

Skyjo action – Groupe 1

BOUDET Duncan 1A-A1 – BOUDET Romane 1A-A1

Éléments d'analyse

Skyjo action est un jeu pour 2 à 8 joueurs, se jouant sur plusieurs manches. Chaque joueur reçoit 12 cartes, qu'il dispose face cachée devant lui. Le but du jeu est de marquer le moins de points possible en remplaçant ses cartes par celles de la pioche ou de la défausse. Les valeurs des cartes vont de -2 à 12 points. Pour pimenter le jeu, les joueurs peuvent jouer des cartes actions pour faciliter leur perte de point. La manche se termine quand un joueur a relevé toutes ces cartes. La partie se termine quand un joueur dépasse 100 points. Le joueur a besoin d'observer les jeux des autres pour pouvoir jouer judicieusement.

Schéma des Structures de données utilisées

```
struct etoile
{
    int ligne;
    int colonne;
};

struct classemt
{
    int classement;
    int score;
};

struct pioche
{
    int nombre[CARTE_PIOCHE_NOMBRE_NB];
    int nombre_nb;
    int nombre_defausse[CARTE_PIOCHE_NOMBRE_NB];
    int nombre_defausse_nb;
    int nombre_defausse_dessous[9];
    int nombre_defausse_dessous_nb;
    int action[CARTE_PIOCHE_ACTION_NB];
    int action_nb;
    int action_visible[CARTE_PIOCHE_ACTION_FACE_VISIBLE];
    int action_defausse[CARTE_PIOCHE_ACTION_NB];
    int action_defausse_nb;
    int premier_fin_manche;
    int tour_restant;
};

struct joueur
{
    int joueur_no;
    int deck_nombre
    [CARTE_JOUEUR_NOMBRE_L][CARTE_JOUEUR_NOMBRE_C];
    int deck_nombre_cache
    [CARTE_JOUEUR_NOMBRE_L][CARTE_JOUEUR_NOMBRE_C];
    int nb_action; //nombre de carte action dans le jeu du joueur.
    char prenom [DIM_STR];
    int score;
    S_etoile pos_etoile[12];
    int nb_etoile;
    int nb_skyjo_ligne;
    int nb_skyjo_colonne;
    int nb_coups;
    int score_manche;
    int classement;
};
```

Éléments de conception

Fonctionnement du programme :

[Phase d'initialisation]

Initialisation de toutes les variables de la partie

Génération du plateau (sous programme)

Affichage du plateau

[Tour d'un joueur]

Lui présenter toutes les données qu'il a besoin pour prendre la bonne décision

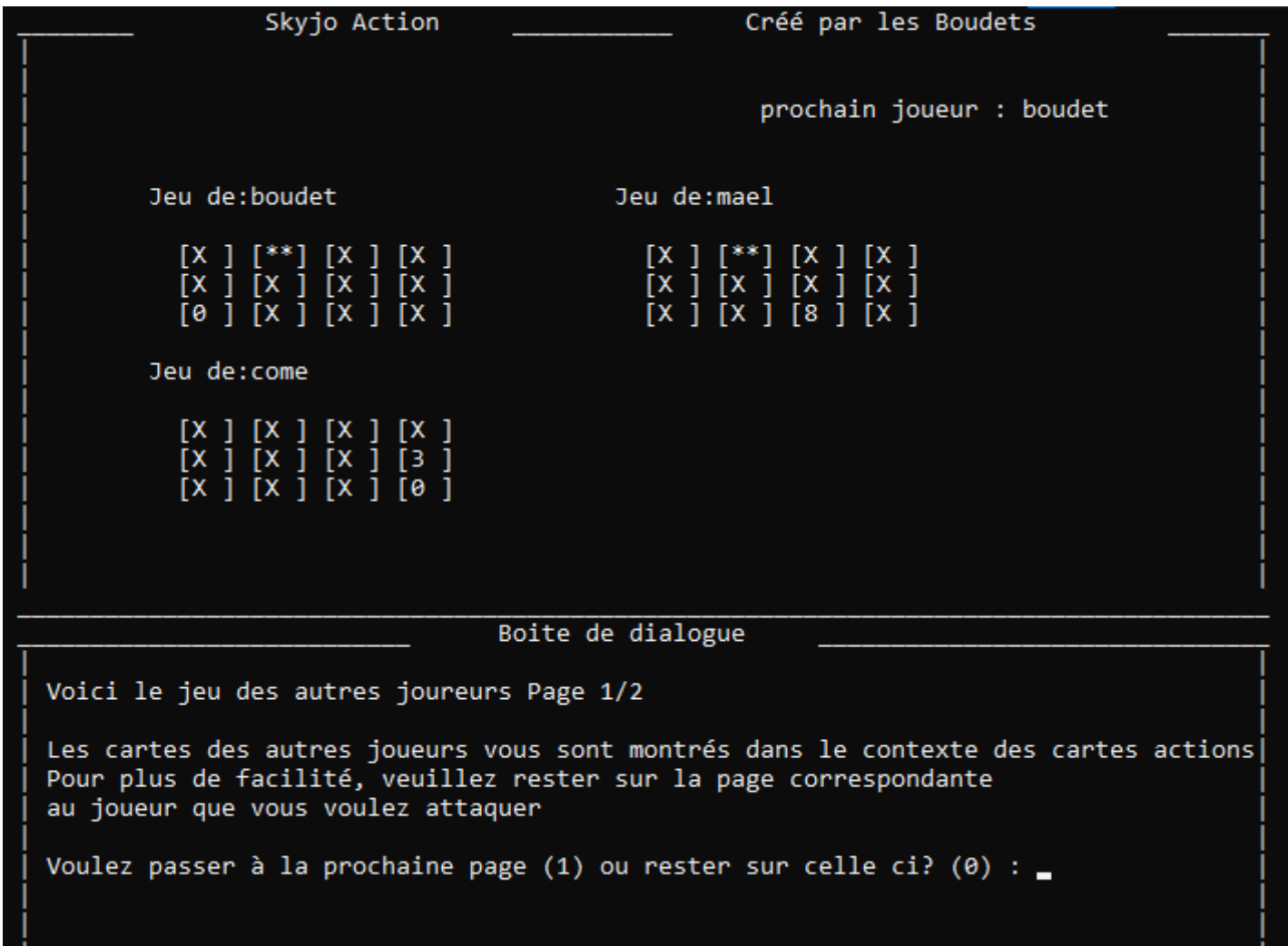
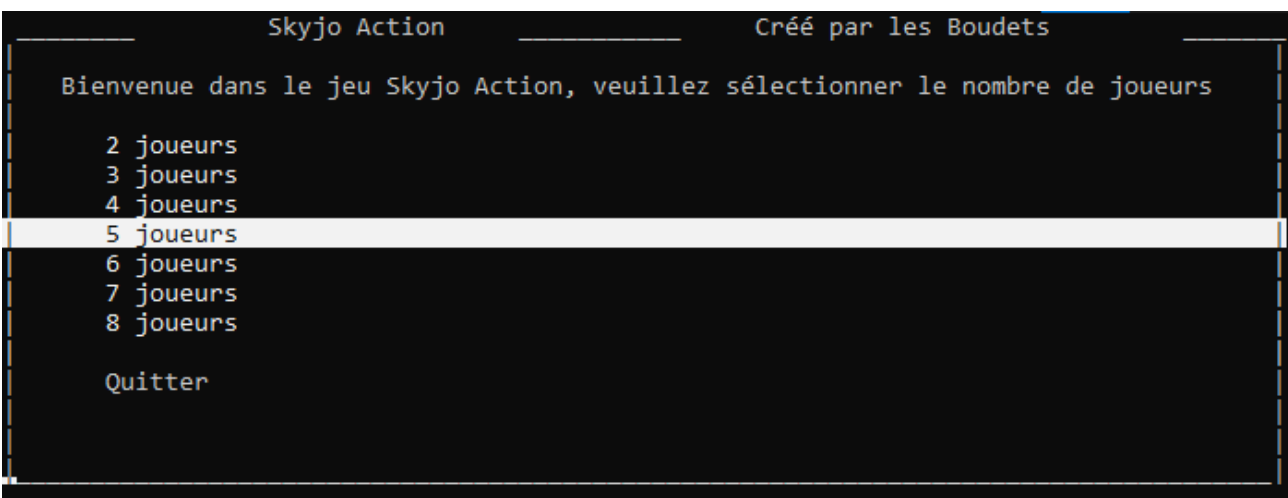
Lui proposer toutes les actions qu'il peut faire

[Fin de manche/tour]

Tester les conditions d'arrêts après que le tour de table soit terminé

Actualiser les scores en fonction des règles de jeu

Capture d'écran



Lots prévus

Semaine 1-2	Semaine 2-3	Semaine 4-6	Semaine 6-8
Analyse	Conception	Ecriture/Debug	Ecriture/Debug

Conclusion

(que changeriez-vous dans votre démarche projet pour être plus efficient une prochaine fois ?)

Utilisation de tableaux dynamiques pour les prénoms des joueurs

Utiliser le maximum de sous-programmes dès le début de l'écriture

Créer un affichage plus souple permettant d'afficher tous les joueurs à la fois

Passer plus de temps sur l'analyse pour avoir un programme plus cohérent (ex: les variables tour_restant et fin_premier_manche)