Monopoly EG - Modélisation UML des classes

Card User Cell Gameboard + id: int + id: int + id: int + id: int + name: string + nickname: string + type: int + cells: Cell[] + description: string + email: string + property: Property + players: Player[] + effectType: int + currPlayer: Player - password: string // hash + effectVal1: int + friends: int[] + currTimer: int + effectVal2: int + inscriptionDatetime: datetime + getName(): string + startedAt: datetime + level: int + timer: int // seconds + execute(): void # socket: Socket + chat: Chat + bank: Bank + cards: Card[] Player Chat + properties: Property[] // qui n'ont pas encore été + id: int + id: int achetées + user: User + messages: ChatMessage[] + isOnline: bool + rollDice(): [int, int] + currentCell: Cell + pawn: int Street + balance: int ChatMessage + properties: Property[] + housesNb: int + jailJokerCards: int + id: int + hotel: bool + remainingTurnsInJail: int // countdown 3 -> 0 + sender: Player + housePrice: int + content: string + hotelPrice: int + isInPrison(): bool + datetimeSent: datetime + type: int + getRealRentalPrice(): int + offer: Offer Property + id: int + name: string + description: string + price: int **TrainStation PublicCompany** + rentalPrice: int + mortgagePrice: int + isMortgaged: bool + getRealRentalPrice(): int + getRealRentalPrice(): int Bid (= enchère) PlayerBid (= enchère d'un joueur) Lobby Matchmaking + id: int + id: int + id: int + id: int + playersBids: PlayerBid[] + player: Player + nbPlayers: int // 2 -> 8 (peut être + nbPlayers: int // 2 -> 8 changé par le joueur) + datetimeBidGlobal: datetime // + amount: int + playersWaiting: Player[][] pour vérifier expiration de l'enchère + datetimeBid: datetime // pour + players: Player[] vérifier expiration de l'enchère + chat: Chat + addPlayerBid(PlayerBid): void + finalizeBid(): Player // renvoie le + addPlayer(User): void + kickPlayer(User): void vainqueur + searchGame(): void Au moment où les joueurs du lobby lancent la partie (searchGame), on Offer passe en "matchmaking" + id: int + sender: Player + receiver: Player + property: Property + amount: int

+ datetimeOffer: datetime // pour vérifier expiration de l'offre (+ il faudra fixer une durée de vie max

pour une offre)

Quest

- + id: int
- + description: string
- + reward: ?