

RPGManager

Character sheet manager for role-playing games.

Description

This application is used for table-top role playing games, like Dungeons & Dragons or Pathfinder. The goal is to make each player use the application for his character. The main activity of the app is the character sheet. It is used like a character sheet in real table-top RPGs. This application also allow the user to throw a dice, with dices between a 4-faces to a 30-faces dice. This application interacts with a server through JSON and gets all the data from the website where the game master leads the game from.

Name David Schirduan		Look	
Misc Note: -Still don't know why Lauren puts up with me...gotta check that quest journal -Lost the reward from my last quest. -I have a kitten!		Black, Curly Hair Comfortable Clothes Sandals Tall, Lanky Body	
Assign these scores to your stats: 16(+2), 15(+1), 13(+1), 12(+0), 9 (+0), 8(-1)			
Strength 9 ■ weak -1 +0 str	Dexterity 12 ■ Shaky - +0 dex	Constitution 8 ■ Sick - -1 con	Intelligence 13 ■ Stunned -1 +1 int
Wisdom 15 ■ Confused -1 +1 Wis	Charisma 16 ■ scarred -1 +2 cha	Damage D6	
Armor -1		HP	
Alignment <input type="checkbox"/> Facilitator: Use your charisma and technological skill to bring people together and accomplish great things (for great justice!)		Starting Moves The Dave Abides Whenever life gets stressful, put on some James Taylor, make yourself a cup of Pu-er Tea, lean back, and take it easy. While you are in this state, you gain +1 to all Resist Stress rolls. Sportball When you absolutely MUST go outside, Roll + Dex, and engage in one of the following physical activities: Kayaking, Racquetball, Jogging around Folly Beach Vidja-ing Whenever you want to relax, but you don't want to fall asleep, break out the laptop/console and pick a game from the following genres: <ul style="list-style-type: none">MetroidvaniaPlatformerRPGRoguelike Social Buffalo When you get sick of being alone, harass your friends and try to start up a small group to hangout and play games. Roll + Cha. <ul style="list-style-type: none">On a 10+, you succeed, and get a gnarly group of people to hangout and enrich your lifeOn a 7-9, you do get a group of people, but WAY more than you could handle or keep track ofOn a 6-, you over-plan, stress out, and end up frustrated	
Background <input type="checkbox"/> -Raised in Charleston, SC for most of life. -Grew up with an interest in science -Discovered love for computers and Linux in high school -Created strong community groups in college -Survived classes while working 2 part-time jobs in technology fields -After graduation, discovered passion for tabletop games			
Gear Lenovo Yoga (1/2 Weight, Portable, 6 Charges per Day) GM Toolkit (5 uses, 1 weight) Bag of Programming Books (5 uses, 1 weight, +1 Int when used) Shirt with ironic/video game reference (0 armor, -1 Cha) Battered Jeep (clumsy, awkward, 3 capacity, 3 armor)			
David Schirduan		Level 23 XP XXXX	

Dungeon World character sheet

Activities

Character Sheet Activity

This activity is used to display the specifications of the character chosen in the options. All the information displayed is taken from the website. The main information (nickname, alignment, race and class) about the character are displayed on top. Then are the health, the dice of attack and the defense of the character. The characteristics (Level, Experience, Strength, Dexterity, Constitution, Intelligence, Wisdom and Charisma) come next and are editable by the user through a NumberPicker which is displayed when clicking on the characteristic. Then, there are the bonds, gears and moves of the character.

LTE

73 %

16:09

Character Sheet

Nickname

foo

Alignment

Neutral

Race

Human

Class

Rogue

13/20

d4

2

LVL

7(43)

STR

6

DEX

28

CON

6

INT

99

WIS

5

CHA

10

Bonds

bar, from north tavern.

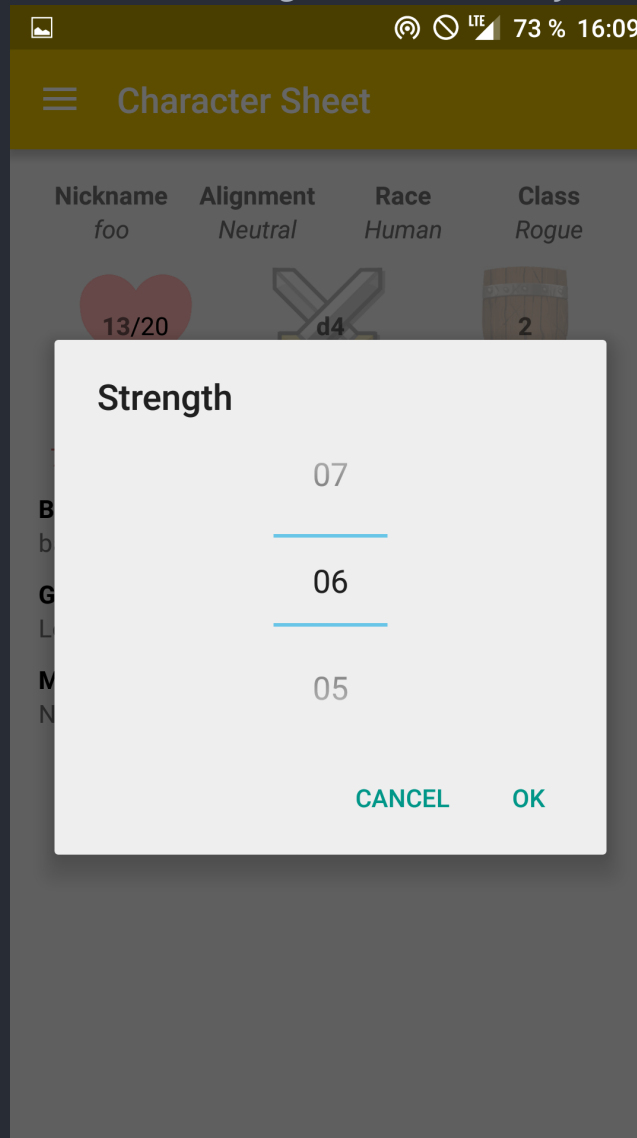
Gear

Long Sword, Wooden Shield.

Moves

Not so strong attack

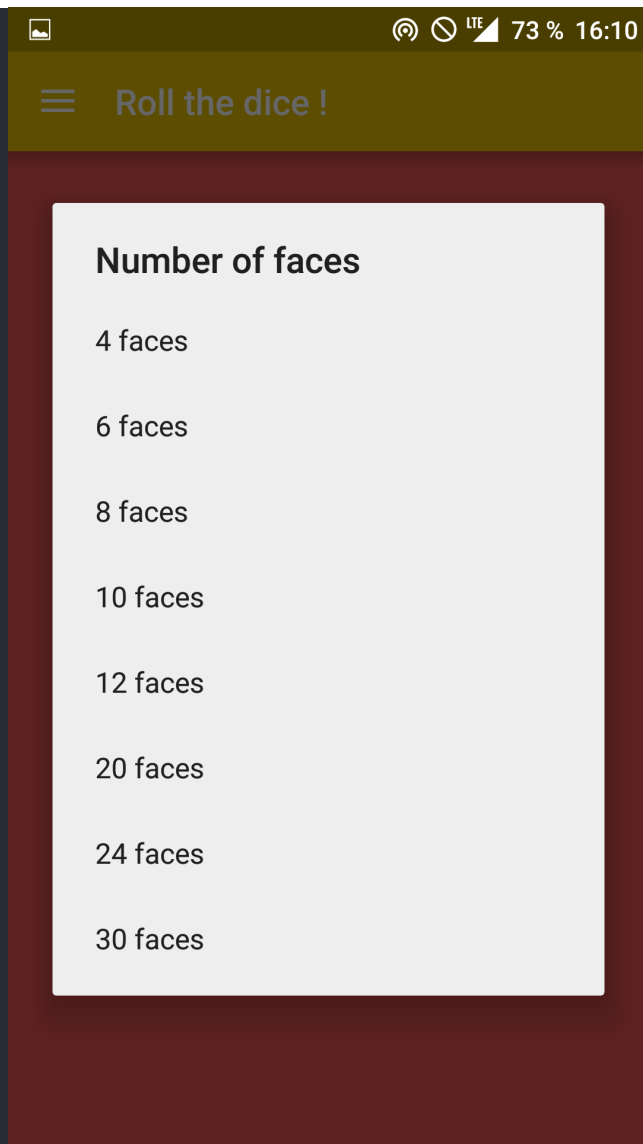
RPGManager main activity



The user can modify the value of his stats

Dice Activity

This activity is just a button with a dice image on it. When clicking the button, a pop-up appears and the user is able to change the number of faces of the dice. In this activity, we are able to shake the device to launch a dice a gives a random number.



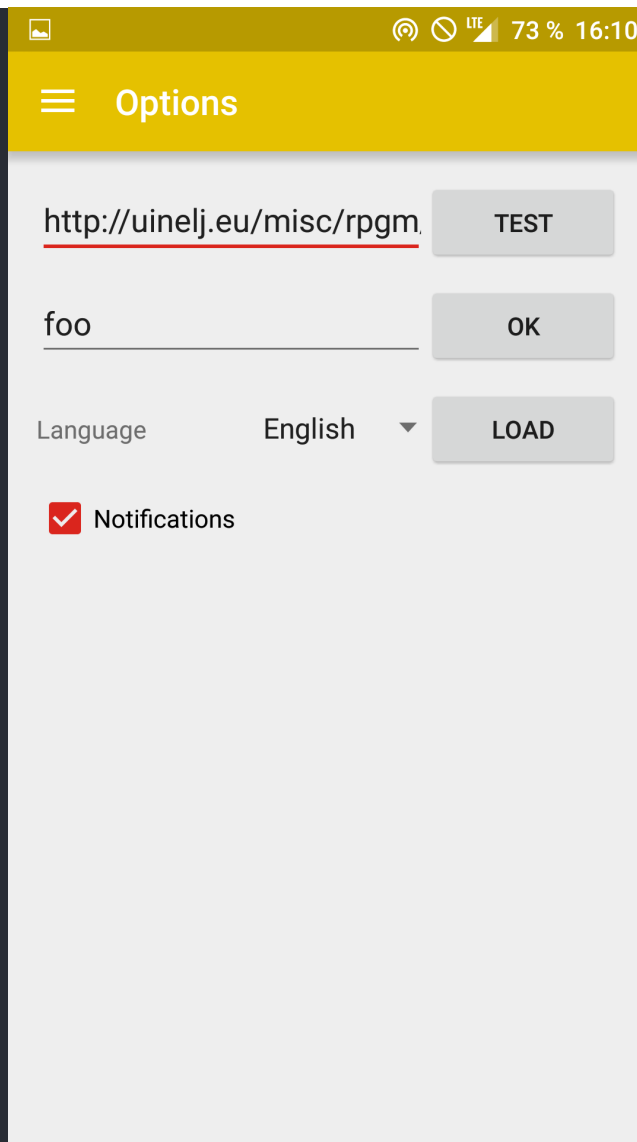
The user can choose what dice he has to roll



After shaking the device, we can see the roll on the screen

Options Activity

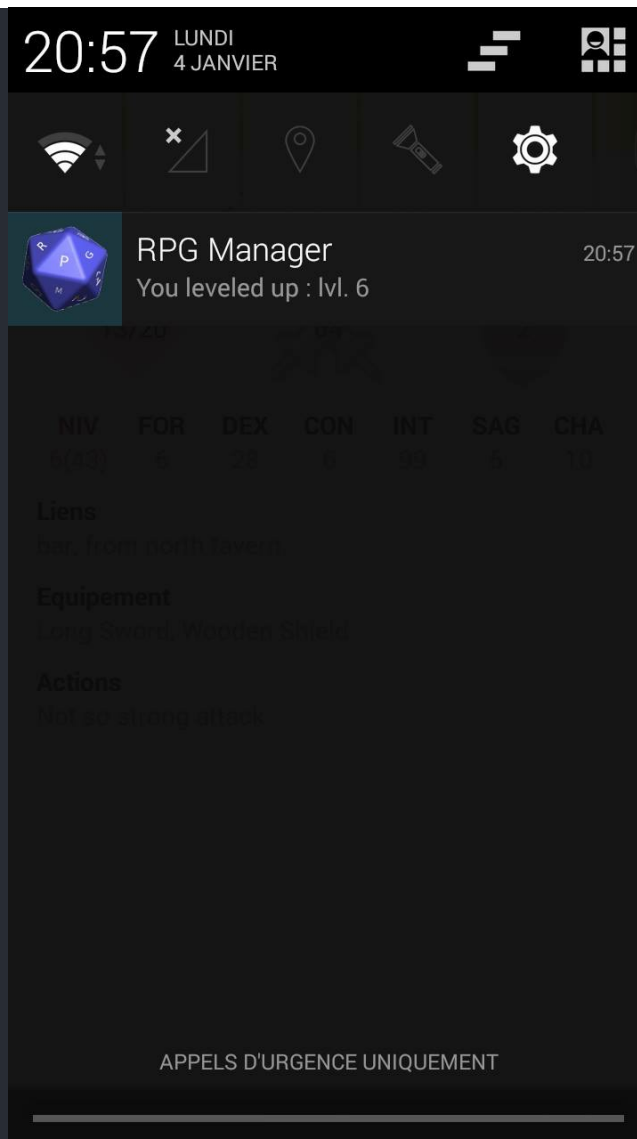
There are 4 options: the URL of the server to connect, the nickname to load when refreshing the data, the language to select (English and French), and a check box to enable or disable notifications. Upon testing the URL, a message appears if the destination is reachable. When changing the language, the button reloads the current activity to charge the newly selected language.



Here we can configure the server location, the nickname of the player, and some other things

Background Services

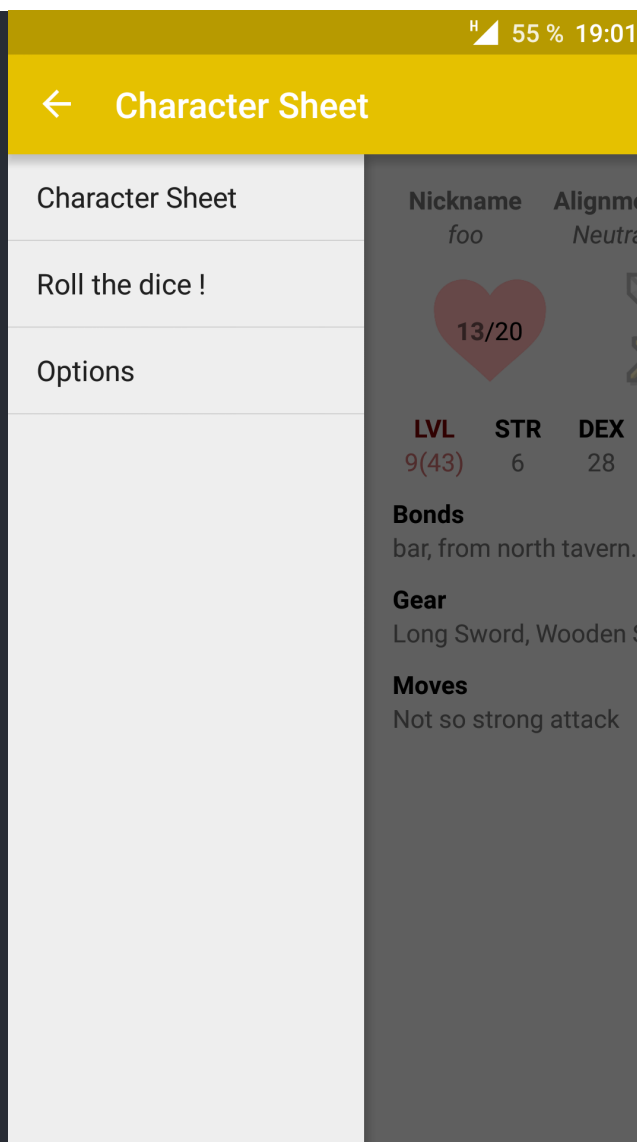
The background service we are using is simple: every 30 seconds, the level and the health of the character are checked from the server. If the health came down to 0 during the 30 seconds, a notification pops up. And if the character leveled up during the 30 seconds, another notification pops up. Before getting the data from the server, the service looks up for the notification option in the Option Activity to see if the user decided to stop receiving notifications. This service stops as soon as the notification check box is unchecked and restarts when it is checked again.



The notifications tells the user when he dies or levels up

Drawer

To switch between applications, we used a navigation drawer. This menu is declared in the DrawerActivity (which extends AppCompatActivity) and each of our activity extends this DrawerActivity. Each of these activities are declared with `android:launchMode="singleTask"` and are called via the intent flag `Intent.FLAG_ACTIVITY_REORDER_TO_FRONT`. To launch the background task, we use a category on the intent `notifIntent.addCategory(NotificationService.TAG);` (`NotificationService.TAG` being a public String with a specific name) to be able to call it from different activities. We first stop the service and then starts it so it doesn't launch twice.



The drawer provides an easy way to switch between the different activities

Sensor

In this application we use the accelerometer of the device as our sensor. It is used in order to detect, in the *Dice Activity*, when the user is shaking his device. When we detect that it's the case, we display the result of the dice roll.

Server communication

The application makes requests to a HTTP Server, who will answer by sending a JSON response. For example, if we want to get the

strength value of the player *foo*, the application will query the server with

`http://serverlocation/rpgm/action.php?a=get&id=foo&field=str`

and get this JSON answer :

```
{"success":true,"field":"str","value":"6"}
```

The server itself handles four actions :

- `createUser` by POST, in order to create a user
- `rmUser` by GET, in order to remove a user
- `get` by GET, in order to receive the stats of a user,
- `update` by GET, in order to change one of the user's values,