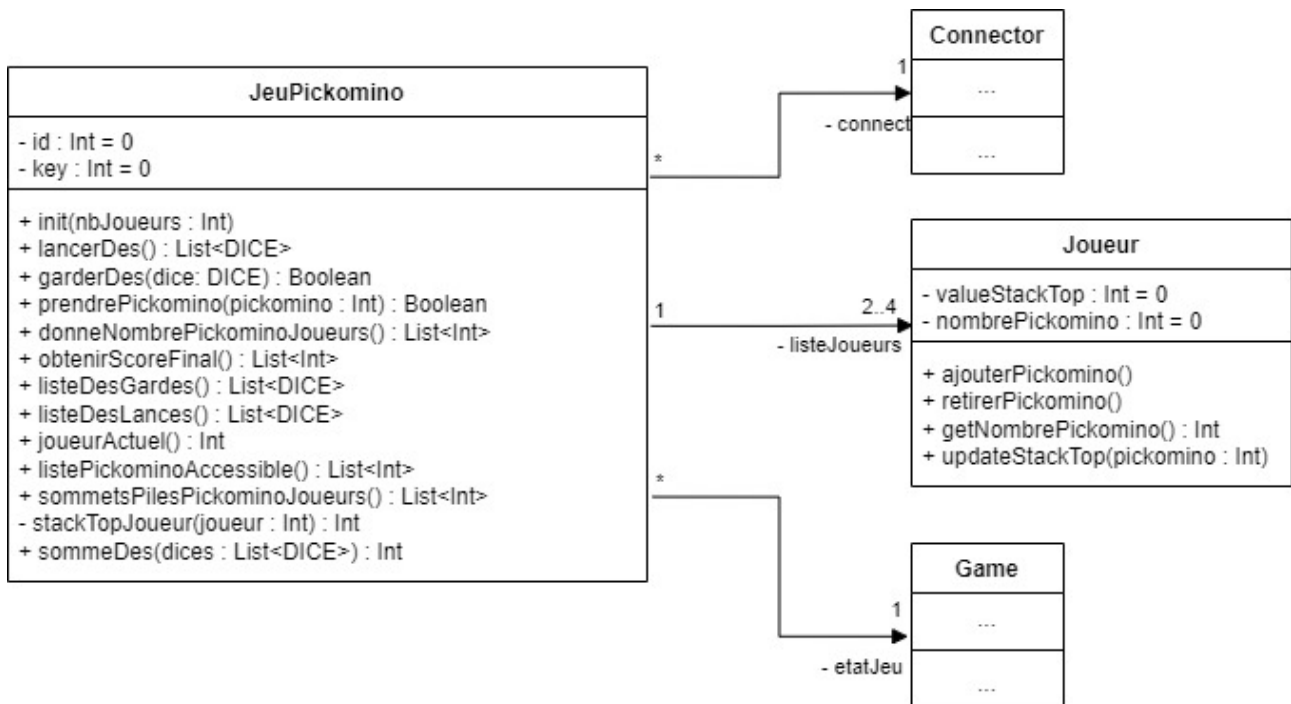


## Diagramme de classes UML pour le jeu Pickomino



Pour le jeu Pickomino, nous avons considéré 2 classes (JeuPickomino et Joueur) ou l'une utilise l'autre. Il est spécifié ici que la classe JeuPickomino utilise la classe Connector et la classe Game de la librairie PickominoLib.

La classe Joueur représente un joueur de la partie. Il possède une variable représentant la valeur du sommet de sa pile de Pickomino (0 représentant aucun) et le nombre de Pickomino qu'il possède.

La classe JeuPickomino va créer une liste de joueurs dans la fonction 'init', un objet Connector ainsi qu'un objet Game. Nous précisons que les attributs 'connect', 'etatJeu' et 'listeJoueurs' sont déclarés 'lateinit'

La fonction 'init' ne représente pas un init d'une classe, mais bien une fonction. Ce choix a été fait car notre IHM possèdera un menu, sur ce menu, il y aura un bouton "jouer" qui lancera la partie avec le nombre de joueurs choisis. Au même moment (dans le contrôleur du bouton jouer) ce fera cette fameuse fonction 'init' qui créera une nouvelle partie. L'intérêt ici est de pouvoir quand même créer un objet JeuPickomino() sans pour autant lancer une partie dès la création de cet objet. Nous pouvons voir ça comme un 'lateinit' du init de la classe.

Les autres fonctions de la classe JeuPickomino permettent d'interagir avec le serveur pour jouer.

Certaines utilisent l'id et la key pour fonctionner. Passer par cette classe permet de pouvoir exécuter ces fonctions sans renseigner en paramètre l'id et la clé (car déjà présent en attribut de la classe).

Certaines de ces classes seront utiles pour les affichages.