

ChessRPG

Pierre-Louis Gondras
Robin Greyl
Tiffanie Schreyeck

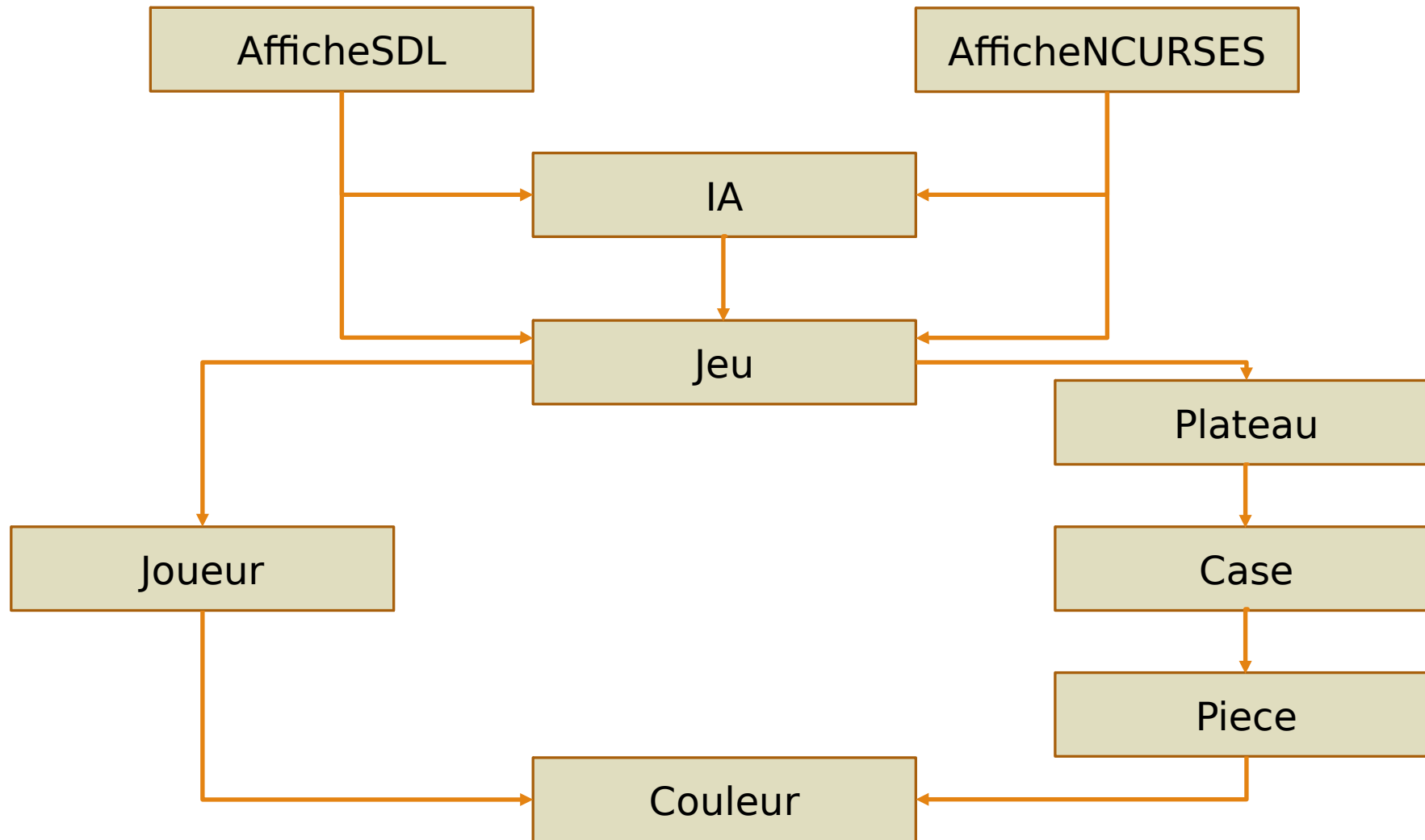
Principe du jeu

Echecs	RPG
<ul style="list-style-type: none">• Plateau• Pièces• Déplacement des pièces	<ul style="list-style-type: none">• Combat entre les pièces• Points de vie et d'attaque pour chaque pièce• Tuer le roi pour gagner la partie

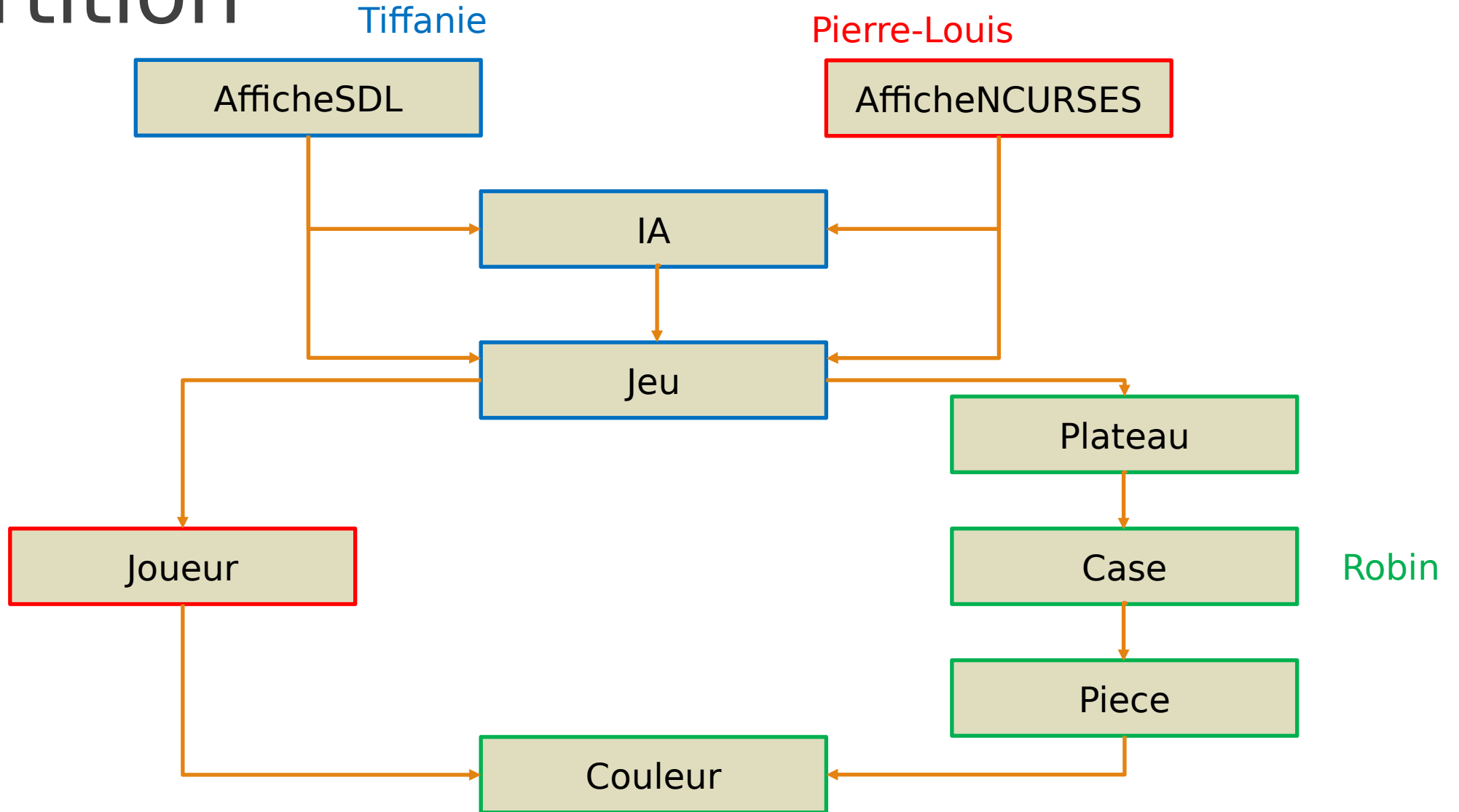
Fonctionnalités

- Choix du nom des joueurs, des différents types de pièces
- Jeu fonctionnel avec une interface graphique et console
- Affichage des PV & PA au survol
- Affichage des logs
- possibilité de jouer contre l'IA

Diagramme des modules



Répartition




Module : Couleur

```
typedef enum {BLANC, NOIR, BLEU, JAUNE, NUM_COULEUR} Couleur ;
```



Couleurs disponibles pour les pièces



Dernier élément : nombre d'éléments

Module : Joueur

typedef struct

{

char nomJoueur[13] ;

Couleur couleur ;

int nbPieces;

Piece * ensPieces[16];

} Joueur;

void initJoueur(**Joueur** * joueur)

void setPieceJoueur(**Joueur** * joueur, **Piece** * piece)

char * getNomJoueur(**Joueur** * joueur)

Couleur getCouleurJoueur(**Joueur** * joueur)

Module : Piece

```
typedef enum {PION, TOUR, CAVALIER, FOU, DAME, ROI} Type ;
```

```
typedef struct
```

```
{
```

```
    Type type ;
```

```
    Couleur couleur ;
```

```
    int pointsVie ;
```

```
    int pointsAttaque ;
```

```
    Int posX ;
```

```
    Int posY ;
```

```
} Piece ;
```

```
void initPiece(Piece * piece, Type type, Couleur couleur)
```


Module : Case

```
typedef enum {CBLANC, CNOIR, CBLEU, CROUGE} CouleurCase ;
```

Survol

Atteignable

```
typedef struct
```

```
{
```

```
    CouleurCase couleurCase ;
```

```
    Piece * piece ;
```

```
} Case ;
```

```
bool caseValide(int x, int y)
```

```
void initCase(Case * cell)
```

```
void detruireCase(Case * cell)
```

Module : Plateau

typedef struct

```
{  
    Case ** echiquier ;  
    int nbTours ;  
} Plateau ;
```

```
void reinitCouleursEchiquier(Plateau * plateau)  
void initPlateau(Plateau * plateau, Couleur C1, Couleur C2)
```

Module : Jeu

typedef struct

```
{  
    Plateau plateau ;  
    Joueur J1 ;  
    Joueur J2 ;  
    Joueur * joueurActif ;  
    char log[1000] ;  
    TypeJeu typeJeu;  
    Couleur couleurGagnant;  
} Jeu ;  
  
void selectPiece(Jeu * jeu, int posX, int posY)  
void deplacerPiece(Jeu * jeu, Piece * piece, int posX, int posY)  
Piece * combatPieces(Jeu * jeu, Piece * pieceAtt, Piece * pieceDef)
```

Module : IA

typedef struct

```
{  
    Piece * piece;  
  
    int nbCasesControlees;  
    Case * casesControlees[27];  
  
    int nbPiecesControlees;  
    Piece * piecesControlees[8];  
  
    int nbDeplacementsPossibles;  
    Case * deplacementsPossibles[27];  
} IApiece ;
```

typedef struct

```
{  
    IApiece iaPiecesJ1[16];  
    IApiece iaPiecesJ2[16];  
  
} IAjeu ;
```

void ia (**Jeu** * jeu)

void casesPiecesControleesIApiece (**IApiece** * iaPiece, **Jeu** * jeu, **int** x, **int** y)

bool pieceVulnerableIAjeu (**IAjeu** * iaJeu, **Jeu** * jeu, **Piece** * piece, **int** x, **int** y)

Piece * joueurVulnerableIAjeu (**IAjeu** * iaJeu, **Jeu** * jeu, **Joueur** * joueur)

Module : AfficheNCURSES

typedef struct

{

Jeu jeu;

WINDOW * echiquier ;

WINDOW * fond;

WINDOW * jActif;

WINDOW * stats ;

WINDOW * logs;

WINDOW * victoire;

} **JeuNCURSES**;

void NcursesInit(**JeuNCURSES** * jeuNcurses)

void affichagePlateau(**JeuNCURSES** * jeuNcurses)

void boucleEvent(**Jeu** * jeu)

void NcursesLibere(**JeuNCURSES** * jeuNcurses)

Module : AfficheSDL

typedef struct

{

Jeu jeu ;

SDL_Surface* surface_ecran ;

SDL_Surface* surfaces_pieces1[6] ;

SDL_Surface* surfaces_pieces2[6] ;

SDL_Surface* surfaces_couleurs_cases[4] ;

SDL_Surface* surfaces_texte[4] ;

SDL_Surface* lettres[16] ;

TTF_Font* polices[3] ;

SDL_Surface* logs[10] ;

} JeuSDL ;

void SdlInit(**JeuSDL *** jeuSDL) ;

void SdlBoucle(**JeuSDL *** jeuSDL) ;

Conclusion

- Respect du cahier des charges initial
- Respect des contraintes
- Programme fonctionnel (modules, ncurses, SDL)
- Fonctionnalité supplémentaire : IA