AGPL Syntax

⟨*game*⟩ ::= ‘`Gamestate:{`’ [⟨*gamestate*⟩] ‘`}`’
‘`Player:{`’ ⟨*Dec*⟩ ‘`}`’
‘`Move:{`’ ⟨*Dec*⟩ ‘`}`’
‘`isVailid:{`’ ⟨*Exp*⟩ ‘`}`’
‘`possMoves:{`’ ⟨*Exp*⟩ ‘`}`’
‘`outcome:{`’ ⟨*Outcome*⟩ ‘`}`’
‘`initialState:{`’ ⟨*InitState*⟩ ‘`}`’
‘`fromString:{`’ ⟨*Exp*⟩ ‘`}`’
‘`$`’ [⟨*Dec*⟩] ‘`$`’ (Custom declarations)

⟨*gamestate*⟩ ::= ‘`Board:{`’ ⟨*BoardDec*⟩
| ‘`Piece:{`’ ⟨*Dec*⟩
| ‘`Hand:{`’ ⟨*Dec*⟩
| ‘`Turn:{`’ ⟨*Dec*⟩
| ⟨*string*⟩‘`:{`’ ⟨*Dec*⟩

⟨*BoardDec*⟩ ::= ‘`{Matrix[`’⟨*int*⟩‘`][`’⟨*int*⟩‘`]}`’
| ‘`{Array[`’⟨*int*⟩‘`]`’
| ‘`<<`’ ⟨*Dec*⟩ ‘`>>`’ (Custom Type)

⟨*Outcome*⟩ ::= ‘`{winCondition:{`’ ⟨*Exp*⟩ ‘`}`’
‘`tieCondition:{`’ ⟨*Exp*⟩ ‘`}`’
‘`else:{`’ ⟨*Exp*⟩ ‘`}`’
| ‘`<<`’ ⟨*Exp*⟩ ‘`>>`’

⟨*InitState*⟩ ::= ‘`Board:`’ ⟨*BoardInitDec*⟩
‘`Turn:{`’ ⟨*Exp*⟩ ‘`}`’

⟨*BoardInitDec*⟩ ::= ‘`{ all`’ ⟨*Exp*⟩ ‘`}`’ (Initialize board to piece)
| ‘`{`’ ⟨*Exp*⟩ ‘`}`’ (Initialize board to List literal)
| ‘`<<`’ ⟨*Exp*⟩ ‘`>>`’ (Custom initialization func.)

⟨*Exp*⟩ ::= ⟨*Template Haskell Expression*⟩

⟨*Dec*⟩ ::= ⟨*Template Haskell Declaration*⟩