class GameTable

{

public:

GameTable(void);

~GameTable(void);

inline unsigned char get\_grid\_connector(int x, int y)

{

// WARNING: no limit checking

return grid\_connectors[x][y];

}

inline unsigned char is\_grid\_clickable(int x, int y)

{

// WARNING: no limit checking

return grid\_clickable[x][y];

}

inline unsigned char get\_grid\_state(int x, int y)

{

// WARNING: no limit checking

return grid\_state[x][y];

}

// public function for initializing the table

void reset\_table(int percent\_missing\_links);

// public function for updating the table connections state ("markers")

int check\_connections();

// public function for returning new elements on a column

void new\_column(int x);

// public function which "clicks" a grid element, rotating it counterclockwise

void click\_element(int x, int y);

// public function which deletes the connected paths, and inserts new elements

int send\_connections();

inline int can\_send()

{

return can\_send\_connections;

}

inline int is\_animating()

{

// returns ANIM\_NONE, ANIM\_DESTROY, or ANIM\_FALL

return still\_animating;

}

// public function which deletes tiles around x, y and inserts new elements

void bomb\_table(int x, int y);

// public function which updates animations

void update\_anims();

// bit 0: right, bit 1: up, bit 2: left, bit 3: down

unsigned char grid\_anim\_frame[GRID\_W][GRID\_H];

unsigned char grid\_anim\_type[GRID\_W][GRID\_H];

private:

unsigned char grid\_connectors[GRID\_W][GRID\_H];

unsigned char grid\_clickable[GRID\_W][GRID\_H];

unsigned char grid\_state[GRID\_W][GRID\_H];

int can\_send\_connections;

int missing\_links;

int still\_animating;

int new\_elements;

int missing\_link\_elements;

unsigned char get\_new\_element();

unsigned char grid\_element\_rotate(unsigned char grid\_code);

void expand\_connections(int cx, int cy, unsigned char ctype, unsigned char marker);

};