

1 Instructions générales

Créez un nouveau projet en langage Java, nommé selon le schéma “TP5a_numéro”, où “numéro” est votre numéro d’étudiant.

Assurez-vous que la “minSdk” soit bien 19 ! On peut soit choisir ce paramètre à la création du projet, soit le changer dans le `build.gradle` a posteriori. Un projet qui ne compile pas, ou qui ne peut pas être exécuté sur les appareils prêtés (Sdk = 19) recevra 0.

Une fois le projet terminé, exportez votre projet (File → Export → Export to Zip File), et rendez ce fichier sur Eprel.

La Section 2 constitue le sujet du TP. La Section 3 détaille la procédure de notation. On pourra s’y référer pendant la séance pour prendre connaissance des modalités d’évaluation.

2 Midterms racer

L’objectif de ce TP est de réaliser une implémentation du jeu **Midterms racer**, qui fait carton plein aux États-Unis à l’approche des élections de mi-mandat. L’état du jeu, au départ, est représenté en Figure 1.

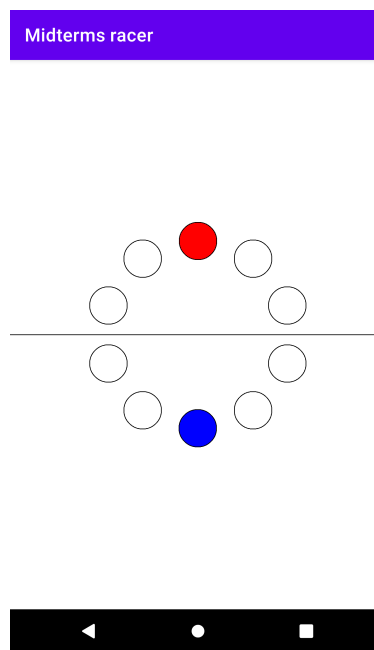


FIGURE 1 – Aperçu de l’écran de départ du jeu.

Dans **Midterms racer**, deux camps s’affrontent : les Républicains (en rouge) et les Démocrates (en bleu).

L’idée du jeu est la suivante : l’interface contient dix emplacements (représentés par des cercles), et deux palets, chacun dans un emplacement.

Par ailleurs, l’écran est divisé en deux parties : celle du haut appartient au camp républicain, et celle du bas au camp démocrate. Le palet de chaque camp commence dans l’emplacement central

de sa partie d'écran (cf. Figure 1).

Quand un joueur ou une joueuse clique sur sa partie d'écran, son palet avance d'un cran dans le sens anti-horaire. La Figure 2 représente l'état du jeu, quand, depuis la position initiale (Figure 1), les Républicains cliquent une fois dans leur partie d'écran (Figure 2a) puis les Démocrates cliquent trois fois dans la leur (Figure 2b).

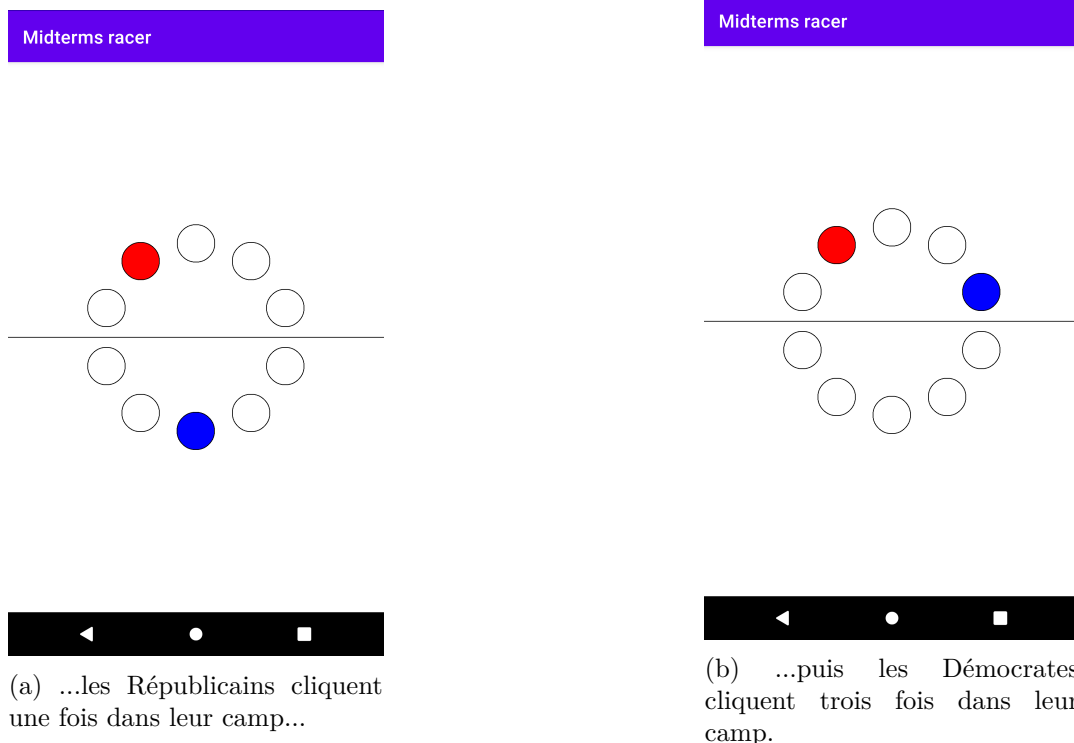


FIGURE 2 – Depuis la position initiale...

L'objectif du jeu est de rattraper le palet adverse. Par exemple, dans la situation décrite en Figure 3, les Démocrates gagnent s'ils arrivent à faire avancer leur palet d'un cran avant que les Républicains n'aient le temps de cliquer sur leur partie d'écran.

2.1 Création du `Plateau`

Question 1 Durant la partie, on cherche à éviter les rotations d'écran.

On ajoutera donc la ligne suivante dans le Manifest, comme attribut de `MainActivity` :

```
<android:screenOrientation=' 'portrait' '>
```

Question 2 Créez une class `Plateau` (vide pour l'instant) qui étend `View`.

On créera un nouveau `Plateau` dans `MainActivity::onCreate()`, qu'on déclarera comme unique élément de notre layout via `setContentView()`.

Question 3 Faites en sorte qu'un `Plateau`, en se dessinant, trace une ligne noire horizontale coupant l'écran en deux : ce sera la ligne de séparation entre les deux camps.

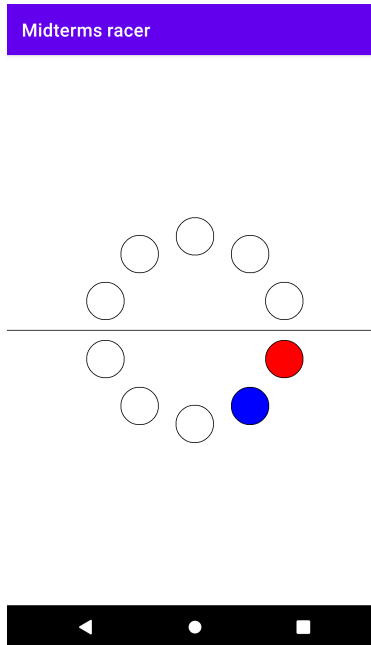


FIGURE 3 – Les Démocrates s’apprêtent à gagner : ils seront victorieux si leur palet est avancé avant que celui des Républicains ne prenne le large.

Question 4 Recopiez les méthodes `Plateau::computeCenters()` et `Plateau::computeRadius()` fournies ci-dessous. La première retourne un `float [10][2]`, qui contient la liste des positions (sous forme de tableau de taille 2) des centres des 10 emplacements, dans l’ordre précisé en Figure 4. La deuxième méthode retourne le rayon des emplacements.

À l’aide de ces méthodes, modifiez votre code pour qu’en se dessinant, un `Plateau` fasse apparaître les 10 emplacements, avec les bons centres et les bons rayons. On utilisera le même `Paint` que pour la Question 3. Pour rappel, les dimensions d’une `View` doivent être calculées dans `Plateau::onDraw()`, et pas dans le constructeur (puisque les dimensions ne sont pas encore fixées à ce moment-là).

```
private float [][] computeCenters(){
    float [][] centers = new float [10][2];
    float w = getWidth(), h = getHeight();
    float radius = Math.min(w,h)/4;
    float centerX = w/2, centerY = h/2;

    for(int i = 0; i<10; i++){
        double angle = Math.PI/2 + 2*i*Math.PI/10;
        centers[i][0] = (float) (centerX + Math.cos(angle)*radius);
        centers[i][1] = (float) (centerY - Math.sin(angle)*radius);
    }
    return centers;
}
```

```
private float computeRadius(){
    return Math.min(getWidth(), getHeight())/20;
}
```

Midterms racer

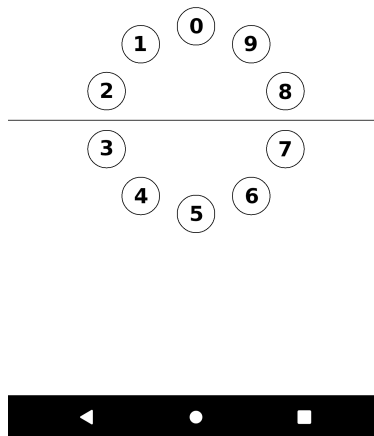


FIGURE 4 – Les positions des différents emplacements dans le tableau `centers`.

Question 5 Modifiez une dernière fois `Plateau` pour afficher les palets des Républicains (en `Color.RED`, dans l'emplacement 0) et des Démocrates (en `Color.BLUE`, dans l'emplacement 5). On créera pour cela deux nouveaux `Paint`.

2.2 Moteur de jeu

Il nous reste maintenant à faire en sorte qu'un clic sur l'écran fasse avancer le palet correspondant à la partie de l'écran qui a été touchée.

Question 6 Trouvez un moyen de retenir la position courante des deux palets.

Implémentez `Plateau::onTouchEvent()` pour qu'un `MotionEvent.ACTION_DOWN` sur la moitié supérieure de l'écran (respectivement, sur la moitié inférieure de l'écran) fasse avancer le palet rouge (respectivement, le palet bleu) d'un cran dans le sens inverse des aiguilles d'une montre.

Pour rappel, un exemple du comportement attendu est précisé en Figure 2.

Question 7 Pour finir, vérifiez la condition de victoire. Si le mouvement d'un palet le fait arriver sur le même emplacement que l'autre palet, on veut afficher un `Toast` "Les Républicains contrôlent le Sénat !" ou "Les Démocrates contrôlent le Sénat !" (en fonction du camp victorieux), et positionner chaque palet sur son emplacement initial.

Par exemple, la situation qui suit une victoire des Démocrates est représentée en Figure 5.

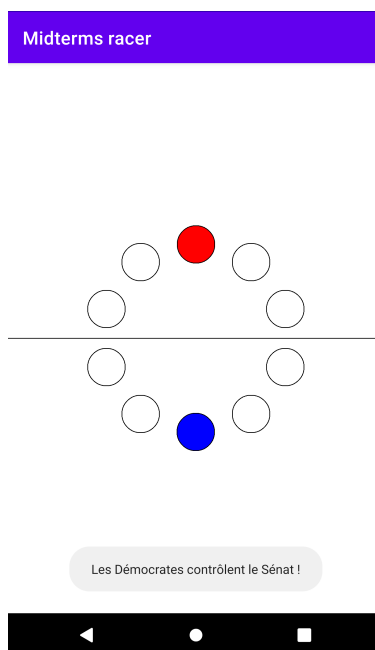


FIGURE 5 – Après une victoire des Démocrates, un `Toast` s’affiche et le jeu repart à zéro.

3 Notation

Au moment de corriger le TP de votre camarade, téléchargez le fichier zip qui vous a été attribué sur Eprel, et importez le projet dans Android Studio (`File` → `New` → `Import Project`). Merci de ne pas chercher à savoir à laquelle ou auquel de vos camarades appartient le numéro d’étudiant présent dans le nom du projet.

Pour la notation, on se contentera de tester l’application sur un appareil, sans en regarder le code. Un code qui ne compile recevra une note de 0. La liste des éléments à vérifier sur l’application que vous avez à noter est donnée en Figure 6. Pour chaque critère, on attribuera :

- 0, si le critère n’est pas du tout respecté.
- La moitié des points si le critère est partiellement respecté.
- L’intégralité des points si le critère est parfaitement respecté.

Critère	Note
Au lancement...	
...l'écran est séparé en deux parts égales par une ligne noire horizontale	2
...les dix emplacements sont bien tracés en noir à l'écran	5
...un palet rouge est situé dans l'emplacement le plus en haut	2
...un palet bleu est situé dans l'emplacement le plus en bas	2
Après deux ou trois clics sur la moitié supérieure de l'écran...	
...chaque clic fait avancer le palet rouge d'un emplacement, dans le sens anti-horaire	2
En cliquant ensuite deux ou trois fois sur la moitié inférieure de l'écran...	
...chaque clic fait avancer le palet bleu d'un emplacement, dans le sens anti-horaire	2
En cliquant alternativement sur les deux palets...	
...on arrive à faire plusieurs tours de piste	2
Quand le palet rouge rattrape le palet bleu...	
...le message "Les Républicains ont gagné le Sénat !" ou le message "Les Républicains contrôlent le Sénat !" apparaît quelques secondes à l'écran	1
...les palets retrouvent leur position initiale	1
Quand le palet bleu rattrape le palet rouge...	
...le message "Les Démocrates ont gagné le Sénat !" ou le message "Les Démocrates contrôlent le Sénat !" apparaît quelques secondes à l'écran	1

FIGURE 6 – Liste des critères à vérifier, et nombre de points associés.