% Author:

% Date: 03/08/2018

% knowledge base

hasBow(violin).

hasBow(viola).

hasBow(doublebass).

hasBow(cello).

hasStrings(violin).

hasStrings(viola).

hasStrings(guitar).

hasStrings(harp).

hasStrings(sitar).

hasStrings(banjo).

hasStrings(oud).

isWooden(violin).

isWooden(viola).

hasPegs(violin).

hasPegs(viola).

isPlucked(guitar).

isPlucked(harp).

isPlucked(sitar).

isPlucked(banjo).

isPlucked(oud).

plucked(musicbox).

plucked(jewsharp).

isShaken(tambouring).

isShaken(jinglebells).

isClapped(guban).

isClapped(cymbals).

has(clarinet,singlereed).

has(oboe,doublereed).

has(basson,doublereed).

has(harmonica,freereed).

has(reed\_organ,freereed).

has(X,singlereed):- typeOf(X,saxophone).

shapeOf(conical,talkingdrum).

shapeOf(conical,udu).

shapeOf(conical,djembe).

shapeOf(cylindrical,bassdrum).

shapeOf(cylindrical,drumset).

shapeOf(cylindrical,snaredrum).

shapeOf(cylindrical,multi-tenordrum).

shapeOf(cylindrical,tenordrum).

shapeOf(barrel,barreldrum).

typeOf(aacousticguitar,guitar).

typeOf(bassguitar,guitar).

typeOf(altosax,saxophone).

typeOf(sopranosax,saxophone).

typeOf(tenorsax,saxophone).

typeOf(bodhran,framedrums).

typeOf(gobletdrum,framedrums).

typeOf(X,tubulardrums):- shapeOf(conical,X).

typeOf(X,tubulardrums):- shapeOf(cylindrical,X).

typeOf(X,tubulardrums):- shapeOf(barrel,X).

classOf(flute,picollo).

classOf(flute,flutemain).

classOf(concussion,guban).

classOf(concussion,cymbals).

classOf(reedInstrument,X) :- has(X,singlereed).

classOf(reedInstrument,X) :- has(X,doublereed).

classOf(reedInstrument,X) :- has(X,freereed).

classOf(plucked,X):- plucked(X).

classOf(shaken,X):- isShaken(X).

classOf(concussion,X) :- isClapped(X).

classOf(drums,kettledrums).

classOf(drums,fictiondrums).

classOf(drums,X) :- typeOf(X,framedrums).

classOf(drums,X):- typeOf(X,tubulardrums).

classOf(grandpiano,babygrand).

classOf(grandpiano,petitegrand).

classOf(grandpiano,concertgrand).

classOf(grandpiano,livingroomgrand).

classOf(uprightpiano,spinet).

classOf(uprightpiano,console).

classOf(uprightpiano,studio).

classOf(uprightpiano, professionalpiano).

group(brass,trumpet).

group(brass,trombone).

group(brass,french-horn).

group(brass,bugle).

group(brass,cornet).

group(bowed,X):- hasBow(X).

group(plucked,X):- isPlucked(X).

group(plucked,X):- typeOf(X,guitar).

group(woodwind,X):- classOf(flute,X).

group(woodwind,X):- classOf(reedInstrument,X).

group(idiophones,xylophones).

group(idiophones,X) :- classOf(concussion,X).

group(idiophones,X) :- classOf(plucked,X).

group(idiophones,X) :- classOf(shaken,X).

group(membranophones,X) :- classOf(drums,X).

group(membranophones,X) :- classOf(kazoos,X).

group(piano,X) :- classOf(grandpiano,X).

group(piano,X) :- classOf(uprightpiano,X).

group(piano,X) :- classOf(electricpiano,X).

familyOf(string,X) :- group(bowed,X).

familyOf(string,X) :- group(plucked,X).

familyOf(string,X) :- hasStrings(X),isWooden(X),hasPegs(X).

familyOf(wind,X) :- group(woodwind,X).

familyOf(wind,X) :- group(brass,X).

familyOf(percussion,X) :- group(idiophones,X).

familyOf(percussion,X) :- group(membranophones,X).

familyOf(keyboard,X) :- group(piano,X).

instruments(X) :- familyOf(string,X).

instruments(X) :- familyOf(wind,X).

instruments(X) :- familyOf(keyboard,X).

instruments(X) :- familyOf(percussion,X).

plays\_classical(X):- group(piano,X); group(bowed,X); group(woodwind,X); group(brass,X).

plays\_jazz(X) :- typeOf(X,guitar); typeOf(X,saxophone); group(brass,X);

plays\_jazz(harp).

plays\_blues(X) :- shapeOf(cylindrical,X); typeOf(X,guitar); group(piano,X).

plays\_blues(harmonica).

plays\_blues(doublebass).

plays\_rock\_n\_roll(X) :- group(piano,X); typeOf(X,guitar); typeOf(X,saxophone); shapeOf(cylindrical,X).

played\_with(X,hand) :- familyOf(string,X); familyOf(percussion,X); familyOf(keyboard,X).

played\_with(X,mouth) :- familyOf(wind,X).

print\_one\_instr\_group :- group(X,Y),familyOf(Z,Y), write(Y),write(' is in the group of '),write(X), write(' Instruments of '), write(Z), write(' family'),nl.

print\_instr\_groups:- print\_one\_instr\_group, fail.

print\_instr\_groups.

print\_one\_instr\_family :- familyOf(X,Y), write(X),write(' is the family of '),write(Y),nl.

print\_instr\_families:- print\_one\_instr\_family, fail.

print\_instr\_families.

print\_all\_instruments :- instruments(X), write(X), nl, fail.

print\_all\_instruments.

% procedures to interact with the user

start :- display\_menu,

get\_from\_menu(Answer),

answer\_this(Answer).

display\_menu :- write('What do you want to do?'),nl,

write('1. Know different types of instruments and their families'),nl,

write('2. Know the family of a certain instrument'),nl,

write('3. Know the instruments under each group of a family'),nl,

write('4. Know the group a certain instrument belongs to'),nl,

write('5. Make a decision on which instrument to learn'),nl,

write('6. Get all the instruments in the ontology'),nl,

write('Type a number, 1 to 6 --- ').

get\_from\_menu(Answer) :- get(Code),

get0(\_),

interpret(Code,Answer),

!.

get\_from\_menu(Answer) :- nl,

put(7),

write('Select fron 1 to 6: '),

get\_from\_menu(Answer).

answer\_this(know\_instr\_fam) :- print\_instr\_families.

answer\_this(know\_fam\_instr) :- write('what instrument do you wish to know the family it belongs to?'),

read(Instrument),

familyOf(X,Instrument),

write(X),write(' is the family of '),write(Instrument),nl.

answer\_this(know\_grp\_instr) :- print\_instr\_groups.

answer\_this(know\_instr\_grp) :- write('what instrument do you wish to know the group it belongs to?'),

read(Instrument),

group(X,Instrument),

write(Instrument),write(' belongs to the '),write(X),write(' group'),nl.

answer\_this(decide\_instr) :- decision\_qst1,

get\_from\_menu(Answer),

decision\_qst2,

get\_from\_menu(Result),

you\_can\_learn(Answer,Result).

answer\_this(all\_instr) :- print\_all\_instruments.

interpret(49,know\_instr\_fam). /\* ASCII 49 = '1' \*/

interpret(50,know\_fam\_instr). /\* ASCII 50 = '2' \*/

interpret(51,know\_grp\_instr). /\* ASCII 51 = '3' \*/

interpret(52,know\_instr\_grp). /\* ASCII 52 = '4' \*/

interpret(53,decide\_instr). /\* ASCII 53 = '5' \*/

interpret(54,all\_instr). /\* ASCII 54 = '6' \*/

interpret(65,rock). /\* ASCII 65 = 'A' \*/

interpret(66,jazz). /\* ASCII 66 = 'B' \*/

interpret(67,blues). /\* ASCII 67 = 'C' \*/

interpret(68,classical). /\* ASCII 68 = 'D' \*/

interpret(71,hand). /\* ASCII 71 = 'G' \*/

interpret(72,mouth). /\* ASCII 72 = 'H' \*/

% decision questions

decision\_qst1 :- write('What genre of music do you want to play?'),nl,

write('A. Rock n Roll'),nl,

write('B. Jazz'),nl,

write('C. Blues'),nl,

write('D. Classical'),nl,

write('Type a letter in caps, A to D --- ').

decision\_qst2 :- write('How do you want to play?'),nl,

write('G. Hand'),nl,

write('H. Mouth'),nl,

write('Type a letter in caps, G or H --- ').

%decision kb

you\_can\_learn(rock,hand) :- plays\_rock\_n\_roll(X),played\_with(X,hand),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(rock,mouth) :- plays\_rock\_n\_roll(X),played\_with(X,mouth),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(rock,hand).

you\_can\_learn(rock,mouth).

you\_can\_learn(jazz,hand) :- plays\_jazz(X),played\_with(X,hand),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(jazz,mouth) :- plays\_jazz(X),played\_with(X,mouth),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(jazz,hand).

you\_can\_learn(jazz,mouth).

you\_can\_learn(blues,hand) :- plays\_blues(X),played\_with(X,hand),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(blues,mouth) :- plays\_blues(X),played\_with(X,mouth),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(blues,hand).

you\_can\_learn(blues,mouth).

you\_can\_learn(classical,hand) :- plays\_classical(X),played\_with(X,hand),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(classical,mouth) :- plays\_classical(X),played\_with(X,mouth),write('you can learn the '), write(X),nl,fail.

you\_can\_learn(classical,hand).

you\_can\_learn(classical,mouth).