KTAB

Generated by Doxygen 1.8.6

Thu Jan 21 2016 04:38:24

Contents

1 Namespace Index **1**

1.1 Namespace List 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 File Index **7**

4.1 File List 7

5 Namespace Documentation **9**

5.1 DemoLeon Namespace Reference 9

5.1.1 Function Documentation 9

5.1.1.1 demoEUEcon 9

5.1.1.2 demoEUEcon 9

5.1.1.3 demoMaxEcon 9

5.1.1.4 demoMaxEcon 9

5.1.1.5 demoSetup 9

5.1.2 Variable Documentation 9

5.1.2.1 TolIFD 10

5.2 DemoMtch Namespace Reference 10

5.2.1 Function Documentation 10

5.2.1.1 demoDivideSweets 10

5.2.1.2 demoMaxSupport 10

5.2.1.3 demoMtchSUSN 10

5.2.1.4 equivMtchPstn 10

5.2.1.5 multiMtchSUSN 10

5.2.1.6 oneMtchSUSN 10

5.2.1.7 showMtchPstn 10

5.2.1.8 stableMtchState 10

5.3 KBase Namespace Reference 10

5.3.1 Typedef Documentation 12
5.3.1.1 VUI 12
5.3.1.2 W64 12
5.3.2 Enumeration Type Documentation 12
5.3.2.1 BigRAdjust 13
5.3.2.2 BigRRange 13
5.3.2.3 PCEModel 13
5.3.2.4 ReportingLevel 13
5.3.2.5 ThirdPartyCommit 13
5.3.2.6 VotingRule 13
5.3.2.7 VPModel 14
5.3.3 Function Documentation 14
5.3.3.1 antiLemke 14
5.3.3.2 bigRAName 14
5.3.3.3 bigRRName 14
5.3.3.4 crossSite 14
5.3.3.5 displayProgramEnd 14
5.3.3.6 displayProgramStart 14
5.3.3.7 dot 14
5.3.3.8 iMat 14
5.3.3.9 inv 14
5.3.3.10 joinH 14
5.3.3.11 joinV 14
5.3.3.12 lCorr 14
5.3.3.13 makePerp 14
5.3.3.14 maxAbs 14
5.3.3.15 mean 14
5.3.3.16 ndxMaxAbs 14
5.3.3.17 newChars 14
5.3.3.18 norm 14
5.3.3.19 operator∗ 14
5.3.3.20 operator∗ 14
5.3.3.21 operator+ 14
5.3.3.22 operator+ 14
5.3.3.23 operator- 15
5.3.3.24 operator- 15
5.3.3.25 operator/ 15
5.3.3.26 operator<< 15
5.3.3.27 operator<< 15
5.3.3.28 operator<< 15

5.3.3.29 operator<< 15
5.3.3.30 operator<< 15
5.3.3.31 operator<< 15
5.3.3.32 operator<< 15
5.3.3.33 pcmName 15
5.3.3.34 popBack 15
5.3.3.35 projBox 15
5.3.3.36 projPos 15
5.3.3.37 qrtc 15
5.3.3.38 qTrans 15
5.3.3.39 rescale 15
5.3.3.40 rotl 15
5.3.3.41 rotr 15
5.3.3.42 sameShape 15
5.3.3.43 sqr 15
5.3.3.44 stdv 15
5.3.3.45 sum 15
5.3.3.46 tpcName 15
5.3.3.47 trans 15
5.3.3.48 ueIndices 15
5.3.3.49 uiSeq 15
5.3.3.50 viABG 16
5.3.3.51 viBSHe96 16
5.3.3.52 vpmName 16
5.3.3.53 vrName 16
5.3.4 Variable Documentation 16
5.3.4.1 MASK32 16
5.3.4.2 MASK64 16
5.3.4.3 WordLength 16
5.4 KGraph Namespace Reference 16
5.5 MDemo Namespace Reference 16
5.5.1 Typedef Documentation 17
5.5.1.1 BVec 17
5.5.2 Function Documentation 17
5.5.2.1 demoDBObject 17
5.5.2.2 demoEMod 17
5.5.2.3 demoPCE 17
5.5.2.4 demoSpVSR 17
5.5.2.5 tbv 17
5.5.2.6 theta2D 17

5.5.2.7 thetaBV 17
5.6 Tetris Namespace Reference 17
5.6.1 Enumeration Type Documentation 17
5.6.1.1 SchemeShapes 17
5.6.1.2 SchemeWindows 18
5.6.1.3 TCode 18
5.6.2 Function Documentation 18
5.6.2.1 demoCoords 18
5.6.2.2 newChar 18
5.6.2.3 tetrisTimer 18
5.7 TXDemo Namespace Reference 18
5.7.1 Function Documentation 18
5.7.1.1 demoTX2 18
5.8 UDemo Namespace Reference 18
5.8.1 Typedef Documentation 19
5.8.1.1 BVec 19
5.8.2 Function Documentation 19
5.8.2.1 antiLemke 19
5.8.2.2 bsu 19
5.8.2.3 bvu 19
5.8.2.4 demoABG00 19
5.8.2.5 demoAntiLemke 19
5.8.2.6 demoEllipse 19
5.8.2.7 demoEllipseLVI 19
5.8.2.8 demoGA 19
5.8.2.9 demoGHC 19
5.8.2.10 demoMatrix 20
5.8.2.11 demoThreadLambda 20
5.8.2.12 demoThreadSynch 20
5.8.2.13 demoUIndices 20
5.8.2.14 demoVHC00 20
5.8.2.15 demoVHC01 20
5.8.2.16 demoVHC02 20
5.8.2.17 demoVHC03 20
5.8.2.18 eNorm 20
5.8.2.19 eUnitize 20
5.8.2.20 nProd 20
5.8.2.21 parallelMatrixMult 20
5.8.2.22 projEllipse 20
5.8.2.23 show 20

6 Class Documentation 21
6.1 KBase::Actor Class Reference 21
6.1.1 Constructor & Destructor Documentation 21
6.1.1.1 Actor 21
6.1.1.2 ∼Actor 21
6.1.2 Member Function Documentation 21
6.1.2.1 thirdPartyVoteSU 22
6.1.2.2 vote 22
6.1.2.3 vProbLittle 22
6.1.3 Member Data Documentation 22
6.1.3.1 desc 22
6.1.3.2 name 22
6.2 Tetris::Board Class Reference 22
6.2.1 Constructor & Destructor Documentation 23
6.2.1.1 Board 23
6.2.1.2 ∼Board 23
6.2.2 Member Function Documentation 23
6.2.2.1 clearLines 23
6.2.2.2 clearOneLine 23
6.2.2.3 drawBackground 23
6.2.2.4 drawCurrShape 23
6.2.2.5 drawFragments 23
6.2.2.6 drawShape 23
6.2.2.7 drawUnitSquare 23
6.2.2.8 emptyBoard 23
6.2.2.9 nFromIJ 23
6.2.2.10 placeShape 23
6.2.2.11 randomizeFragments 23
6.2.2.12 randomizeRow 23
6.2.2.13 resetCurrPiece 24
6.2.2.14 rotateDown 24
6.2.2.15 stepGame 24
6.2.2.16 testSDrop 24
6.2.2.17 testShape 24
6.2.2.18 tryHDrop 24
6.2.2.19 tryLMove 24
6.2.2.20 tryLRot 24
6.2.2.21 tryRMove 24
6.2.2.22 tryRRot 24
6.2.2.23 trySDrop 24

6.2.2.24 update 24

6.2.3 Member Data Documentation 24

6.2.3.1 clms 24

6.2.3.2 currI 24

6.2.3.3 currJ 24

6.2.3.4 currShape 24

6.2.3.5 nextShape 24

6.2.3.6 rows 24

6.3 KGraph::Canvas Class Reference 24

6.3.1 Constructor & Destructor Documentation 25

6.3.1.1 Canvas 25

6.3.1.2 ∼Canvas 25

6.3.2 Member Function Documentation 25

6.3.2.1 clearMaps 25

6.3.2.2 end 25

6.3.2.3 handle 25

6.3.2.4 onDrag 25

6.3.2.5 onKeyDown 25

6.3.2.6 onMove 26

6.3.2.7 onPush 26

6.3.2.8 onRelease 26

6.3.2.9 updateMaps 26

6.3.3 Member Data Documentation 26

6.3.3.1 pict 26

6.3.3.2 xMap 26

6.3.3.3 yMap 26

6.4 Tetris::ControlState Class Reference 26

6.4.1 Constructor & Destructor Documentation 26

6.4.1.1 ControlState 26

6.4.1.2 ControlState 26

6.4.2 Member Data Documentation 26

6.4.2.1 bg 27

6.4.2.2 gt 27

6.4.2.3 pc 27

6.4.2.4 rt 27

6.5 KGraph::CoordMap Class Reference 27

6.5.1 Constructor & Destructor Documentation 27

6.5.1.1 CoordMap 27

6.5.1.2 ∼CoordMap 27

6.5.2 Member Function Documentation 27

6.5.2.1 d2s 27
6.5.2.2 s2d 27
6.5.3 Member Data Documentation 27
6.5.3.1 ad 27
6.5.3.2 as 27
6.5.3.3 bd 27
6.5.3.4 bs 27
6.6 KBase::EModel< PT > Class Template Reference 28
6.6.1 Constructor & Destructor Documentation 28
6.6.1.1 EModel 28
6.6.1.2 ∼EModel 28
6.6.2 Member Function Documentation 28
6.6.2.1 nthOption 28
6.6.2.2 numOptions 28
6.6.2.3 setOptions 28
6.6.3 Member Data Documentation 28
6.6.3.1 enumOptions 28
6.6.3.2 theta 29
6.7 KBase::EPosition< PT > Class Template Reference 29
6.7.1 Constructor & Destructor Documentation 29
6.7.1.1 EPosition 29
6.7.1.2 ∼EPosition 29
6.7.2 Member Data Documentation 29
6.7.2.1 eMod 29
6.7.2.2 ndx 29
6.8 KBase::EState< PT > Class Template Reference 30
6.8.1 Constructor & Destructor Documentation 30
6.8.1.1 EState 30
6.8.1.2 ∼EState 30
6.8.2 Member Function Documentation 30
6.8.2.1 setAllAUtil 30
6.8.2.2 setValues 30
6.8.3 Member Data Documentation 30
6.8.3.1 actorVFn 30
6.8.3.2 getAUtils 31
6.9 KBase::GAOpt< GAP > Class Template Reference 31
6.9.1 Constructor & Destructor Documentation 32
6.9.1.1 GAOpt 32
6.9.1.2 ∼GAOpt 32
6.9.2 Member Function Documentation 32

6.9.2.1 crossPair 32
6.9.2.2 crossPop 32
6.9.2.3 cyclicApply 32
6.9.2.4 dropDups 32
6.9.2.5 fill 32
6.9.2.6 getNth 32
6.9.2.7 init 32
6.9.2.8 mutateOne 32
6.9.2.9 mutatePop 32
6.9.2.10 run 32
6.9.2.11 selectPop 32
6.9.2.12 show 32
6.9.2.13 sortPop 32
6.9.2.14 step 32
6.9.3 Member Data Documentation 32
6.9.3.1 cFrac 32
6.9.3.2 cross 32
6.9.3.3 equiv 33
6.9.3.4 eval 33
6.9.3.5 gpool 33
6.9.3.6 makeGene 33
6.9.3.7 mFrac 33
6.9.3.8 mutate 33
6.9.3.9 pSize 33
6.9.3.10 rng 33
6.9.3.11 showGene 33
6.10 KBase::GHCSearch< HCP > Class Template Reference 33
6.10.1 Constructor & Destructor Documentation 33
6.10.1.1 GHCSearch 33
6.10.1.2 ∼GHCSearch 33
6.10.2 Member Function Documentation 33
6.10.2.1 run 33
6.10.3 Member Data Documentation 33
6.10.3.1 eval 34
6.10.3.2 nghbrs 34
6.10.3.3 show 34
6.11 KBase::KException Class Reference 34
6.11.1 Constructor & Destructor Documentation 34
6.11.1.1 KException 34
6.11.1.2 ∼KException 34

6.11.2 Member Data Documentation 34
6.11.2.1 msg 34
6.12 KBase::KMatrix Class Reference 34
6.12.1 Constructor & Destructor Documentation 35
6.12.1.1 KMatrix 35
6.12.1.2 KMatrix 35
6.12.1.3 ∼KMatrix 35
6.12.2 Member Function Documentation 35
6.12.2.1 arrayInit 35
6.12.2.2 begin 35
6.12.2.3 begin 35
6.12.2.4 cbegin 35
6.12.2.5 cend 36
6.12.2.6 end 36
6.12.2.7 end 36
6.12.2.8 map 36
6.12.2.9 mapV 36
6.12.2.10 mPrintf 36
6.12.2.11 numC 36
6.12.2.12 numR 36
6.12.2.13 operator() 36
6.12.2.14 operator() 36
6.12.2.15 uniform 36
6.12.3 Friends And Related Function Documentation 36
6.12.3.1 inv 36
6.12.4 Member Data Documentation 36
6.12.4.1 clms 36
6.12.4.2 rows 36
6.12.4.3 vals 36
6.13 DemoLeon::LeonActor Class Reference 36
6.13.1 Constructor & Destructor Documentation 37
6.13.1.1 LeonActor 37
6.13.1.2 ∼LeonActor 37
6.13.2 Member Function Documentation 37
6.13.2.1 posUtil 37
6.13.2.2 randomize 37
6.13.2.3 setShareUtilScale 37
6.13.2.4 shareToUtil 37
6.13.2.5 vote 37
6.13.2.6 vote 37

6.13.3 Member Data Documentation 37
6.13.3.1 eMod 37
6.13.3.2 idNum 37
6.13.3.3 maxS 37
6.13.3.4 minS 38
6.13.3.5 refS 38
6.13.3.6 refU 38
6.13.3.7 vCap 38
6.13.3.8 vr 38
6.14 DemoLeon::LeonModel Class Reference 38
6.14.1 Constructor & Destructor Documentation 39
6.14.1.1 LeonModel 39
6.14.1.2 ∼LeonModel 39
6.14.2 Member Function Documentation 39
6.14.2.1 infsDegree 39
6.14.2.2 makeBaseYear 39
6.14.2.3 makeFTax 39
6.14.2.4 makeIOModel 39
6.14.2.5 monteCarloShares 39
6.14.2.6 randomFTax 39
6.14.2.7 stateDist 39
6.14.2.8 vaShares 39
6.14.2.9 xprtDemand 39
6.14.3 Friends And Related Function Documentation 39
6.14.3.1 LeonActor 39
6.14.4 Member Data Documentation 39
6.14.4.1 aL 40
6.14.4.2 bL 40
6.14.4.3 eps 40
6.14.4.4 L 40
6.14.4.5 M 40
6.14.4.6 maxSub 40
6.14.4.7 maxTax 40
6.14.4.8 N 40
6.14.4.9 posTol 40
6.14.4.10 rho 40
6.14.4.11 vas 40
6.14.4.12 x0 40
6.15 DemoLeon::LeonState Class Reference 40
6.15.1 Constructor & Destructor Documentation 41

6.15.1.1 LeonState 41
6.15.1.2 ∼LeonState 41
6.15.2 Member Function Documentation 41
6.15.2.1 doSUSN 41
6.15.2.2 equivNdx 41
6.15.2.3 pDist 41
6.15.2.4 setAllAUtil 41
6.15.2.5 stepSUSN 41
6.15.3 Member Data Documentation 41
6.15.3.1 eMod 41
6.16 KBase::Model Class Reference 41
6.16.1 Constructor & Destructor Documentation 43
6.16.1.1 Model 43
6.16.1.2 ∼Model 43
6.16.2 Member Function Documentation 43
6.16.2.1 actrNdx 43
6.16.2.2 addActor 43
6.16.2.3 addState 43
6.16.2.4 bigRfromProb 43
6.16.2.5 coalitions 43
6.16.2.6 condPCE 43
6.16.2.7 createTableSQL 43
6.16.2.8 demoSQLite 43
6.16.2.9 estNRA 43
6.16.2.10 markovPCE 43
6.16.2.11 nProd 43
6.16.2.12 probCE 43
6.16.2.13 run 43
6.16.2.14 scalarPCE 43
6.16.2.15 sqlAUtil 43
6.16.2.16 vote 43
6.16.2.17 vProb 43
6.16.2.18 vProb 43
6.16.2.19 vProb 43
6.16.3 Member Data Documentation 43
6.16.3.1 actrs 43
6.16.3.2 history 43
6.16.3.3 numAct 44
6.16.3.4 rng 44
6.16.3.5 scenName 44

6.16.3.6 smpDB 44
6.16.3.7 stop 44
6.17 DemoMtch::MtchActor Class Reference 44
6.17.1 Member Enumeration Documentation 45
6.17.1.1 PropModel 45
6.17.2 Constructor & Destructor Documentation 45
6.17.2.1 MtchActor 45
6.17.2.2 ∼MtchActor 45
6.17.3 Member Function Documentation 45
6.17.3.1 maxProbEUPstn 45
6.17.3.2 posUtil 45
6.17.3.3 rAct 45
6.17.3.4 randomize 45
6.17.3.5 rPos 45
6.17.3.6 vote 45
6.17.3.7 vote 45
6.17.4 Member Data Documentation 45
6.17.4.1 idNum 45
6.17.4.2 pMod 45
6.17.4.3 sCap 45
6.17.4.4 vals 45
6.17.4.5 vr 45
6.18 KBase::MtchGene Class Reference 46
6.18.1 Constructor & Destructor Documentation 46
6.18.1.1 MtchGene 46
6.18.1.2 ∼MtchGene 46
6.18.2 Member Function Documentation 46
6.18.2.1 copySelf 46
6.18.2.2 cross 46
6.18.2.3 equiv 47
6.18.2.4 mutate 47
6.18.2.5 print 47
6.18.2.6 randomize 47
6.18.2.7 setState 47
6.18.3 Member Data Documentation 47
6.18.3.1 actrs 47
6.18.3.2 pstns 47
6.19 DemoMtch::MtchModel Class Reference 47
6.19.1 Constructor & Destructor Documentation 48
6.19.1.1 MtchModel 48

6.19.1.2 ∼MtchModel 48
6.19.2 Member Function Documentation 48
6.19.2.1 randomMS 48
6.19.3 Member Data Documentation 48
6.19.3.1 numCat 48
6.19.3.2 numItm 48
6.20 KBase::MtchPstn Class Reference 48
6.20.1 Constructor & Destructor Documentation 49
6.20.1.1 MtchPstn 49
6.20.1.2 ∼MtchPstn 49
6.20.2 Member Function Documentation 49
6.20.2.1 neighbors 49
6.20.2.2 print 49
6.20.3 Member Data Documentation 49
6.20.3.1 match 49
6.20.3.2 numCat 49
6.20.3.3 numItm 49
6.21 DemoMtch::MtchState Class Reference 49
6.21.1 Constructor & Destructor Documentation 50
6.21.1.1 MtchState 50
6.21.1.2 ∼MtchState 50
6.21.2 Member Function Documentation 50
6.21.2.1 actrCaps 50
6.21.2.2 doBCN 50
6.21.2.3 doSUSN 50
6.21.2.4 equivNdx 50
6.21.2.5 pDist 50
6.21.2.6 setAllAUtil 50
6.21.2.7 stepBCN 50
6.21.2.8 stepSUSN 50
6.22 KGraph::Picture Class Reference 50
6.22.1 Constructor & Destructor Documentation 51
6.22.1.1 Picture 51
6.22.1.2 ∼Picture 51
6.22.2 Member Function Documentation 51
6.22.2.1 add 51
6.22.2.2 connect 51
6.22.2.3 update 51
6.22.2.4 update 51
6.22.3 Member Data Documentation 51

6.22.3.1 canvases 51
6.22.3.2 maxX 51
6.22.3.3 maxY 51
6.22.3.4 minH 51
6.22.3.5 minW 51
6.22.3.6 minX 51
6.22.3.7 minY 52
6.23 KBase::Position Class Reference 52
6.23.1 Constructor & Destructor Documentation 52
6.23.1.1 Position 52
6.23.1.2 ∼Position 52
6.23.2 Member Function Documentation 52
6.23.2.1 print 52
6.23.3 Friends And Related Function Documentation 52
6.23.3.1 operator<< 52
6.24 KBase::PRNG Class Reference 53
6.24.1 Constructor & Destructor Documentation 53
6.24.1.1 PRNG 53
6.24.1.2 ∼PRNG 53
6.24.2 Member Function Documentation 53
6.24.2.1 bits 53
6.24.2.2 setSeed 53
6.24.2.3 uniform 53
6.24.2.4 uniform 53
6.24.3 Member Data Documentation 53
6.24.3.1 mt 53
6.25 Tetris::PVCanvas Class Reference 53
6.25.1 Constructor & Destructor Documentation 54
6.25.1.1 PVCanvas 54
6.25.1.2 ∼PVCanvas 54
6.25.2 Member Function Documentation 54
6.25.2.1 draw 54
6.25.2.2 onDrag 54
6.25.2.3 onKeyDown 54
6.25.2.4 onMove 54
6.25.2.5 onPush 54
6.25.2.6 onRelease 55
6.26 Tetris::Shape Class Reference 55
6.26.1 Constructor & Destructor Documentation 55
6.26.1.1 Shape 55

6.26.1.2 Shape 55
6.26.2 Member Function Documentation 55
6.26.2.1 getName 55
6.26.2.2 getShape 55
6.26.2.3 lrot 56
6.26.2.4 maxX 56
6.26.2.5 maxY 56
6.26.2.6 minX 56
6.26.2.7 minY 56
6.26.2.8 rrot 56
6.26.2.9 setRandomShape 56
6.26.2.10 setShape 56
6.26.2.11 showCoords 56
6.26.2.12 x 56
6.26.2.13 y 56
6.26.3 Member Data Documentation 56
6.26.3.1 idNum 56
6.26.3.2 shapeCounter 56
6.27 MDemo::SQLDB Class Reference 56
6.27.1 Constructor & Destructor Documentation 56
6.27.1.1 SQLDB 56
6.27.1.2 ∼SQLDB 56
6.27.2 Member Function Documentation 56
6.27.2.1 close 57
6.27.2.2 open 57
6.27.2.3 query 57
6.28 KBase::State Class Reference 57
6.28.1 Constructor & Destructor Documentation 58
6.28.1.1 State 58
6.28.1.2 ∼State 58
6.28.2 Member Function Documentation 58
6.28.2.1 addPstn 58
6.28.2.2 clear 58
6.28.2.3 equivNdx 58
6.28.2.4 pDist 58
6.28.2.5 randomizeUtils 58
6.28.2.6 setAllAUtil 58
6.28.2.7 setAUtil 58
6.28.2.8 setOneAUtil 58
6.28.2.9 setUENdx 58

6.28.3 Member Data Documentation 58
6.28.3.1 aUtil 58
6.28.3.2 eIndices 58
6.28.3.3 model 58
6.28.3.4 pstns 58
6.28.3.5 step 58
6.28.3.6 uIndices 58
6.28.3.7 uProb 58
6.29 Tetris::TApp Class Reference 59
6.29.1 Constructor & Destructor Documentation 60
6.29.1.1 TApp 60
6.29.1.2 ∼TApp 60
6.29.2 Member Function Documentation 60
6.29.2.1 applyColorScheme 60
6.29.2.2 applyControlState 60
6.29.2.3 color 60
6.29.2.4 newGame 60
6.29.2.5 pause 60
6.29.2.6 processKey 60
6.29.2.7 quit 60
6.29.2.8 resume 60
6.29.2.9 run 60
6.29.2.10 scoreFn 60
6.29.2.11 setColorScheme 60
6.29.2.12 setDt 60
6.29.2.13 setLevel 60
6.29.2.14 setRandom 60
6.29.2.15 setRC 60
6.29.2.16 stepGame 60
6.29.3 Member Data Documentation 60
6.29.3.1 board 61
6.29.3.2 clms 61
6.29.3.3 colors 61
6.29.3.4 defaultClms 61
6.29.3.5 defaultLevel 61
6.29.3.6 defaultRows 61
6.29.3.7 defaultRP 61
6.29.3.8 dt 61
6.29.3.9 level 61
6.29.3.10 lineCount 61

6.29.3.11 maxPlayTime 61
6.29.3.12 paused 61
6.29.3.13 playTime 61
6.29.3.14 randomPlacement 61
6.29.3.15 rng 61
6.29.3.16 rows 61
6.29.3.17 score 61
6.29.3.18 theApp 61
6.30 UDemo::TargetedBV Class Reference 61
6.30.1 Constructor & Destructor Documentation 62
6.30.1.1 TargetedBV 62
6.30.1.2 ∼TargetedBV 62
6.30.2 Member Function Documentation 62
6.30.2.1 cross 62
6.30.2.2 equiv 62
6.30.2.3 evaluate 62
6.30.2.4 getTarget 62
6.30.2.5 hDist 62
6.30.2.6 mutate 62
6.30.2.7 randomBV 62
6.30.2.8 randomize 62
6.30.2.9 setTarget 62
6.30.2.10 show 62
6.30.2.11 showBits 62
6.30.2.12 tblEval 62
6.30.3 Member Data Documentation 63
6.30.3.1 bits 63
6.30.3.2 target 63
6.31 Tetris::TCanvas Class Reference 63
6.31.1 Constructor & Destructor Documentation 63
6.31.1.1 TCanvas 63
6.31.1.2 ∼TCanvas 63
6.31.2 Member Function Documentation 63
6.31.2.1 draw 63
6.31.2.2 onDrag 64
6.31.2.3 onKeyDown 64
6.31.2.4 onMove 64
6.31.2.5 onPush 64
6.31.2.6 onRelease 64
6.32 MDemo::TwoDPoint Struct Reference 64

6.32.1 Constructor & Destructor Documentation 64

6.32.1.1 TwoDPoint 64

6.32.1.2 TwoDPoint 64

6.32.1.3 ∼TwoDPoint 64

6.32.2 Member Data Documentation 64

6.32.2.1 x 65

6.32.2.2 y 65

6.33 KBase::VctrPstn Class Reference 65

6.33.1 Constructor & Destructor Documentation 65

6.33.1.1 VctrPstn 65

6.33.1.2 VctrPstn 65

6.33.1.3 VctrPstn 65

6.33.1.4 ∼VctrPstn 65

6.33.2 Member Function Documentation 65

6.33.2.1 print 65

6.34 KBase::VHCSearch Class Reference 66

6.34.1 Constructor & Destructor Documentation 66

6.34.1.1 VHCSearch 66

6.34.1.2 ∼VHCSearch 66

6.34.2 Member Function Documentation 66

6.34.2.1 run 66

6.34.2.2 vn1 66

6.34.2.3 vn2 66

6.34.3 Member Data Documentation 66

6.34.3.1 eval 66

6.34.3.2 nghbrs 66

6.34.3.3 report 66

6.35 MDemo::ZActor Class Reference 67

6.35.1 Constructor & Destructor Documentation 67

6.35.1.1 ZActor 67

6.35.1.2 ∼ZActor 67

6.35.2 Member Function Documentation 67

6.35.2.1 posUtil 67

6.35.2.2 vote 67

6.35.2.3 vote 67

7 File Documentation 69

7.1 board.cpp File Reference 69

7.2 board.h File Reference 69

7.3 demo.cpp File Reference 70

7.3.1 Function Documentation 70
7.3.1.1 main 70
7.4 demo.cpp File Reference 70
7.4.1 Function Documentation 71
7.4.1.1 main 71
7.5 demo.h File Reference 71
7.6 demo.h File Reference 71
7.7 demoleon.cpp File Reference 72
7.7.1 Function Documentation 72
7.7.1.1 main 72
7.8 demoleon.h File Reference 73
7.9 demomtch.cpp File Reference 73
7.9.1 Function Documentation 74
7.9.1.1 main 74
7.10 demomtch.h File Reference 74
7.11 edemo.cpp File Reference 74
7.12 edemo.h File Reference 75
7.13 emodel.cpp File Reference 75
7.14 emodel.h File Reference 76
7.15 gaopt.cpp File Reference 76
7.16 gaopt.h File Reference 77
7.17 hcsearch.cpp File Reference 77
7.18 hcsearch.h File Reference 77
7.19 kgraph.cpp File Reference 78
7.20 kgraph.h File Reference 78
7.21 kmatrix.cpp File Reference 79
7.22 kmatrix.h File Reference 79
7.23 kmodel.cpp File Reference 80
7.24 kmodel.h File Reference 81
7.24.1 Function Documentation 82
7.24.1.1 operator== 82
7.25 kmodelsql.cpp File Reference 82
7.26 kposition.cpp File Reference 82
7.26.1 Function Documentation 82
7.26.1.1 operator== 82
7.27 kstate.cpp File Reference 82
7.28 kutils.cpp File Reference 83
7.29 kutils.h File Reference 83
7.30 prng.cpp File Reference 84
7.31 prng.h File Reference 84

7.32 pvcanvas.cpp File Reference 85
7.33 pvcanvas.h File Reference 85
7.34 shape.cpp File Reference 86
7.35 shape.h File Reference 86
7.36 sqlitedemo.cpp File Reference 86
7.37 sqlitedemo.h File Reference 87
7.38 tcanvas.cpp File Reference 87
7.39 tcanvas.h File Reference 87
7.40 tinyxml2demo.cpp File Reference 88
7.41 tinyxml2demo.h File Reference 88
7.42 tmain.cpp File Reference 88
7.42.1 Function Documentation 89
7.42.1.1 main 89
7.43 tmain.h File Reference 89
7.44 tutils.h File Reference 90
7.45 vimcp.cpp File Reference 90
7.46 vimcp.h File Reference 91
7.47 zactor.cpp File Reference 91
7.48 zactor.h File Reference 92

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- KBase::Actor . . . 21
  - DemoLeon::LeonActor . . . 36
  - DemoMtch::MtchActor . . . 44
  - MDemo::ZActor . . . 67
- Tetris::ControlState . . . 26
- KGraph::CoordMap . . . 27
- Fl_Box
  - KGraph::Canvas . . . 24
    - Tetris::PVCanvas . . . 53
    - Tetris::TCanvas . . . 63
- KBase::GAOpt< GAP > . . . 31
- KBase::GHCSearch< HCP > . . . 33
- KBase::KException . . . 34
- KBase::KMatrix . . . 34
  - KBase::VctrPstn . . . 65
- KBase::Model . . . 41
  - DemoLeon::LeonModel . . . 38
  - DemoMtch::MtchModel . . . 47
  - KBase::EModel< PT > . . . 28
- KGraph::Picture . . . 50
  - Tetris::Board . . . 22
- KBase::Position . . . 52
  - KBase::EPosition< PT > . . . 29
  - KBase::MtchPstn . . . 48
    - KBase::MtchGene . . . 46
  - KBase::VctrPstn . . . 65
- KBase::PRNG . . . 53
- Tetris::Shape . . . 55
- MDemo::SQLDB . . . 56
- KBase::State . . . 57
  - DemoLeon::LeonState . . . 40
  - DemoMtch::MtchState . . . 49
  - KBase::EState< PT > . . . 30
- Tetris::TApp . . . 59
- UDemo::TargetedBV . . . 61
- MDemo::TwoDPoint . . . 64
- KBase::VHCSearch . . . 66

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

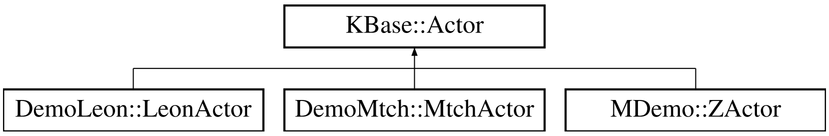
KBase::Actor . . . 21
Tetris::Board . . . 22
KGraph::Canvas . . . 24
Tetris::ControlState . . . 26
KGraph::CoordMap . . . 27
KBase::EModel< PT > . . . 28
KBase::EPosition< PT > . . . 29
KBase::EState< PT > . . . 30
KBase::GAOpt< GAP > . . . 31
KBase::GHCSearch< HCP > . . . 33
KBase::KException . . . 34
KBase::KMatrix . . . 34
DemoLeon::LeonActor . . . 36
DemoLeon::LeonModel . . . 38
DemoLeon::LeonState . . . 40
KBase::Model . . . 41
DemoMtch::MtchActor . . . 44
KBase::MtchGene . . . 46
DemoMtch::MtchModel . . . 47
KBase::MtchPstn . . . 48
DemoMtch::MtchState . . . 49
KGraph::Picture . . . 50
KBase::Position . . . 52
KBase::PRNG . . . 53
Tetris::PVCanvas . . . 53
Tetris::Shape . . . 55
MDemo::SQLDB . . . 56
KBase::State . . . 57
Tetris::TApp . . . 59
UDemo::TargetedBV . . . 61
Tetris::TCanvas . . . 63
MDemo::TwoDPoint . . . 64
KBase::VctrPstn . . . 65
KBase::VHCSearch . . . 66
MDemo::ZActor . . . 67

Chapter 3

Class Documentation

3.1 KBase::Actor Class Reference

Inheritance diagram for KBase::Actor:



Public Member Functions

- **Actor** (string n, string d)
- virtual double **vote** (unsigned int p1, unsigned int p2, const State ∗st) const =0

Static Public Member Functions

- static double **thirdPartyVoteSU** (double wk, VotingRule vr, ThirdPartyCommit comm, double pik, double pjk, double uki, double ukj, double ukk)
- static double **vProbLittle** (VotingRule vr, double wn, double uni, double unj, double contrib_i_ij, double contrib_j_ij)

Public Attributes

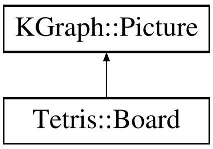
- string **name** = "GA"
- string **desc** = "Generic Actor"

The documentation for this class was generated from the following files:

- kmodel.h
- kmodel.cpp

3.2 Tetris::Board Class Reference

Inheritance diagram for Tetris::Board:



Public Member Functions

- **Board** (unsigned int r, unsigned int c)
- void **randomizeFragments** (double f)
- void **rotateDown** ()
- virtual void **update** (Canvas ∗c) const
- unsigned int **nFromIJ** (int i, int j) const
- bool **resetCurrPiece** ()
- unsigned int **clearLines** ()
- unsigned int **stepGame** ()
- bool **tryLRot** ()
- bool **tryRRot** ()
- bool **tryLMove** ()
- bool **tryRMove** ()
- bool **testSDrop** ()
- bool **trySDrop** ()
- bool **tryHDrop** ()
- void **drawShape** (int i, int j, Canvas ∗cnvs) const
- void **drawUnitSquare** (Fl_Color clr1, int i, int j, bool dotP, Fl_Color clr2, Canvas ∗cnvs) const

Public Attributes

- unsigned int **rows** = 0
- unsigned int **clms** = 0
- Shape **currShape** = Shape()
- int **currI** = 0
- int **currJ** = 0
- Shape **nextShape** = Shape()

Protected Member Functions

- void **drawBackground** (Canvas ∗c) const
- void **drawCurrShape** (Canvas ∗c) const
- void **drawFragments** (Canvas ∗c) const
- bool **testShape** (Shape s, int i, int j) const
- void **randomizeRow** (unsigned int i)
- vector< TCode > **emptyBoard** () const
- void **placeShape** (Shape s, int i, int j)
- bool **clearOneLine** (const unsigned int i)

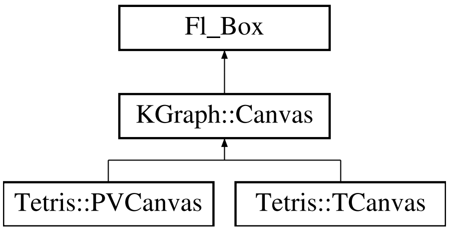
Additional Inherited Members

The documentation for this class was generated from the following files:

- board.h
- board.cpp

3.3 KGraph::Canvas Class Reference

Inheritance diagram for KGraph::Canvas:



Public Member Functions

- Canvas (int x, int y, int w, int h, const char ∗l=0)

  Abstract base class.
- void **end** ()
- void **updateMaps** ()
- void **clearMaps** ()
- virtual int **handle** (int ev)
- virtual void **onMove** (int x, int y)
- virtual void **onDrag** (int x, int y)
- virtual void **onPush** (int x, int y, int b)
- virtual void **onRelease** (int x, int y, int b)
- virtual void **onKeyDown** (int x, int y, int k)

Public Attributes

- Picture ∗ **pict** = nullptr
- CoordMap ∗ **xMap** = nullptr
- CoordMap ∗ **yMap** = nullptr

The documentation for this class was generated from the following files:

- kgraph.h
- kgraph.cpp

3.4 Tetris::ControlState Class Reference

Public Member Functions

- **ControlState** (unsigned int bv, unsigned int pv, unsigned int gv, unsigned int rv)

Public Attributes

- unsigned int **bg** = 1
- unsigned int **pc** = 2
- unsigned int **gt** = 1
- unsigned int **rt** = 0

The documentation for this class was generated from the following file:

- tmain.h

3.5 KGraph::CoordMap Class Reference

Public Member Functions

- **CoordMap** (int s1, double d1, int s2, double d2)
- int **d2s** (double d)
- double **s2d** (int s)

Protected Attributes

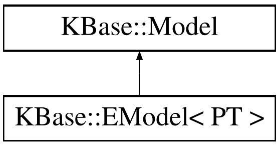
- double **as** = 0
- double **bs** = 0
- double **ad** = 0
- double **bd** = 0

The documentation for this class was generated from the following files:

- kgraph.h
- kgraph.cpp

3.6 KBase::EModel< PT > Class Template Reference

Inheritance diagram for KBase::EModel< PT >:



Public Member Functions

- **EModel** (PRNG ∗r, string d="")
- void **setOptions** ()
- unsigned int **numOptions** () const
- PT ∗ **nthOption** (unsigned int i) const

Public Attributes

- function< vector< PT ∗ >)> **enumOptions** = nullptr

Protected Attributes

- vector< PT ∗ > **theta** = {}

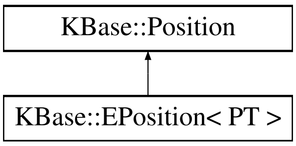
Additional Inherited Members

The documentation for this class was generated from the following files:

- emodel.h
- emodel.cpp

3.7 KBase::EPosition< PT > Class Template Reference

Inheritance diagram for KBase::EPosition< PT >:



Public Member Functions

- **EPosition** (EModel< PT > ∗m, int n)

Protected Attributes

- EModel< PT > ∗ **eMod** = nullptr
- int **ndx** = -1

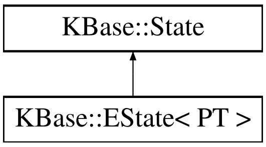
Additional Inherited Members

The documentation for this class was generated from the following files:

- emodel.h
- emodel.cpp

3.8 KBase::EState< PT > Class Template Reference

Inheritance diagram for KBase::EState< PT >:



Public Member Functions

- **EState** (EModel< PT > ∗mod)
- void **setValues** ()

Protected Member Functions

- void **setAllAUtil** (ReportingLevel rl)

Protected Attributes

- function< vector< KMatrix >)> **getAUtils** = nullptr
- function< vector< double > unsigned int j, const EModel < PT > ∗)> **actorVFn** = nullptr

Additional Inherited Members

The documentation for this class was generated from the following files:

- emodel.h
- emodel.cpp

3.9 KBase::GAOpt< GAP > Class Template Reference

Public Member Functions

- **GAOpt** (unsigned int s)
- void **init** (vector< GAP ∗ > ipop)
- void **fill** (PRNG ∗rng)
- void **run** (PRNG ∗rng, double c, double m, unsigned int maxI, double sTh, unsigned int maxS, ReportingLevel srl, unsigned int &iter, unsigned int &sIter)
- tuple< double, GAP ∗ > **getNth** (unsigned int n)
- void **show** ()
- void **sortPop** ()

Public Attributes

- function< tuple< GAP ∗, GAP ∗ > const GAP ∗g1, const GAP ∗g2, PRNG ∗rng)> **cross** = nullptr
- function< GAP ∗(const GAP ∗g1, PRNG ∗rng)> **mutate** = nullptr
- function< double(const GAP ∗g1)> **eval** = nullptr
- function< void(const GAP ∗)> **showGene** = nullptr
- function< GAP ∗(PRNG ∗rng)> **makeGene** = nullptr
- function< bool(const GAP ∗g1, const GAP ∗g2)> **equiv** = nullptr

Protected Member Functions

- void **step** ()
- void **mutatePop** ()
- void **crossPop** ()
- void **dropDups** ()
- void **selectPop** ()
- GAP ∗ **mutateOne** (const GAP ∗g1, PRNG ∗rng)
- tuple< GAP ∗, GAP ∗ > **crossPair** (const GAP ∗g1, const GAP ∗g2, PRNG ∗rng)
- void **cyclicApply** (function< void(unsigned int i)> fn, double f)

Protected Attributes

- vector< tuple< double, GAP ∗ > > **gpool** = {}
- unsigned int **pSize** = 0
- double **cFrac** = 1.0
- double **mFrac** = 0.5
- PRNG ∗ **rng** = nullptr

The documentation for this class was generated from the following file:

- gaopt.h

3.10 KBase::GHCSearch< HCP > Class Template Reference

Public Member Functions

- tuple< double, HCP, unsigned int, unsigned int > **run** (HCP p0, ReportingLevel srl, unsigned int iMax, unsigned int sMax, double sTol)

Public Attributes

- function< double(const HCP)> **eval** = nullptr
- function< vector< HCP >const HCP)> **nghbrs** = nullptr
- function< void(const HCP)> **show** = nullptr

The documentation for this class was generated from the following file:

- hcsearch.h

3.11 KBase::KException Class Reference

Public Member Functions

- **KException** (string m)

Public Attributes

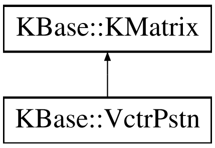
- string **msg** =""

The documentation for this class was generated from the following files:

- kutils.h
- kutils.cpp

3.12 KBase::KMatrix Class Reference

Inheritance diagram for KBase::KMatrix:



Public Member Functions

- **KMatrix** (unsigned int nr, unsigned int nc, double iv=0.0)
- double **operator()** (unsigned int i, unsigned int j) const
- double & **operator()** (unsigned int i, unsigned int j)
- void **mPrintf** (string) const
- unsigned int **numR** () const
- unsigned int **numC** () const

- vector< double >::iterator **begin** ()
- vector< double >::iterator **end** ()
- vector< double >::const_iterator **cbegin** ()
- vector< double >::const_iterator **cend** ()
- vector< double >::const_iterator **begin** () const
- vector< double >::const_iterator **end** () const

Static Public Member Functions

- static KMatrix **uniform** (PRNG ∗rng, unsigned int nr, unsigned int nc, double a, double b)
- static KMatrix **map** (function< double(unsigned int i, unsigned int j)> f, unsigned int nr, unsigned int nc)
- static void **mapV** (function< void(unsigned int i, unsigned int j)> f, unsigned int nr, unsigned int nc)
- static KMatrix **arrayInit** (const double mv[ ], const unsigned int &rows, const unsigned int &clms)

Protected Attributes

- unsigned int **rows** = 0
- unsigned int **clms** = 0
- vector< double > **vals** = vector<double>()

Friends

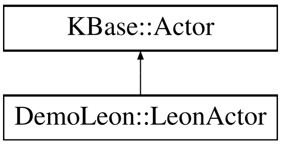
- KMatrix **inv** (const KMatrix &m)

The documentation for this class was generated from the following files:

- kmatrix.h
- kmatrix.cpp

3.13 DemoLeon::LeonActor Class Reference

Inheritance diagram for DemoLeon::LeonActor:



Public Member Functions

- **LeonActor** (string n, string d, LeonModel ∗em, unsigned int id)
- double **vote** (unsigned int p1, unsigned int p2, const State ∗st) const
- virtual double **vote** (const Position ∗ap1, const Position ∗ap2) const
- double **posUtil** (const Position ∗ap1) const
- void **randomize** (PRNG ∗rng)
- void **setShareUtilScale** (const KMatrix &runs)
- double **shareToUtil** (double gdpShare) const

Public Attributes

- const LeonModel ∗ **eMod** = nullptr
- unsigned int **idNum** = 0
- KMatrix **vCap** = KMatrix()
- VotingRule **vr** = VotingRule::Proportional
- double **minS** = 0
- double **refS** = 0.5
- double **refU** = 0.5
- double **maxS** = 1

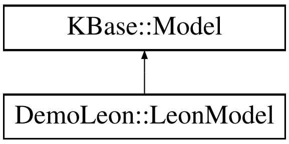
Additional Inherited Members

The documentation for this class was generated from the following files:

- demoleon.h
- demoleon.cpp

3.14 DemoLeon::LeonModel Class Reference

Inheritance diagram for DemoLeon::LeonModel:



Public Member Functions

- **LeonModel** (PRNG ∗r, string d="")
- tuple< KMatrix, KMatrix, KMatrix, KMatrix > **makeBaseYear** (unsigned int numF, unsigned int numCG, unsigned int numS, PRNG ∗rng)
- void **makeIOModel** (const KMatrix &trns, const KMatrix &rev, const KMatrix &xprt, const KMatrix &cons, PRNG ∗rng)
- KMatrix **xprtDemand** (const KBase::KMatrix &tau) const
- KMatrix **randomFTax** (PRNG ∗rng)
- KMatrix **makeFTax** (const KBase::KMatrix &tax) const
- double **infsDegree** (const KMatrix &tax) const
- KMatrix **vaShares** (const KMatrix &tax, bool normalizeSharesP) const
- KMatrix **monteCarloShares** (unsigned int nRuns, KBase::PRNG ∗rng)

Static Public Member Functions

- static double **stateDist** (const LeonState ∗s1, const LeonState ∗s2)

Public Attributes

- double posTol = 1E-5

  how close together positions must be to be considered equivalent

Protected Attributes

- unsigned int **L** = 0
- unsigned int **M** = 0
- unsigned int **N** = 0
- double **maxSub** = 0.5
- double **maxTax** = 0.5
- KMatrix **x0** = KMatrix()
- KMatrix **eps** = KMatrix()
- KMatrix **aL** = KMatrix()
- KMatrix **bL** = KMatrix()
- KMatrix **rho** = KMatrix()
- KMatrix **vas** = KMatrix()

Friends

- class **LeonActor**

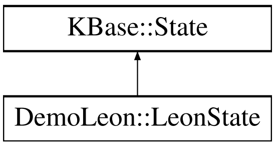
Additional Inherited Members

The documentation for this class was generated from the following files:

- demoleon.h
- demoleon.cpp

3.15 DemoLeon::LeonState Class Reference

Inheritance diagram for DemoLeon::LeonState:



Public Member Functions

- **LeonState** (LeonModel ∗em)
- virtual tuple< KMatrix, VUI > **pDist** (int persp) const
- LeonState ∗ **stepSUSN** ()

Public Attributes

- const LeonModel ∗ **eMod** = nullptr

Protected Member Functions

- LeonState ∗ **doSUSN** (ReportingLevel rl) const
- virtual bool equivNdx (unsigned int i, unsigned int j) const
- void **setAllAUtil** (ReportingLevel rl)

Additional Inherited Members

3.15.1 Member Function Documentation

3.15.1.1 bool DemoLeon::LeonState::equivNdx (unsigned int *i*, unsigned int *j*) const [protected], [virtual]

Compare two actual positions in the current state

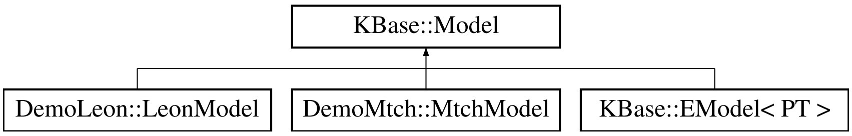
Implements KBase::State.

The documentation for this class was generated from the following files:

- demoleon.h
- demoleon.cpp

3.16 KBase::Model Class Reference

Inheritance diagram for KBase::Model:



Public Member Functions

- **Model** (PRNG ∗r, string d)
- void **run** ()
- virtual unsigned int **addActor** (Actor ∗a)
- int **actrNdx** (const Actor ∗a) const
- int **addState** (State ∗s)
- void **sqlAUtil** (unsigned int t)

Static Public Member Functions

- static double **nProd** (double x, double y)
- static double **vote** (VotingRule vr, double wi, double uij, double uik)
- static KMatrix **coalitions** (function< double(unsigned int ak, unsigned int pi, unsigned int pj)> vfn, unsigned int numAct, unsigned int numOpt)
- static KMatrix **vProb** (VPModel vpm, const KMatrix &c)
- static KMatrix **vProb** (VotingRule vr, VPModel vpm, const KMatrix &w, const KMatrix &u)
- static KMatrix **probCE** (PCEModel pcm, const KMatrix &pv)
- static KMatrix **scalarPCE** (unsigned int numAct, unsigned int numOpt, const KMatrix &w, const KMatrix &u, VotingRule vr, VPModel vpm, ReportingLevel rl)
- static void **demoSQLite** ()
- static KMatrix **bigRfromProb** (const KMatrix &p, BigRRange rr)
- static double **estNRA** (double rh, double ri, BigRAdjust ra)

Public Attributes

- function< bool(unsigned int iter, const State ∗s)> **stop** = nullptr
- vector< Actor ∗ > **actrs** = {}
- unsigned int **numAct** = 0
- PRNG ∗ **rng** = nullptr
- vector< State ∗ > **history** = {}

Static Protected Member Functions

- static string **createTableSQL** (unsigned int tn)
- static tuple< double, double > **vProb** (VPModel vpm, const double s1, const double s2)
- static KMatrix **markovPCE** (const KMatrix &pv)
- static KMatrix **condPCE** (const KMatrix &pv)

Protected Attributes

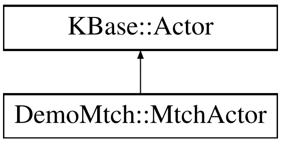
- sqlite3 ∗ **smpDB** = nullptr
- string **scenName** = "Scen"

The documentation for this class was generated from the following files:

- kmodel.h
- kmodel.cpp
- kmodelsql.cpp

3.17 DemoMtch::MtchActor Class Reference

Inheritance diagram for DemoMtch::MtchActor:



Public Types

- enum **PropModel** { **ExpUtil**, **Probability**, **AgreeUtil** }

Public Member Functions

- **MtchActor** (string n, string d)
- double **vote** (unsigned int p1, unsigned int p2, const State ∗st) const
- virtual double **vote** (const Position ∗ap1, const Position ∗ap2) const
- double **posUtil** (const Position ∗ap1) const
- void **randomize** (PRNG ∗rng, double minCap, double maxCap, unsigned int id, unsigned int numI)
- tuple< double, MtchPstn > **maxProbEUPstn** (PropModel pm, const MtchState ∗mst) const

Static Public Member Functions

- static MtchPstn ∗ **rPos** (unsigned int numI, unsigned int numA, PRNG ∗rng)
- static MtchActor ∗ **rAct** (unsigned int numI, double minCap, double maxCap, PRNG ∗rng, unsigned int i)

Public Attributes

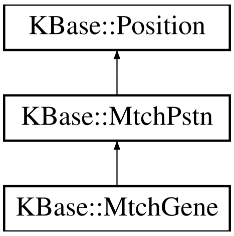
- unsigned int **idNum** = 0
- VotingRule **vr** = VotingRule::Proportional
- PropModel **pMod** = PropModel::ExpUtil
- double **sCap** = 0
- vector< double > **vals** = {}

The documentation for this class was generated from the following files:

- demomtch.h
- demomtch.cpp

3.18 KBase::MtchGene Class Reference

Inheritance diagram for KBase::MtchGene:



Public Member Functions

- void **randomize** (PRNG ∗rng)
- MtchGene ∗ **mutate** (PRNG ∗rng) const
- tuple< MtchGene ∗, MtchGene ∗ > **cross** (const MtchGene ∗g2, PRNG ∗rng) const
- bool **equiv** (const MtchGene ∗g2) const
- void **setState** (vector< Actor ∗ > as, vector< MtchPstn ∗ > ps)

Protected Member Functions

- virtual void **print** (ostream &os) const
- void **copySelf** (MtchGene ∗) const

Protected Attributes

- vector< Actor ∗ > **actrs** = {}
- vector< MtchPstn ∗ > **pstns** = {}

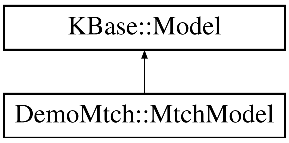
Additional Inherited Members

The documentation for this class was generated from the following files:

- kmodel.h
- kposition.cpp

3.19 DemoMtch::MtchModel Class Reference

Inheritance diagram for DemoMtch::MtchModel:



Public Member Functions

- **MtchModel** (PRNG ∗rng, string d="")

Static Public Member Functions

- static MtchModel ∗ **randomMS** (unsigned int numA, unsigned int numI, VotingRule vr, MtchActor::PropModel pMod, PRNG ∗rng)

Public Attributes

- unsigned int **numItm** = 0
- unsigned int **numCat** = 0

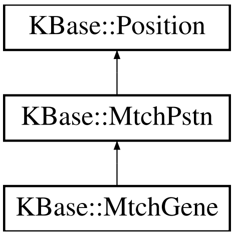
Additional Inherited Members

The documentation for this class was generated from the following files:

- demomtch.h
- demomtch.cpp

3.20 KBase::MtchPstn Class Reference

Inheritance diagram for KBase::MtchPstn:



Public Member Functions

- virtual vector< MtchPstn > **neighbors** (unsigned int nVar) const

Public Attributes

- unsigned int **numItm** = 0
- unsigned int **numCat** = 0
- VUI **match** = {}

Protected Member Functions

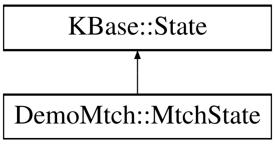
- virtual void **print** (ostream &os) const

The documentation for this class was generated from the following files:

- kmodel.h
- kposition.cpp

3.21 DemoMtch::MtchState Class Reference

Inheritance diagram for DemoMtch::MtchState:



Public Member Functions

- **MtchState** (Model ∗mod)
- KMatrix **actrCaps** () const
- tuple< KMatrix, VUI > **pDist** (int persp) const
- MtchState ∗ **stepSUSN** ()
- MtchState ∗ **stepBCN** ()

Protected Member Functions

- MtchState ∗ **doSUSN** (ReportingLevel rl) const
- MtchState ∗ **doBCN** (ReportingLevel rl) const
- bool **equivNdx** (unsigned int i, unsigned int j) const
- void **setAllAUtil** (ReportingLevel rl)

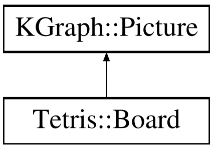
Additional Inherited Members

The documentation for this class was generated from the following files:

- demomtch.h
- demomtch.cpp

3.22 KGraph::Picture Class Reference

Inheritance diagram for KGraph::Picture:



Public Member Functions

- void **add** (Canvas ∗c)
- void **update** () const
- virtual void **connect** (Canvas ∗c)
- virtual void **update** (Canvas ∗c) const

Public Attributes

- double **minX** = 0
- double **maxX** = 1
- double **minW** = 1E-6
- double **minY** = 0
- double **maxY** = 1
- double **minH** = 1E-6

Protected Attributes

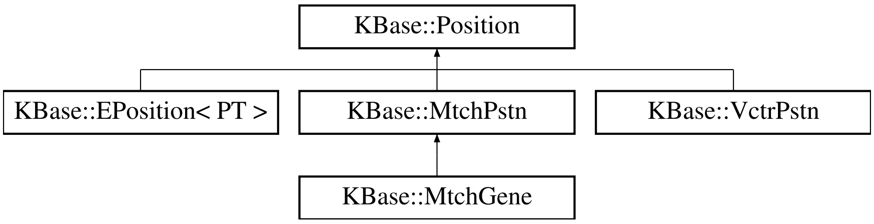
- vector< Canvas ∗ > **canvases** = {}

The documentation for this class was generated from the following files:

- kgraph.h
- kgraph.cpp

3.23 KBase::Position Class Reference

Inheritance diagram for KBase::Position:



Protected Member Functions

- virtual void **print** (ostream &os) const =0

Friends

- ostream & **operator**<< (ostream &os, const Position &p)

The documentation for this class was generated from the following files:

- kmodel.h
- kposition.cpp

3.24 KBase::PRNG Class Reference

Public Member Functions

- uint64_t **uniform** ()
- double **uniform** (double a, double b)
- vector< bool > **bits** (unsigned int nb)
- uint64_t **setSeed** (uint64_t)

Protected Attributes

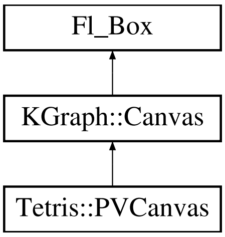
- mt19937_64 **mt** = mt19937_64()

The documentation for this class was generated from the following files:

- prng.h
- prng.cpp

3.25 Tetris::PVCanvas Class Reference

Inheritance diagram for Tetris::PVCanvas:



Public Member Functions

- **PVCanvas** (int x, int y, int w, int h, const char *l=0)
- void **onMove** (int x, int y)
- void **onDrag** (int x, int y)
- void **onPush** (int x, int y, int b)
- void **onRelease** (int x, int y, int b)
- void **onKeyDown** (int x, int y, int k)

Protected Member Functions

- virtual void **draw** ()

Additional Inherited Members

The documentation for this class was generated from the following files:

- pvcanvas.h
- pvcanvas.cpp

3.26 Tetris::Shape Class Reference

Public Member Functions

- **Shape** (TCode p)
- void **setShape** (TCode p)
- void **setRandomShape** ()
- TCode **getShape** () const
- char **getName** () const
- int **x** (int index) const
- int **y** (int index) const
- int **minX** () const
- int **maxX** () const
- int **minY** () const
- int **maxY** () const
- void **showCoords** () const
- Shape **lrot** () const
- Shape **rrot** () const

Public Attributes

- unsigned int **idNum** = 0

Static Public Attributes

- static unsigned int **shapeCounter** = 0

The documentation for this class was generated from the following files:

- shape.h
- shape.cpp

3.27 MDemo::SQLDB Class Reference

Public Member Functions

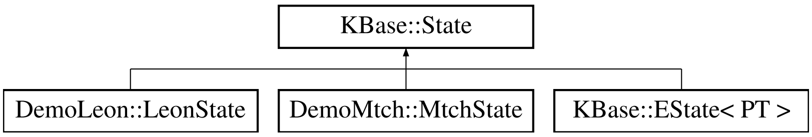
- **SQLDB** (char ∗filename)
- bool **open** (char ∗filename)
- tuple< unsigned int, vector
  < vector< string > > > **query** (const char ∗query)
- void **close** ()

The documentation for this class was generated from the following files:

- sqlitedemo.h
- sqlitedemo.cpp

3.28 KBase::State Class Reference

Inheritance diagram for KBase::State:



Public Member Functions

- **State** (Model ∗mod)
- void **randomizeUtils** (double minU, double maxU, double uNoise)
- void **clear** ()
- virtual void **addPstn** (Position ∗p)
- virtual tuple< KMatrix, VUI > **pDist** (int persp) const =0
- void **setAUtil** (int perspH=-1, ReportingLevel rl=ReportingLevel::Silent)
- void setUENdx ()

Public Attributes

- Model ∗ **model** = nullptr
- function< State ∗()> **step** = nullptr
- vector< Position ∗ > **pstns** = {}
- vector< KMatrix > **aUtil** = {}

Protected Member Functions

- virtual bool **equivNdx** (unsigned int i, unsigned int j) const =0
- virtual void **setAllAUtil** (ReportingLevel rl)=0
- virtual void **setOneAUtil** (unsigned int perspH, ReportingLevel rl)

Protected Attributes

- VUI **uIndices** = {}
- VUI **eIndices** = {}
- KMatrix **uProb** = KMatrix()

3.28.1 Member Function Documentation

3.28.1.1 void KBase::State::setUENdx ( )

Looking only at the positions in this state, return a vector of indices of unique positions.

The documentation for this class was generated from the following files:

- kmodel.h
- kstate.cpp

3.29 Tetris::TApp Class Reference

Public Member Functions

- **TApp** (uint64_t s)
- void **run** ()
- void **newGame** ()
- void **stepGame** ()
- void **pause** ()
- void **resume** (double delay)
- Fl_Color **color** (unsigned int i) const
- void **setRC** (unsigned int r, unsigned int c)
- void **setLevel** (unsigned int lvl)
- void **setRandom** (bool rp)
- double setDt ()
- void **applyControlState** (ControlState cs)
- void **applyColorScheme** (ControlState cs)
- void **processKey** (int x, int y, int k)
- void **quit** ()

Public Attributes

- unsigned int **level** = 3
- double **dt** = 0.1
- PRNG ∗ **rng** = nullptr
- Board ∗ **board** = nullptr
- unsigned int **rows** = 0
- unsigned int **clms** = 0
- bool **randomPlacement** = false
- double **playTime** = 5∗60
- double **maxPlayTime** = 10∗60
- bool **paused** = true

Static Public Attributes

- static TApp ∗ **theApp** = nullptr

Protected Member Functions

- unsigned int **scoreFn** (unsigned int clc)
- void **setColorScheme** ()

Protected Attributes

- unsigned int **lineCount** = 0
- unsigned int **score** = 0
- const unsigned int **defaultLevel** = 3
- const unsigned int **defaultRows** = 24
- const unsigned int **defaultClms** = 12
- const bool **defaultRP** = false
- vector< Fl_Color > **colors** = {}

3.29.1 Member Function Documentation

3.29.1.1 double Tetris::TApp::setDt ( )

set the time between updates, given the current level

The documentation for this class was generated from the following files:

- tmain.h
- tmain.cpp

3.30 UDemo::TargetedBV Class Reference

Public Member Functions

- virtual void **randomize** (PRNG ∗rng)
- virtual TargetedBV ∗ **mutate** (PRNG ∗rng) const
- virtual tuple< TargetedBV ∗, TargetedBV ∗ > **cross** (const TargetedBV ∗g2, PRNG ∗rng) const
- virtual void **show** () const
- virtual bool **equiv** (const TargetedBV ∗g2) const
- double **evaluate** ()
- double **tblEval** (double minD, vector< double > weights, vector< BVec > tbl) const
- unsigned int **hDist** (BVec bv) const

Static Public Member Functions

- static void **setTarget** (BVec trgt)
- static BVec **getTarget** ()
- static void **showBits** (BVec bv)
- static BVec **randomBV** (PRNG ∗rng, unsigned int nb)

Public Attributes

- BVec **bits** = BVec()

Static Public Attributes

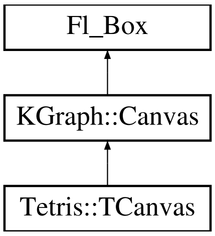
- static BVec **target**

The documentation for this class was generated from the following files:

- kutils/src/demo.h
- kutils/src/demo.cpp

3.31 Tetris::TCanvas Class Reference

Inheritance diagram for Tetris::TCanvas:



Public Member Functions

- **TCanvas** (int x, int y, int w, int h, const char ∗l=0)
- void **onMove** (int x, int y)
- void **onDrag** (int x, int y)
- void **onPush** (int x, int y, int b)
- void **onRelease** (int x, int y, int b)
- void **onKeyDown** (int x, int y, int k)

Protected Member Functions

- virtual void **draw** ()

Additional Inherited Members

The documentation for this class was generated from the following files:

- tcanvas.h
- tcanvas.cpp

3.32 MDemo::TwoDPoint Struct Reference

Public Member Functions

- **TwoDPoint** (unsigned int a, unsigned int b)

Public Attributes

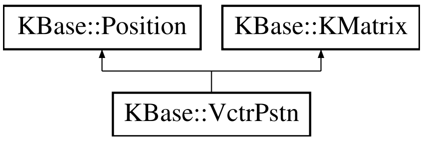
- unsigned int **x** = 0
- unsigned int **y** = 0

The documentation for this struct was generated from the following files:

- edemo.h
- edemo.cpp

3.33 KBase::VctrPstn Class Reference

Inheritance diagram for KBase::VctrPstn:



Public Member Functions

- **VctrPstn** (unsigned int nr, unsigned int nc)
- **VctrPstn** (const KMatrix &m)

Protected Member Functions

- virtual void **print** (ostream &os) const

Additional Inherited Members

The documentation for this class was generated from the following files:

- kmodel.h
- kposition.cpp

3.34 KBase::VHCSearch Class Reference

Public Member Functions

- tuple< double, KMatrix,
  unsigned int, unsigned int > **run** (KMatrix p0, unsigned int iMax, unsigned int sMax, double sTol, double s0, double shrink, double grow, double minStep, ReportingLevel rl)

Static Public Member Functions

- static vector< KMatrix > **vn1** (const KMatrix &m0, double s)
- static vector< KMatrix > **vn2** (const KMatrix &m0, double s)

Public Attributes

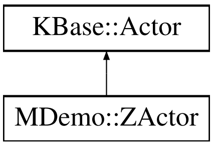
- function< double(const KMatrix &)> **eval** = nullptr
- function< vector< KMatrix >
  const KMatrix &, double)> **nghbrs** = nullptr
- function< void(const KMatrix &)> **report** = nullptr

The documentation for this class was generated from the following files:

- hcsearch.h
- hcsearch.cpp

3.35 MDemo::ZActor Class Reference

Inheritance diagram for MDemo::ZActor:



Public Member Functions

- **ZActor** (string n, string d)
- double **vote** (unsigned int p1, unsigned int p2, const State ∗st) const
- virtual double **vote** (const Position ∗ap1, const Position ∗ap2) const
- double **posUtil** (const Position ∗ap1) const

Additional Inherited Members

The documentation for this class was generated from the following files:

- zactor.h
- zactor.cpp