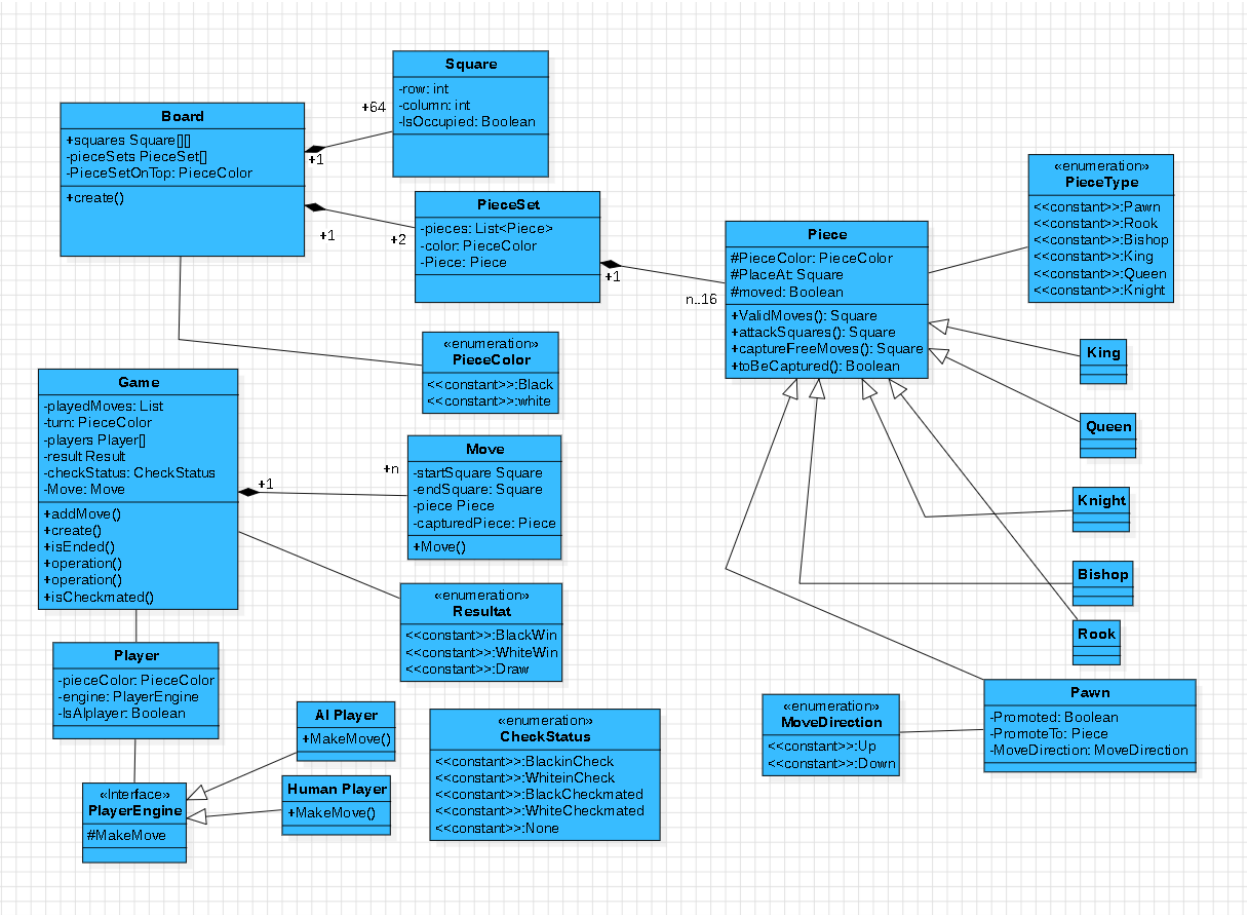
– Rendu 4 – Conception détaillée et

Préparation Tests unitaires

Etape 1: 2è raffinement avec la préparation de la programmation en parcourant les éléments de modèles à traduire et à programmer (attributs et opérations)



Etape 2: Invariants

A - Invariants

Classe Square

Invariant : La formule en logique propositionnelle qui représente l'invariant de la classe Square pourrait être exprimée comme suit :

$$(piece=\text{null}) \vee ((piece.type \in \{'King','Queen','Rook','Bishop','Knight','Pawn'\}) \wedge (occupant \in \{'Player1','Player2'\}))$$

- Cela signifie qu'une case (Square) doit soit être vide ($piece$=null*piece*=null), soit contenir une pièce de type spécifié (Roi, Reine, Tour, Fou, Cavalier, Pion) avec un occupant précisément défini (occupant étant Player1 ou Player2).

B - Tables de décisions des tests unitaires

❖ Opération movePiece() de la classe Piece

- Conditions :
  - C1 : La pièce est-elle sur l'échiquier ? (True/False)
  - C2 : Le mouvement est-il légal pour cette pièce ? (True/False)
  - C3 : La destination est-elle occupée par une pièce alliée ? (True/False)
- Actions :
  - A1 : Déplacer la pièce
  - A2 : Ne pas déplacer la pièce
- Table de décision :

C1	C2	C3	A1	A2
True	True	False	X	
True	False	–		X
False	–	–		X
False	–	True		X

❖ Opération isCheck() de la classe King

- Conditions :
  - C1 : Le roi est-il attaqué par une ligne de vue directe (True/False)
- Actions :
  - A1 : Retourner True

- A2 : Retourner False

- Table de décision :

$C1$	$A1$	$A2$
True	X	
False		X