BigZaphod / AStar

Branch: master ▾ **AStar / AStar.h** Find file Copy path

BigZaphod Messed with some data types and added goalNode as a parameter to the … 128d35b on 20 Apr 2012

1 contributor

72 lines (54 sloc) 3.65 KB

```
 LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

#ifndef AStar_h
#define AStar_h

#include <stdlib.h>

typedef struct __ASNeighborList *ASNeighborList;
typedef struct __ASPath *ASPath;

typedef struct {
    size_t  nodeSize;                                                                      // the size of the structure
    void    (*nodeNeighbors)(ASNeighborList neighbors, void *node, void *context);             // add nodes to the neighbor
    float   (*pathCostHeuristic)(void *fromNode, void *toNode, void *context);             // estimated cost to transit
    int     (*earlyExit)(size_t visitedCount, void *visitingNode, void *goalNode, void *context);   // early termination, return
    int     (*nodeComparator)(void *node1, void *node2, void *context);                    // must return a sort order
} ASPathNodeSource;

// use in the nodeNeighbors callback to return neighbors
void ASNeighborListAdd(ASNeighborList neighbors, void *node, float edgeCost);

// if goalNode is NULL, it searches the entire graph and returns the cheapest deepest path
// context is optional and is simply passed through to the callback functions
// startNode and nodeSource is required
// as a path is created, the relevant nodes are copied into the path
ASPath ASPathCreate(const ASPathNodeSource *nodeSource, void *context, void *startNode, void *goalNode);

// paths created with ASPathCreate() must be destroyed or else it will leak memory
void ASPathDestroy(ASPath path);

// if you want to make a copy of a path result, this function will do the job
// you must call ASPathDestroy() with the resulting path to clean it up or it will cause a leak
```

```
ASPath ASPathCopy(ASPath path);

// fetches the total cost of the path
float ASPathGetCost(ASPath path);

// fetches the number of nodes in the path
size_t ASPathGetCount(ASPath path);

// returns a pointer to the given node in the path
void *ASPathGetNode(ASPath path, size_t index);

#endif
```