

Interaction Homme-Machine : Multimodalité et Mobilité

Durée : 1 heure. Documents autorisés : Support de cours et notes de cours autorisés. Le barème mentionné est indicatif.

QUESTION 1 JEU TIC-TAC-TOE MULTIMODAL 6 points

Nous considérons la conception d'interaction multimodale pour le jeu tic-tac-toe sur un téléphone mobile (Figure 1) ou une tablette. Les joueurs posent tour à tour un rond, pour l'un, une croix, pour l'autre, dans une grille de 3x3 cases. Le but du jeu est d'obtenir un alignement (en ligne, colonne ou diagonale) de ses trois signes. La tâche interactive de ce jeu est donc de sélectionner un rond ou une croix et une case.



Figure 1 : Jeu tic-tac-toe
(<https://dribbble.com/shots/1710515-Tic-Tac-Toe>)



Figure 2 : Lever pour parler
"raise-to-talk"
(<https://dl.acm.org/cms/attachment/ad4b984c-08f0-4aad-a3c9-a19c31275836/f8.html>)

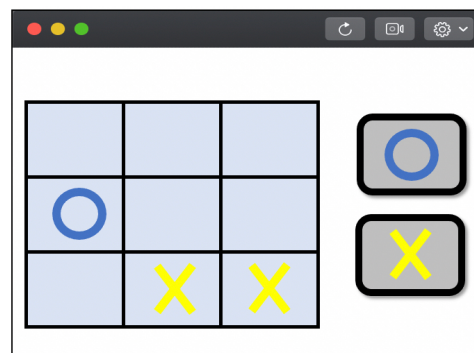


Figure 3 : Jeu tic-tac-toe graphique sur ordinateur.

-a- (2,5 points) La première modalité considérée est la reconnaissance de la parole. Le choix du rond ou de la croix est fait automatiquement et l'utilisateur énonce des commandes vocales pour sélectionner une case : « Sélectionne [numéro de case] », les cases étant numérotées de 1 à 9. L'activation de la reconnaissance de parole est faite soit par l'appui sur un bouton sur l'écran ("push-to-talk"), soit par un geste dans l'air ("raise-to-talk"). Pour le geste, l'utilisateur lève le téléphone à sa bouche pour lancer la commande, et abaisse le téléphone pour terminer la commande et commencer la reconnaissance, comme illustré à la Figure 2.

- Décrivez les trois modalités mises en jeu sous la forme <dispositif, langage>

- Caractérissez les trois modalités mises en jeu (et justifiez vos réponses)

- Quelles sont les relations temporelles existantes entre les modalités (Anachronisme, Séquence, Concomitance etc..) ? Justifiez votre réponse.

- Faites le diagramme CARE complet de l'interaction basée sur la parole que ce soit couplée avec un bouton appuyé sur l'écran ou un geste dans l'air (Figure 2). La tâche de ce diagramme CARE est <sélectionner(numéro de case)>.

-b- (3 points) Proposez une interaction multimodale de type complémentarité mettant donc en jeu au moins deux modalités pour sélectionner un rond ou une croix ainsi qu'une case : caractérisez les deux modalités et faites le diagramme CARE pour la tâche <poser (rond/croix) (numéro de case)>.

Annoncez les avantages de votre solution conçue avec les caractéristiques des modalités mises en jeu.

Remarque : Les possibilités sont multiples comme l'interaction tangible en posant des objets physiques de forme circulaire ou en croix directement sur l'écran tactile, l'interaction au pied pour sélectionner une case etc.

-c- (0,5 point) L'interaction multimodale de type complémentarité que vous avez conçue pour la question (b) est-elle de type alterné ou synergique ? Justifiez votre réponse et expliquez la différence entre alterné et synergique.

QUESTION 2 JEU TIC-TAC-TOE SUR ORDINATEUR 6,5 points

Nous considérons l'application tic-tac-toe graphique de la Figure 3. L'application comporte deux zones : une palette et un plateau de jeu. La palette contient deux boutons « rond » et « croix ». Pour jouer, l'utilisateur clique sur l'un des deux boutons graphiques, et tout en laissant le bouton de la souris appuyé déplace la souris. La forme sélectionnée (rond ou croix) suit le curseur. Enfin l'utilisateur relâche le bouton de la souris sur une case. L'élément est alors posé. En cliquant à nouveau sur l'un des deux boutons graphiques, l'utilisateur peut jouer à nouveau.

-a- (1,5 point) Modélisez l'action de jouer avec la Théorie de l'Action. Le but de l'utilisateur est « jouer un tour ». Définissez les distances articulatoires et sémantiques de la Théorie de l'Action (remarque : 4 distances à définir) en les illustrant avec le jeu Tic-Tac-Toe.

-b- (1,5 point) Décrivez l'action de jouer à l'aide d'un tableau UAN.

-c- (1,5 point) Codez avec KEYSTROKE la méthode ci-dessus pour jouer. Nous considérons que l'utilisateur a les mains sur le clavier. A la fin d'un tour de jeu l'utilisateur ramène ses mains sur le clavier. Justifiez votre réponse.

-d- (0,5 point) Calculez le temps de réalisation pour jouer, en considérant les valeurs moyennes suivantes :

- Déplacer la souris sur une cible ou un point : 1100 ms
- Appuyer ou relâcher le bouton de la souris : 100 ms
- Bouger les mains entre la souris et le clavier : 400 ms
- Acte de penser à l'opération à effectuer : 1200 ms

-e- (1,5 point) Nous considérons qu'au lieu d'avoir une palette, l'application ne comporte que le plateau de jeu. L'utilisateur clique sur une case. Tout en laissant le bouton de la souris appuyé l'utilisateur énonce la commande vocale "rond" ou "croix" puis relâche le bouton de la souris. Codez avec KEYSTROKE cette nouvelle façon de jouer et calculez le temps de réalisation pour jouer un tour en considérant que la valeur moyenne pour énoncer la commande vocale est 260 ms. Concluez sur la façon la plus rapide de jouer.

QUESTION 3 LOI DE FITTS 2 points

-a- (0,5 point) Expliquez la loi de Fitts et son intérêt pour la conception d'applications interactives. (0,5 point)

-b- (1,5 point) Donnez un exemple de conception d'IHM et argumentez en quoi elle est meilleure qu'une autre par rapport à la loi de Fitts. Plus précisément, vous devez décrire une bonne conception ainsi qu'une moins bonne conception et argumenter en utilisant cette loi.

QUESTION 4 TIC-TAC-TOE DEVELOPPEMENT 5,5 points

L'entreprise Nyantac souhaite créer un jeu tic-tac-toe multi-joueur. Le principe est simple : on marche dans la rue, quand un adversaire est proche de nous, notre téléphone vibre pour nous signaler un potentiel adversaire. En regardant son téléphone, un bouton « Start game » est affiché. En cliquant dessus la partie commence. Chaque joueur pourra jouer un rond ou une croix à l'aide de la reconnaissance vocale.

Un développeur de Nyantac commence le projet en créant une nouvelle « empty activity ». Il veut déjà essayer de faire vibrer le téléphone. Il modifie uniquement le fichier MainActivity.java comme suit :

```
public class MainActivity extends AppCompatActivity {
    Button button;
    Vibrator vibrator;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = findViewById(R.id.button);
        vibrator = (Vibrator) getSystemService(VIBRATOR_SERVICE);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (Build.VERSION.SDK_INT >= 26) {
                    vibrator.vibrate(VibrationEffect.createOneShot(200,
                        VibrationEffect.DEFAULT_AMPLITUDE));
                } else {
                    vibrator.vibrate(200);
                }
            }
        });
    }
}
```

-1- Lorsque qu'il lance l'application, elle crashe immédiatement. Android Studio nous indique :
"Call requires permission which may be rejected by user: code should explicitly handle a potential SecurityException..."

-a- (0.5 point) Une ligne de code doit être ajoutée dans un autre fichier pour demander cette permission, lequel ?

-b- (1 point) En 3 lignes, expliquez le principe des permissions utilisateur : quel est le principe général, quel est l'intérêt pour l'utilisateur final ?

-2- (1 point) Le développeur veut maintenant implémenter la reconnaissance vocale. Les commandes vocales sont de la forme : « placer rond 3 2 » « placer croix 1 2 ». Écrivez la fonction qui à partir d'une chaîne de caractères place un rond ou une croix sur la grille. Vous disposez des deux fonctions suivantes (pas besoin de les implémenter) :

- DrawCircle(int x, int y)

- DrawCross(int x, int y)

Ces fonctions dessinent un rond (DrawCircle) ou une croix (DrawCross) dans la case de la x ligne et y colonne. Rappel JAVA : fonction de la classe String : `split (" ")` renvoie un tableau de chaînes de caractères des mots séparés par un espace. Par exemple : `String[] sousChaines = maChaine.split(" ");`

-3- (3 points) Avant de passer en mode multi-joueur, on se propose d'implémenter une version avec un seul joueur. Quand l'utilisateur se trouve proche d'un point d'intérêt, son téléphone vibre. En regardant son téléphone, un bouton lui permet de jouer contre une IA en utilisant la reconnaissance vocale. La reconnaissance vocale est activée à l'aide d'un bouton. Une amie vous a implémenté la fonction suivante :

```
public boolean isCloseEnoughToPlay(double latitude, double longitude);
```

Elle retourne `True` si le joueur est proche d'un point d'intérêt et `False` sinon.

Sans écrire exactement le code final, expliquez en détails tout ce que vous devez faire pour implémenter cette version avec un seul joueur, en précisant quel(s) fichier(s) vous modifiez.