Valitsin aiheekseni labyrintin lyhimmän reitin ratkaisemisen käyttämällä A Star –algoritmia. Labyrintti toteutetaan kokonaislukutaulukkona, jossa seinää merkitään numerolla 1 ja seinätöntä kohtaa numerolla 0. Labyrintillä on ehkä myös graafinen toteutus, jossa ratkaisun etenemistä voi seurata.

Labyrintissä voi liikkua ylös, alas sekä kulmittain. Kustannus liikkumisesta on pikselien määrä, eli suoraan ruudusta toiseen liikuttaessa 50 ja kulmittain sqrt(50^2+50^2). Labyrintti on siis suuntaamaton verkko. A Star –algoritmi käyttää apunaan jo kuljettujen ruutujen ja seuraavana vuorossa mahdollisesti olevien ruutujen listoja (open ja closed), jotka toteutetaan minimikekoina ja joihin tallennetaan myös ruutujen etäisyyksiä. Algoritmi laskee ensin jokaiselle solmulle arvon, joka on matka solmusta itsestään maaliin. Tämän jälkeen aloitussolmusta lähtien aletaan läpikäydä mahdollista reittiä. Jokainen aloitussolmun viereinen solmu saa myös matkan arvon, joka on matka lähtösolmusta solmuun. Tämän jälkeen lasketaan kaksi edellämainittua matkaa yhteen ja siirrytään pienimmän arvon omaavaan solmuun. Lopuksi lyhin reitti löytyy peruuttamalla maalisolmusta lähtöön asti. Jokainen solmu saa siis algoritmin edistyessä itselleen viittauksen edeltävään solmuun. Algoritmin valittu heuristiikka päättää lopulta, mihin solmuun edetään solmuja läpikäydessä. Seinäksi merkittyyn solmuun ei voi edetä. Algoritmin aika- ja tilavaatimus riippuu heuristiikasta, mutta on parhaimmassa |h(x)-h\*(x)| = O(logh\*(x)), jossa h\* on optimaalinen heuristiikka (http://en.wikipedia.org/wiki/A\*\_search\_algorithm)