

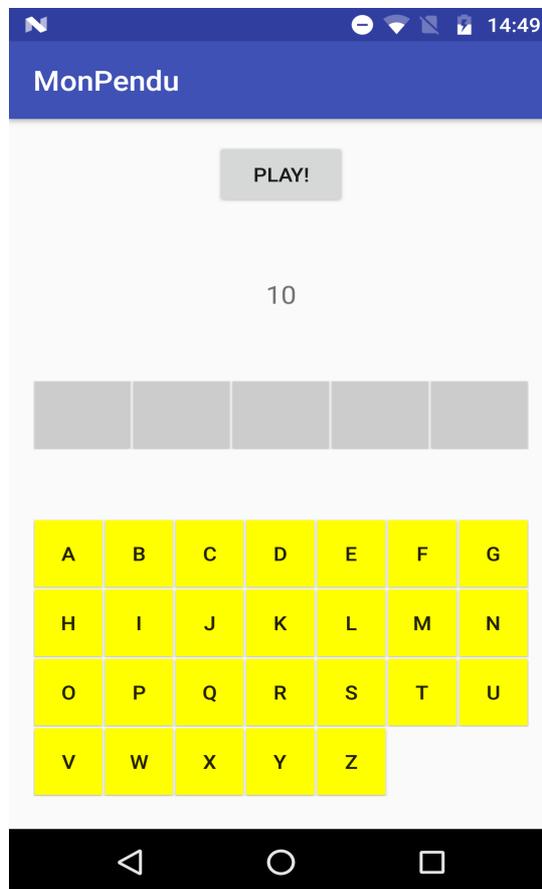
# Application MonPendu

Il s'agit de développer une application permettant de jouer au jeu du Pendu.

Rappel des règles :

- l'application choisit au hasard un mot dans un dictionnaire,
- le joueur a 10 essais pour deviner le mot,
- le joueur propose une lettre,
- si le mot ne contient pas la lettre proposée, le joueur perd un essai,
- sinon la lettre s'affiche pour toutes les positions où elle apparaît dans le mot.

Pour développer votre application, vous respecterez scrupuleusement les contraintes imposées, notamment au niveau de la construction de l'interface.



## PARTIE 1

Votre application sera composée d'un `LinearLayout` principal de type vertical. Dans ce `LinearLayout` vous placerez :

- un bouton centré permettant de lancer le jeu,
- une zone d'affichage centrée permettant d'afficher le nombre de coups restants (Indice : afin de centrer ces deux éléments , placez les chacun dans un `LinearLayout` de type horizontal dont vous aurez modifié l'attribut **gravity**),
- un `LinearLayout` de type horizontal qui sera rempli dynamiquement par votre application

- pour représenter les lettres du mot à trouver. Cet élément aura pour identifiant **affichage**, un `LinearLayout` de type vertical qui sera rempli dynamiquement par votre application pour afficher le clavier de saisie. Cet élément aura pour identifiant **clavier**.

## PARTIE 2

Au lancement de votre application, lors de l'appui sur le bouton **PLAY**, une requête sera effectuée vers :

<http://veron.univ-tln.fr/ANDROID/PENDU/pendu.php>

afin de récupérer au format JSON le mot que le joueur devra deviner. Votre application devra remplir alors la zone **affichage** avec autant de boutons qu'il y a de lettres dans le mot. Ces boutons occuperont tous l'espace disponible de façon équitable et seront tous délimités par une marge haute, basse, gauche et droite de 3 pixels. Vous utiliserez la constante **Color.LTGRAY** pour définir la couleur de fond de ces boutons. Ces boutons seront utilisés pour afficher les lettres proposées par l'utilisateur lorsqu'elles font partie du mot à trouver.

## PARTIE 3

Dans la zone **clavier**, votre application devra générer 4 `LinearLayout` de type horizontal. Dans les 3 premiers, 24 boutons seront ajoutés dynamiquement (8 par `LinearLayout`) afin de représenter les lettres de A à U. Pour chacun des 3 `LinearLayout`, les boutons occuperont tout l'espace disponible de façon équitable et seront tous délimités par une marge haute, basse, gauche et droite de 3 pixels. Un dernier `LinearLayout` contiendra uniquement 5 boutons visibles représentant les lettres de V à Z. Un écouteur sera placé sur chaque bouton, la gestion de cet écouteur sera effectuée par votre application principale et non pas localement par les boutons.

## PARTIE 4

Implémentez votre écouteur de façon à respecter les règles du jeu. Si le nombre d'essais atteint 0, un message d'échec devra s'afficher à la place du nombre d'essais. Si le mot est trouvé, un message de succès devra s'afficher à la place du nombre d'essais. Dans les deux cas, il ne doit alors plus être possible à l'utilisateur de saisir de nouvelles lettres au clavier (désactivation des écouteurs). Un appui sur le bouton **PLAY** doit permettre de réinitialiser le jeu et de sélectionner un nouveau mot à trouver.

## PARTIE 5

En utilisant les **Shared Preferences**, stockez le meilleur score (nombre minimal de coups) et affichez le à chaque lancement de l'application.