

Turn

- diceToBePlace: Dice
- diceForSinglePlayer: Dice
- selectDiceActionCounter: int

- useToolCardActionsCounter: int
- currentToolCard: ToolCard
- actionsToPerform: int
- currentActionsStepCounter: int
- model: Model
- player: Player

- + getSelectedDiceActionCounter(): int
- + getUseToolCardActionsCounter(): int
- + setSelectedDiceActionCounter(int): void
- + setUseToolCardActionsCounter(int): void
- + getCurrentToolCard(): ToolCard
- + setCurrentToolCard(int): void
- + setActionsToPerform(int): void
- + getDiceToBePlaced(): Dice
- + setDiceToBePlaced(Dice): void
- + getDiceForSinglePlayer(): Dice
- + setDiceForSinglePlayer(Dice): void
- resetTurn(): void
- removeToken(): void

```
classDiagram
    class Turn {
        -diceToBePlace: Dice
        -diceForSinglePlayer: Dice
        -selectDiceActionCounter: int
        -useToolCardActionsCounter: int
        -currentToolCard: ToolCard
        -actionsToPerform: int
        -currentActionsStepCounter: int
        -model: Model
        -player: Player
        +getSelectedDiceActionCounter(): int
        +getUseToolCardActionsCounter(): int
        +setSelectedDiceActionCounter(int): void
        +setUseToolCardActionsCounter(int): void
        +getCurrentToolCard(): ToolCard
        +setCurrentToolCard(int): void
        +setActionsToPerform(int): void
        +getDiceToBePlaced(): Dice
        +setDiceToBePlaced(Dice): void
        +getDiceForSinglePlayer(): Dice
        +setDiceForSinglePlayer(Dice): void
        -resetTurn(): void
        -removeToken(): void
    }
    class PrivateResponse
    Turn --> PrivateResponse
```

PrivateResponse