Projet : Morpion (Tic-Tac-Toe)

Fall 25/26

ECE: Bachelor 1

20 Novembre 2025

ECE
ÉCOLE D'INGÉNIEURS
ENGINEERING SCHOOL

Objectifs du Projet

Ce projet consiste à développer un jeu complet du **Morpion (Tic-Tac-Toe)** en langage C.
À l'issue du projet, l'étudiant sera capable de :

- manipuler et exploiter efficacement des tableaux 2D, et pointeurs
- structurer un projet en plusieurs fonctions et procedures,
- implémenter une alternance de joueurs et une logique de tour,
- gérer les interactions clavier de manière sécurisée,
- détecter toutes les situations logiques : victoire, match nul, erreurs de saisie,
- produire un projet en clode : :Blocks propre, lisible et testé.

Présentation Générale de la Version Console

Vous devez réaliser un Morpion entièrement fonctionnel dans un terminal, sans aucune interface graphique.
L'affichage doit être clair et mis à jour à chaque tour.

Les joueurs saisissent leurs coups via le clavier :
numéro de ligne (0, 1 ou 2) et *numéro de colonne* (0, 1 ou 2).

Fonctionnalités Obligatoires et Attendus Finaux(1/2)

Votre programme devra obligatoirement :

1. Afficher la grille sous forme ASCII, mise à jour après chaque coup.
2. Demander au joueur courant (X ou O) où jouer.
3. Vérifier la validité de la saisie :
   - indices de ligne/colonne entre 0 et 2,
   - case vide et jouable.
4. Empêcher tout coup invalide, sans planter.
5. Alterner automatiquement entre les joueurs X et O.
6. Chronométrer le temps utilisé par chaque joueur
   - un chronomètre indépendant pour X et pour O,
   - affichage à la fin de la partie :

```
Temps total joueur X = ... secondes
Temps total joueur O = ... secondes
```

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Fonctionnalités Obligatoires et Attendus Finaux (2/2)

Votre programme devra obligatoirement :

7. Enregistrer l'intégralité de la partie dans un fichier texte :
   - chaque coup doit être sauvegardé : numéro de tour, joueur, coordonnées,
   - exemple : `Tour 3 : Joueur X joue (1,2)`
   - le fichier doit être affiché à la fin de la partie.
8. Détecter toutes les conditions de victoire :
   - 3 symboles identiques alignés horizontalement,
   - verticalement,
   - ou diagonalement.
9. Détecter un match nul lorsque la grille est remplie.
10. Afficher un message final clair et l'historique du jeu.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Contraintes Techniques STRICTES(1/2)

Votre code doit obligatoirement contenir les fonctions suivantes :

Fonctions imposées

- `void initBoard(char board[3][3]);`
- `void printBoard(char board[3][3]);`
- `int checkWin(char board[3][3]);`
- `int isFull(char board[3][3]);`
- `void logMove(FILE *f, int turn, char player, int row, int col);`

Fonctions recommandées

- `void playTurn(char board[3][3], char player);`
- `double measureTime(char player);` (chronométrage par joueur)

Contraintes Techniques STRICTES(2/2)

CONTRAINTES IMPORTANTES

- **Aucune variable globale n'est autorisée.**
- Le tableau `board` doit être passé entre fonctions.
- Le chrono doit être basé sur les fonctions `clock()` ou `time()`.
- Le fichier d'historique doit être nommé : `historique_morpion_datexxx.txt`
- Le fichier doit être rouvert en fin de partie et affiché dans le terminal.
- Le code projet doit être entièrement commenté et lisible.

Cas Particuliers à Gérer (Obligatoires)

Votre projet doit également gérer correctement :

- Saisie hors limites (ex : ligne 5, colonne -1),
- Saisie non numérique (ex : “a”, “ !” ou chaîne vide),
- Tentative de jouer sur une case déjà occupée,
- Enchaînement de plusieurs erreurs sans planter le programme,
- Réaffichage propre de la grille après chaque tentative valide,
- Fichier d'historique introuvable ou non accessible,
- Chronométrage cohérent même en cas d'erreur de saisie.

Tout comportement imprévu ou plantage entraînera une réduction importante de la note.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Barème de notation (20 points)

Évaluation détaillée

- **Affichage et mise à jour de la grille : 5 pts**
  - grille correcte, claire et stable : 2 pts
  - rafraîchissement cohérent : 1 pt
  - bonne lisibilité : 2 pts
- **Validation des coups + gestion des erreurs : 5 pts**
  - contrôle des bornes : 2 pts
  - gestion case déjà occupée : 1 pt
  - programme robuste face aux erreurs : 2 pts
- **Détection victoire / égalité : 5 pts**
  - lignes / colonnes : 2 pts
  - diagonales : 1 pt
  - match nul : 2 pts

Barème de notation (20 points)

Évaluation détaillée

- **Structure, organisation et qualité du code : 5 pts**
  - respect des fonctions imposées : 2 pts
  - absence de variables globales : 1 pt
  - commentaires / indentation : 1 pt
  - propreté générale : 1 pt

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Livrables

Votre rendu doit contenir :

- le code source complet : `.c` ;
- un fichier texte `README.txt` contenant :
  - votre stratégie d'implémentation,
  - les fonctions principales,
  - les tests réalisés,
  - les difficultés rencontrées ;
- une video de 3 minutes montrant une partie complète.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Notation de la Présentation Orale (Vidéo 7 à 10 minutes)(1/2)

Chaque étudiant doit enregistrer une **présentation orale** d'une durée comprise entre **7 et 10 minutes**, en expliquant son travail et sa démarche de réalisation.

Éléments évalués (5 points)

- **Clarté de la présentation : 1 pt** Explication fluide, vocabulaire correct, structure logique.
- **Description du fonctionnement du programme : 1.5 pts** Présentation du plateau, gestion des tours, détection victoire/égalité.
- **Explication technique : 1.5 pts** Organisation du code, fonctions développées, gestion des erreurs, chronométrage, fichier historique.
- **Démonstration vidéo : 1 pt** Partie jouée en direct + affichage temps/joueurs + fichier historique.

Présentation Orale (Vidéo 7 à 10 minutes)(2/2)

Contraintes

- Durée obligatoire : entre **7 min et 10 min**. (-1 pt si < 7 min ou > 10 min)
- Le visage de l'étudiant n'est pas obligatoire, mais la voix doit être claire.
- L'écran ou le terminal doit être visible en permanence.
- La vidéo doit être envoyée au format : **.mp4**, **.mkv**, **.mov**.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Formation des Équipes et Enregistrement

Le projet doit être réalisé en **équipes de 3 à 4 étudiants**. Chaque équipe travaille sur un seul rendu commun (code + vidéo + livrables).

Consignes de formation des équipes

- Chaque équipe doit être composée de **3 ou 4 membres** uniquement.
- Les groupes doivent être équilibrés en termes de compétences.
- Un étudiant ne peut appartenir qu'à une seule équipe.
- Le projet entier (conception, codage, tests, vidéo) est réalisé collectivement.

Formation des Équipes et Enregistrement

Enregistrement obligatoire des équipes

- Les équipes doivent être déclarées **dans les 24 heures** suivant le **kick-off** du projet.
- La liste des membres doit est visible dans le fichier partagé avec la promo B1 :
  - contenant : Numero team, noms, prénoms, emails, numéro d'étudiant.
- Toute modification de groupe après ce délai nécessite une **validation de l'enseignant**.

Aucune équipe non enregistrée dans les délais ne sera acceptée.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Timeline du Projet (Durée : 20 jours)(1/3)

Le projet s'étale sur **20 jours calendaires**. Chaque phase correspond à une étape clé à valider avant de passer à la suivante.

Jour 1 — Kick-off du Projet

- Présentation du sujet, attentes et contraintes techniques.
- Formation des équipes (3 à 4 étudiants).
- **Enregistrement des équipes avant J+1**.

Jours 2 à 6 — Conception et Mise en place du squelette

- Définition des structures de données.
- Rédaction des prototypes de fonctions.
- Construction du squelette : fichiers, organisation, commentaires.
- Implémentation de `initBoard()` et `printBoard()`.

Timeline du Projet (Durée : 20 jours)(2/3)

Jours 7 à 12 — Développement du cœur du jeu

- Implémentation de :
  - `checkWin()`,
  - `isFull()`,
  - gestion des erreurs de saisie.
- Développement de `playTurn()`.
- Tests partiels : cas normaux et cas limites.

Timeline du Projet (Durée : 20 jours)(3/3)

Jours 13 à 16 — Fonctionnalités avancées

- Chronométrage du temps total du joueur X et du joueur O.
- Implémentation et écriture de l'historique dans le fichier : `historique_morpion.txt`
- Affichage du fichier d'historique à la fin de la partie.
- Tests complets de robustesse.

Jours 17 à 20 — Finalisation et Rendu

- Finalisation du code, commentaires, indentation, nettoyage.
- Rédaction du fichier `README.txt`.
- Enregistrement de la vidéo de présentation (7–10 minutes).
- Vérification finale + dépôt du projet complet.

Le respect strict du planning est fortement recommandé.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Rapport Écrit — Structure Générale

Chaque équipe doit remettre un **rapport écrit de 6 pages maximum** résumant l'ensemble du travail réalisé.

Structure obligatoire

- Page 1 : Introduction et objectifs du projet
- Pages 2–3 : Conception, architecture et organisation du code
- Page 4 : Chronométrage (X et O) et historique du fichier
- Page 5 : Tests et validation fonctionnelle
- Page 6 : Conclusion, répartition du travail, perspectives

Le rapport doit refléter fidèlement le travail de l'équipe.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL

Rapport Écrit — Contenu Détaillé

Le rapport doit couvrir précisément les points suivants :

Pages 1 à 3 : Aspects techniques

- Explication du choix des structures et variables.
- Description détaillée des fonctions :
  - `initBoard()`, `printBoard()`,
  - `checkWin()`, `isFull()`,
  - `playTurn()`, chronomètre, log fichier.
- Diagramme simple de l'architecture du programme.

Pages 4 à 6 : Analyse et validation

- Méthode utilisée pour chronométrer X et O.
- Extrait du fichier `historique_morpion.txt`.
- Campagne de tests (victoire, égalité, erreurs).
- Perspectives et extensions).

Rapport Écrit — Contraintes de Rédaction

Le rapport doit respecter strictement plusieurs règles de présentation.

Contraintes de mise en forme

- Format **PDF uniquement**.
- Maximum **6 pages** (hors annexes optionnelles).
- Police recommandée : **Arial** ou **Times New Roman**, taille 11.
- Interligne simple, marges classiques.
- Illustrations et extraits de code autorisés, mais concis.

Exigences supplémentaires

- Toute copie ou similitude excessive entre équipes = 0/20.
- Le rapport doit refléter un travail collectif (3–4 étudiants).
- Le fichier PDF doit être nommé :
  `rapport_equipe_XX.pdf`

Deadline de Soumission

Deadline officielle de dépôt

14 Décembre — 23h55
Soumission obligatoire sur la plateforme Boostcamp

Éléments à déposer

- Code source complet du projet (`.c`)
- Rapport écrit (**6 pages**)
- Vidéo de présentation orale (durée : 7–10 minutes)
- Fichier `README.txt`
- Fichier d'historique : `historique_morpion_g_xx.txt`

Règles et Mise en Garde

Mise en garde importante

- Aucune tolérance pour les dépôts en retard.
- La plateforme Boostcamp fait foi pour l'heure officielle de dépôt.
- Tout code non fonctionnel ou non compilable = **pénalisation sévère**.
- Le projet doit fonctionner sur un compilateur standard (GCC) de code : :blocks.

ECE
ÉCOLE D'INGÉNIEURS
ENGINEERING SCHOOL

Interdiction d'utilisation d'IA générative

Interdiction formelle

- **Interdiction totale d'utiliser toute IA générative** (ChatGPT, Copilot, Gemini, Mistral, Claude, etc.) pour :
  - rédiger tout ou partie du rapport,
  - générer ou modifier le code C,
  - générer les commentaires du code,
  - rédiger le README ou autres livrables.
- Le rapport doit être rédigé **manuellement** par les membres de l'équipe.
- Des entretiens individuels de vérification peuvent être effectués.
- Toute violation = **zéro pour l'ensemble du projet**.

Aucune exception ne sera accordée.

ECE ÉCOLE D'INGÉNIEURS ENGINEERING SCHOOL