# Grammar for Language Describing Game Structures

Brooks MacLachlan

March 21, 2019

**Pretty-printed EBNF grammar:**

⟨*game*⟩ ::= ⟨*roundList*⟩ '`played by`' ⟨*teamList*⟩ '`wonBy`' ⟨*winConditionList*⟩

⟨*teamList*⟩ ::= (⟨*team*⟩ {'`,`' ⟨*team*⟩} | ⟨*playerList*⟩ | '`randomly divide`' ⟨*playerList*⟩ '`into`' ⟨*nameList*⟩) ['`all`' ⟨*attributeList*⟩]

⟨*team*⟩ ::= '`team`' ⟨*name*⟩ ⟨*playerList*⟩

⟨*playerList*⟩ ::= ⟨*player*⟩ {'`;`' ⟨*player*⟩}

⟨*player*⟩ ::= ⟨*name*⟩ ⟨*attributeList*⟩

⟨*attributeList*⟩ ::= ⟨*attribute*⟩ {'`,`' ⟨*attribute*⟩}

⟨*attribute*⟩ ::= ⟨*affiliation*⟩ | ⟨*score*⟩ [⟨*number*⟩] | ⟨*resource*⟩ [⟨*number*⟩]

⟨*affiliation*⟩ ::= 'affiliated with' ⟨*name*⟩

⟨*score*⟩ ::= '`score`' ⟨*name*⟩

⟨*resource*⟩ ::= '`resource`' ⟨*name*⟩

⟨*roundList*⟩ ::= ⟨*round*⟩ {'`;`' ⟨*round*⟩} [⟨*modifierList*⟩]

⟨*modifierList*⟩ ::= ⟨*modifier*⟩ {'`,`' ⟨*modifier*⟩}

⟨*modifier*⟩ ::= ⟨*roundReference*⟩ ('`before`' | '`after`') ⟨*phaseReference*⟩ '`insert`' ⟨*phase*⟩

⟨*roundReference*⟩ ::= '`round`' [⟨*number*⟩]

⟨*phaseReference*⟩ ::= '`phase`' [⟨*number*⟩]

⟨*round*⟩ ::= ⟨*phaseList*⟩ ['`repeated`' ⟨*number*⟩ '`times`']

⟨*phase*⟩ ::= ⟨*action*⟩ | ⟨*progression*⟩

⟨*phaseList*⟩ ::= ⟨*phase*⟩ {‘`,`’ ⟨*phase*⟩}

⟨*action*⟩ ::= ⟨*competition*⟩ | ⟨*decision*⟩

⟨*competition*⟩ ::= [‘`scored`’] [‘`team`’] ‘`competition between`’ ⟨*idList*⟩

⟨*decision*⟩ ::= [‘`self-included`’] ‘`vote by`’ ⟨*idList*⟩ ‘`between`’ ⟨*idList*⟩ [‘`tiebroken by`’ (⟨*phase*⟩)] | [‘`self-included`’] ‘`nomination of`’ ⟨*number*⟩ ‘`of`’ ⟨*idList*⟩ ‘`by`’ ⟨*identifier*⟩ | ‘`allocate`’ ⟨*resource*⟩ | [‘`self-included`’] ‘`directed vote by`’ ⟨*idList*⟩ ‘`between`’ ⟨*idList*⟩ | ‘`uses?`’ ⟨*identifier*⟩ ‘`then`’ ⟨*phaseList*⟩ [‘`otherwise`’ ⟨*phaseList*⟩]

⟨*progression*⟩ ::= (⟨*affiliationUpdate*⟩ | ⟨*scoreUpdate*⟩ | ⟨*resourceUpdate*⟩) ‘`for`’ ⟨*idList*⟩

⟨*affiliationUpdate*⟩ ::= ‘`elimination`’ | (‘`add`’ | ‘`remove`’) ⟨*affiliation*⟩ | [‘`evenly`’ | ‘`number preserving`’] ‘`swap`’ ⟨*affiliationList*⟩ [‘`adding`’ ⟨*affiliationList*⟩] | ‘`change`’ ⟨*affiliation*⟩ ‘`to`’ ⟨*affiliation*⟩ | ‘`merge`’ ⟨*affiliationList*⟩ [⟨*affiliation*⟩]

⟨*scoreUpdate*⟩ ::= (‘`increase`’ | ‘`decrease`’) ⟨*score*⟩ ‘`by`’ ⟨*value*⟩ | ‘`set`’ ⟨*score*⟩ ‘`to`’ ⟨*value*⟩

⟨*resourceUpdate*⟩ ::= (‘`gain`’ | ‘`expend`’) ⟨*value*⟩ ⟨*resource*⟩ | ‘`set`’ ⟨*resource*⟩ ‘`to`’ ⟨*value*⟩

⟨*value*⟩ ::= ⟨*number*⟩ | ⟨*resultReference*⟩ | ⟨*allocateReference*⟩ | ⟨*score*⟩ | ⟨*resource*⟩ | ⟨*voteReference*⟩

⟨*resultReference*⟩ ::= ‘`scored competition`’ [⟨*number*⟩]

⟨*allocateReference*⟩ ::= ‘`allocation`’ [⟨*number*⟩]

⟨*idList*⟩ ::= ⟨*identifier*⟩ {‘`,`’ ⟨*identifier*⟩} [‘`except`’ ⟨*idList*⟩]

⟨*identifier*⟩ ::= (‘`everyone`’ | ⟨*name*⟩ | ⟨*affiliation*⟩ | ⟨*idReference*⟩ | ‘`chance`’) [‘`*`’ ⟨*value*⟩]

⟨*idReference*⟩ ::= ⟨*compReference*⟩ (‘`winner`’ | ‘`loser`’) | ⟨*voteReference*⟩ (‘`majority`’ | ‘`minority`’) ‘`voted`’ | (‘`highest`’ | ‘`lowest`’) ⟨*score*⟩ | (‘`most`’ | ‘`least`’) ⟨*resource*⟩

⟨*compReference*⟩ ::= ‘`placed competition`’ [⟨*number*⟩]

⟨*voteReference*⟩ ::= ‘`vote`’ [⟨*number*⟩]

⟨*allocationReference*⟩ ::= '`allocation`' [⟨*number*⟩]

⟨*name*⟩ ::= ('`a`' | ... | '`z`' | '`A`' | ... | '`Z`') {'`a`' | ... | '`z`' | '`A`' | ... | '`Z`' | '`0`' | ... | '`9`'}

⟨*number*⟩ ::= ('`0`' | ... | '`9`')+

⟨*winCondition*⟩ ::= (⟨*score*⟩ | ⟨*resource*⟩) '`hits`' ⟨*number*⟩ | '`final competition`'
| ('`highest`' | '`lowest`') (⟨*score*⟩ | ⟨*resource*⟩) | ⟨*number*⟩ '`member jury vote`'

⟨*winConditionList*⟩ ::= ⟨*winCondition*⟩ {'`,`' ⟨*winCondition*⟩}

Notes:

- make grammar unambiguous
- I've restricted player attributes into 3 categories, with no opportunity for customization outside those three. Am I okay with this?
- affiliation used everywhere, get syntax right
- should allocate be a decision and progression or just progression? Both, but decision will automatically also do the progression. The progression exists for non-decision related resource updates, such as awarding a player more of a resource for winning a competition.
- combine score and resource? Implement them the same way, but keep them separate for nice syntax