|  |  |
| --- | --- |
| **CPU** | **Intel i7-3770 @ 3.4GHz** |
| **Operating System** | **Windows 7 64-bit** |
| **Memory** | **16GB** |



|  |  |
| --- | --- |
| **CPU** | **Intel i7-3770 @ 3.4GHz** |
| **Operating System** | **Windows 7 64-bit** |
| **Memory** | **16GB** |



# Pattern Databases Pseudocode

I do not have enough time to implement the Pattern Database solution but this is how I would have attempted it.

## Database generation

class PatternNode

{

PatternType

tileValues, 0 for tiles not part of the pattern

blankTilePos

heuristicCost

parent

}

determine patterns

initialise board to goal state

initialise the closed list

initialise the FIFO open queue

add start board to queue

while the queue is not empty

pop a node off the queue, call it node

generate node successors

for each successor

if the successor has already been searched, continue

else

add child to queue

find which patterns the change from node to successor affected

for each pattern affected

create new pattern node with new pattern

set new pattern node’s parent to the board state that last affected this pattern

## Heuristic Function

split board into 3 patterns

for each pattern

extract h cost and add it to total h cost

return total