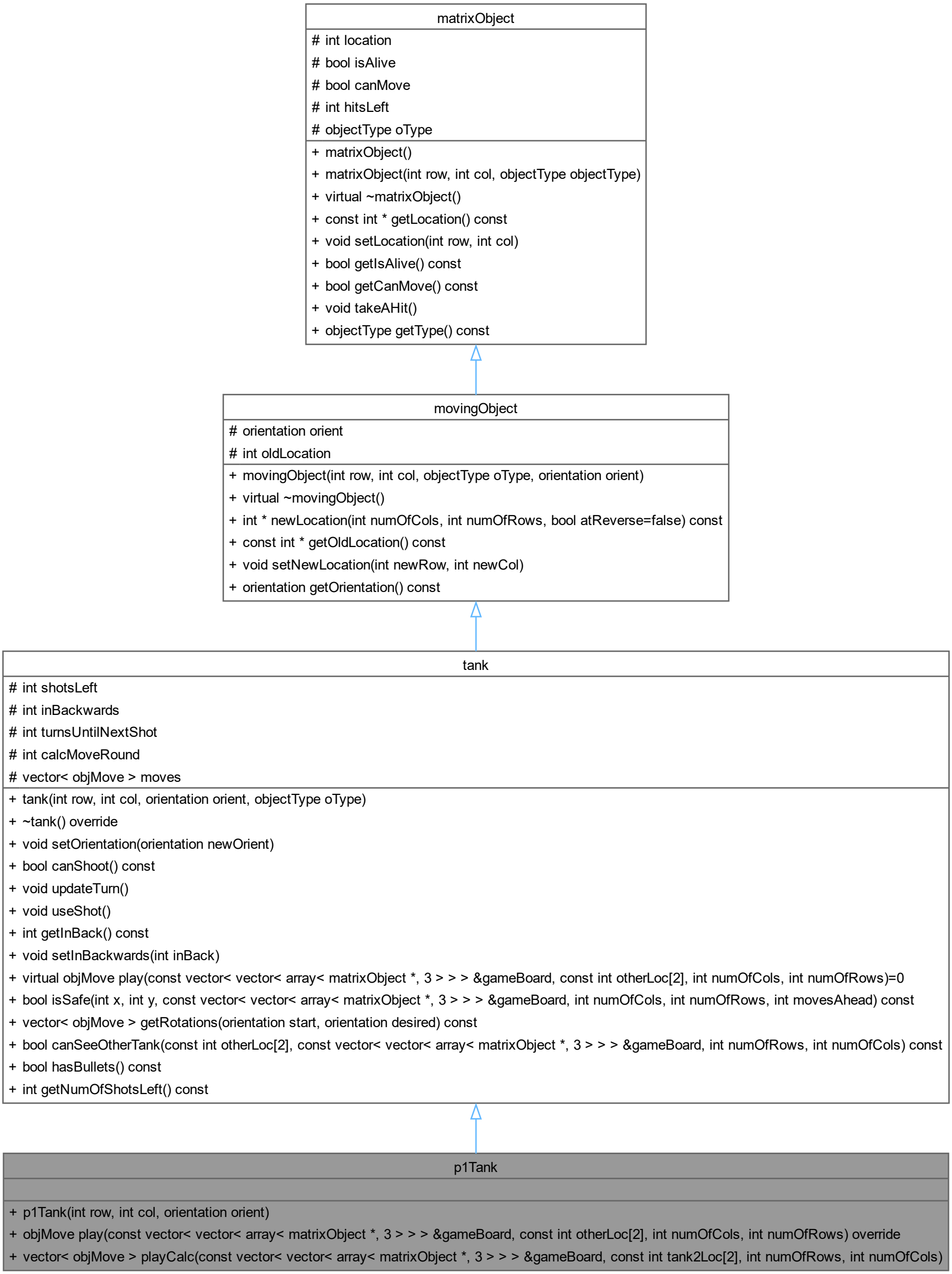
matrixObject
int location
bool isAlive
bool canMove
int hitsLeft
objectType oType
+ matrixObject()
+ matrixObject(int row, int col, objectType objectType)
+ virtual ~matrixObject()
+ const int * getLocation() const
+ void setLocation(int row, int col)
+ bool getIsAlive() const
+ bool getCanMove() const
+ void takeAHit()
+ objectType getType() const
movingObject
orientation orient
int oldLocation
+ movingObject(int row, int col, objectType oType, orientation orient)
+ virtual ~movingObject()
+ int * newLocation(int numOfCols, int numOfRows, bool atReverse=false) const
+ const int * getOldLocation() const
+ void setNewLocation(int newRow, int newCol)
+ orientation getOrientation() const
tank
int shotsLeft
int inBackwards
int turnsUntilNextShot
int calcMoveRound
vector< objMove > moves
+ tank(int row, int col, orientation orient, objectType oType)
+ ~tank() override
+ void setOrientation(orientation newOrient)
+ bool canShoot() const
+ void updateTurn()
+ void useShot()
+ int getInBack() const
+ void setInBackwards(int inBack)
+ virtual objMove play(const vector< vector< array< matrixObject *, 3 > > > &gameBoard, const int otherLoc[2], int numOfCols, int numOfRows)=0
+ bool isSafe(int x, int y, const vector< vector< array< matrixObject *, 3 > > > &gameBoard, int numOfCols, int numOfRows, int movesAhead) const
+ vector< objMove > getRotations(orientation start, orientation desired) const
+ bool canSeeOtherTank(const int otherLoc[2], const vector< vector< array< matrixObject *, 3 > > > &gameBoard, int numOfRows, int numOfCols) const
+ bool hasBullets() const
+ int getNumOfShotsLeft() const
p1Tank
+ p1Tank(int row, int col, orientation orient)
+ objMove play(const vector< vector< array< matrixObject *, 3 > > > &gameBoard, const int otherLoc[2], int numOfCols, int numOfRows) override
+ vector< objMove > playCalc(const vector< vector< array< matrixObject *, 3 > > > &gameBoard, const int tank2Loc[2], int numOfRows, int numOfCols)