ ]
age :	[ ]	[ ]
type :	[ ]	[ ]
HP :	[ ]	[ ]
MP :	[ ]	[ ]
location	[ ]	[ ]
Inventories.		
[ ]	[ ]	[ ]
[ ]		

0/25

0/100

• Question 5:

```
func smallHPotion () {  
    switch character.HP  
    case 1 ... 19  
        character.HP = 20;  
    case 20 ... 29  
        character.HP = 30;  
    case 30 ... 39  
        character.HP = 40;  
    case 40 ... 49  
        character.HP = 50;  
    ...  
    case 90 ... 99  
        character.HP = 100;  
    case 0  
        return;  
}
```

Oh

1,5

modif character.modifyInventory ()?  
{ return;

• Question 6

```
func recoverHP (way: string) {
```

```
    if way == "small HP" {  
        smallHPotion();
```

```
    if way == "full HP." || way == "fountain"  
        character.HP = 100;
```

```
}
```

(fountain)

1,75

• Question 7

```
func pickItem (items: [object]) {
```

```
    let item: object? = items[0];
```

```
    if !item
```

```
        print("There is nothing interesting here, Dude");  
    else
```

```
        character.inventory.add(item);
```

```
}
```

2

