

Projet Blackjack

Table des matières

1. Analyse	2
2. Conception	3
3. Réalisation	4
4. Tests	5
5. Mise en service	5
6. Maintenance	5



1. Analyse

Objectif principal :

L'objectif de ce projet est de développer un jeu de Blackjack multijoueur local en Python, avec une interface graphique interactive réalisée à l'aide de Tkinter. Le jeu permet à deux joueurs humains d'affronter un croupier contrôlé par l'ordinateur, selon les règles classiques du Blackjack.

Le projet met l'accent sur la gestion automatique des scores, des mises et du solde des joueurs, ainsi que sur une expérience utilisateur claire et intuitive. Les cartes sont affichées à l'aide d'images afin d'améliorer l'aspect visuel du jeu.

Exigences et contraintes :

- Interface graphique intuitive (boutons Tirer, Rester, Rejouer, Quitter).
- Calcul automatique des scores avec gestion correcte de l'As (1 ou 11).
- Logique automatique du croupier.
- Mode local uniquement (aucune connexion réseau).
- Gestion des mises et du solde des joueurs.
- Utilisation de Python et Tkinter, sans base de données externe.

Livrables de cette phase :

- Cahier des charges décrivant les fonctionnalités attendues.
- Définition de la structure du projet et de l'interface graphique.

2. Conception

Interface graphique :

- Zone d'affichage des cartes des joueurs et du croupier.
- Zone de saisie et d'affichage des mises.
- Boutons d'action : Tirer, Rester, Rejouer, Quitter.
- Zone d'affichage des scores, du solde et des messages de jeu.

Diagramme de flux :

[Cliquer pour voir.](#)

Organisation du programme :

Le projet est conçu de manière modulaire afin de séparer la logique du jeu, la gestion des joueurs et l'interface graphique. Cette organisation facilite la maintenance et l'évolution du programme.

3. Réalisation

Structure du projet :

- main.py : lancement du jeu.
- gui.py : gestion de l'interface graphique Tkinter.
- blackjack.py : logique principale du jeu.
- player.py : gestion des joueurs (nom, solde, cartes).
- dealer.py : gestion du croupier.
- score_manager.py : sauvegarde locale des résultats.

Technologies utilisées :

- Python
- Tkinter
- Modules standards (random, json, time)

Développement prévu en sprints :

Sprint	Durée	Objectif principal
Sprint 1	1 semaine	Interface Tkinter de base + affichage des cartes
Sprint 2	1 semaine	Logique du jeu (distribution, score, As, croupier, enregistrement des scores)
Sprint 3	1 semaine	Gestion des mises et solde + affichage des résultats
Sprint 4	1 semaine	Améliorations visuelles et tests utilisateurs

4. Tests

Types de tests :

- Tests unitaires : vérification de la distribution des cartes, du calcul des scores et de la gestion de l'As.
- Tests fonctionnels : déroulement complet d'une partie jusqu'à la détermination du gagnant.
- Tests d'interface : fonctionnement des boutons, affichage des cartes et possibilité de rejouer.

5. Mise en service

- Le jeu est fourni sous forme de fichiers Python prêts à être exécutés localement. Une présentation du projet est prévue en classe, accompagnée d'une explication du fonctionnement du jeu et de l'interface.
- Un court manuel utilisateur permet de comprendre rapidement les règles et les actions disponibles.

6. Maintenance

- Correction des bugs mineurs.
- Amélioration possible : ajout d'un mode en ligne, animation des cartes, historique détaillé.