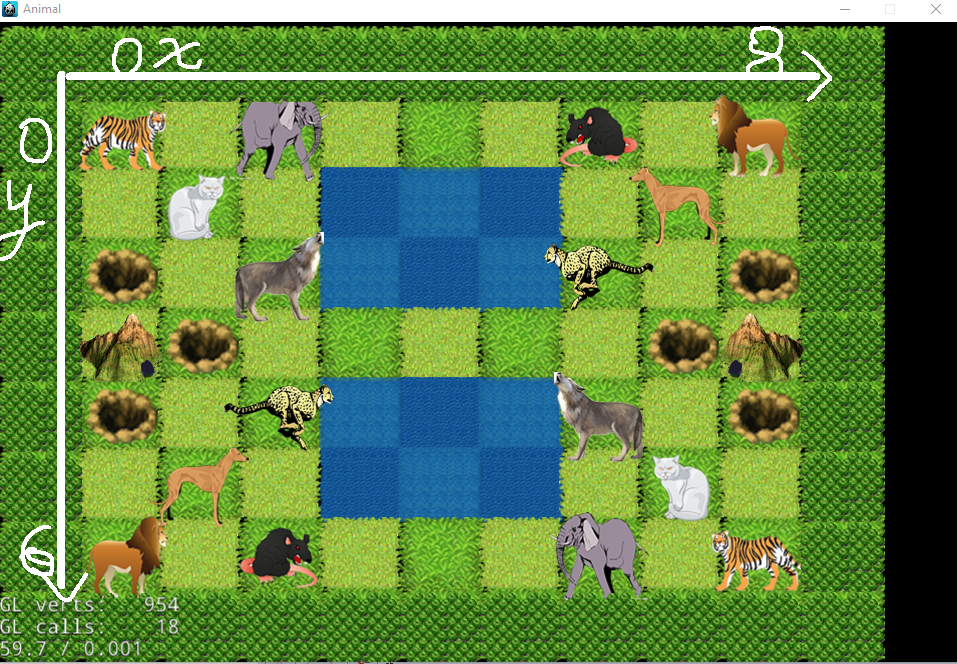
说明

2016-04-12 16:41:26

单机版已经做好，现在要做AI。

这是地图的坐标：



然而C++里一般是调用vec[x][y]这样形式的，也就是说，C++里一般x是向下增长的，y向右增长。代码是里统一按棋盘来的，例如，boardPieces[2][0]是上边的大象。

Player 是0或1，0是左侧，1是右侧

所有棋子都是指针, Pieces \*类型

PointXY类

Pieces类

Board类

Player类

PointXY类只有两个变量

x,y

Pieces类：

enum TypePiece {NIL,RAT,CAT,WOLF,DOG,LEOPARD,TIGER,LION,ELEPHANT};

getType(): 获取棋子类型，例如，判断某个棋子p是不是某一类型t，用

p->getType() == Pieces::t

getPlayer() ： 获取此棋子属于哪个玩家

getPositionBlock() ：获取此棋子位置

Board类：

enum TypeTerrain { NIL, DEN0, DEN1, TRAP, RIVER };

selected是已经选择的棋子，如果没有选择，则棋子为nul\_piece。

nul\_piece也是一个棋子，但是它的Type是NIL，玩家是-1。

getTypeTerrain(PointXY) ： 返回坐标的地形

moveChess(Pieces\* , PointXY dst) ：把棋子移动到dst

getPiece(PointXY p) : 返回坐标为p的棋子，如果是没棋子则返回nul\_piece

动态维护boardPieces.

Player类是一个AI类的接口，有两个函数，第一个函数是根据当前棋盘生成一个Move，第二个函数是给当前棋盘估分。

class Player{

public:

string name;

virtual Move genMove(const Board &board);

virtual float eval( const Board &board);

};

AI\_Min\_Max

class AI\_Min\_Max : public Player{

Move genMove(const Board &board);

float eval( const Board &board);

};

生成棋盘

监听点击

获取点击位置

availableMove(from,to) == 1?

如果selected是空，则赋值

如果selected非空，说明此点击是一个move

N

Y

Move

Move:

画面上改变selected的显示位置

游戏逻辑里

1更新Board类里的boardPieces

2更新棋子类坐标变量

例如一个棋子从[0,0] 移动到了 [1,0]， 则

1 Board类里的boardPieces[0][0]会为nul\_piece, boardPiece[1,0] = selected

2 selected的位置变为[1,0]

如果一个棋子从棋盘上消失，则不显示它但并不会销毁对象，因为可能我们还需要再把它放回去。