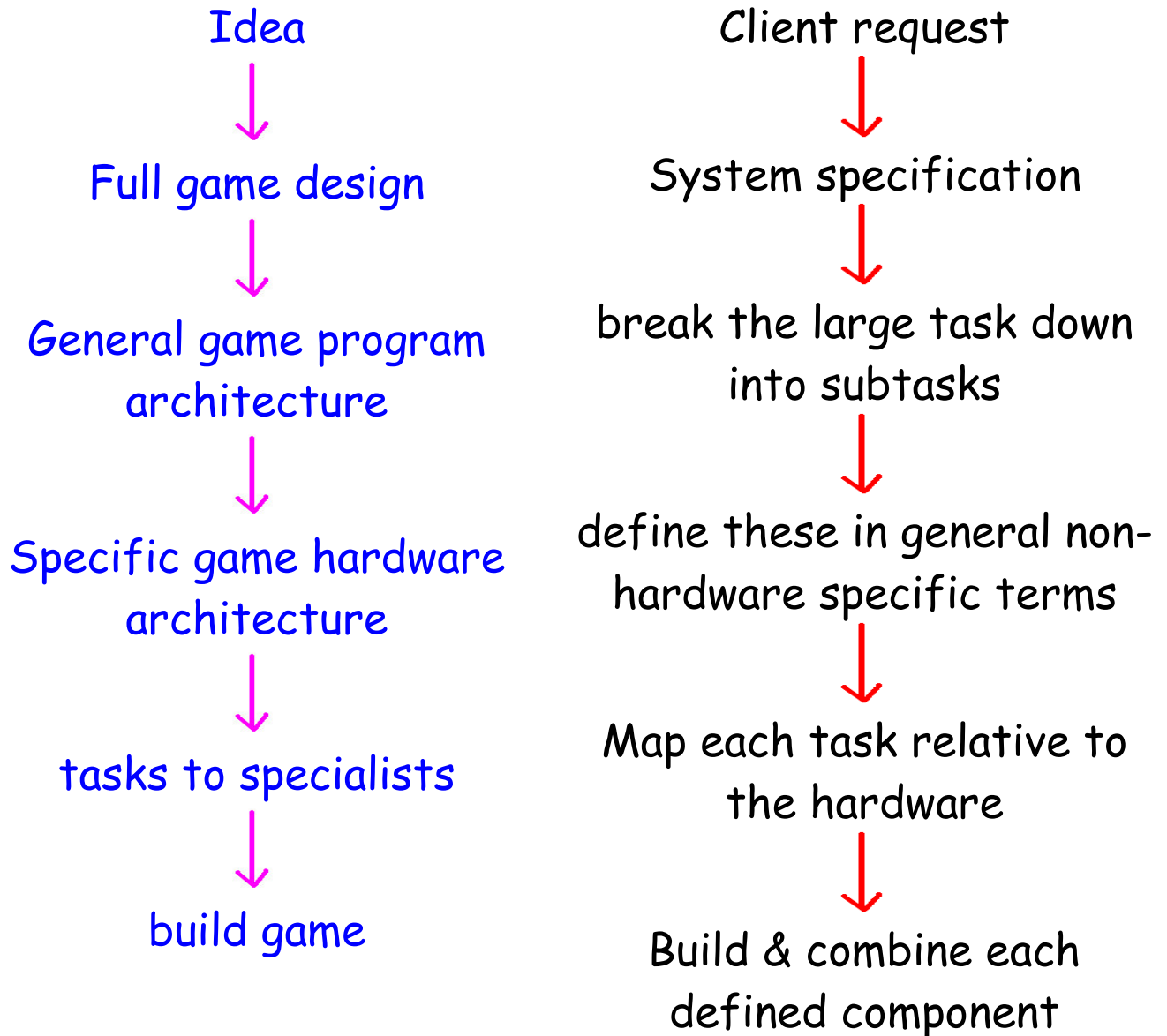


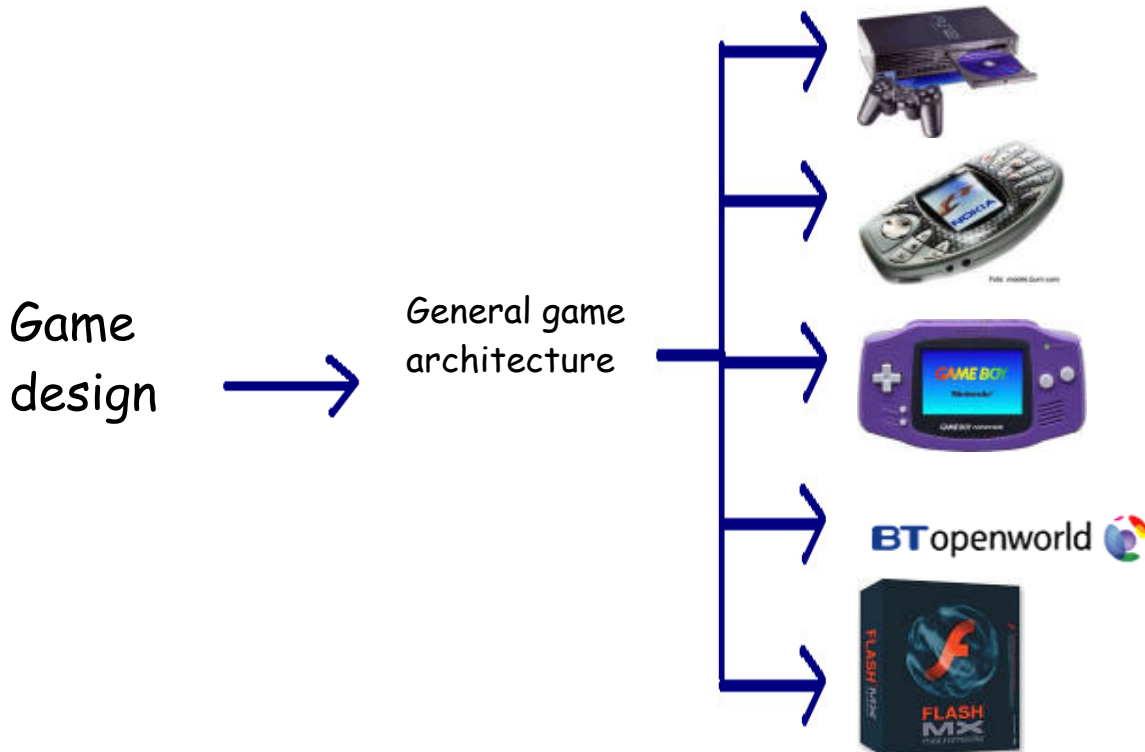
From Game Design to Game Architecture

This returns us to aspects of standard Software Engineering -



By defining a computer program in general, non-hardware specific terms:

- The problem is simplified
- Portability issues are postponed



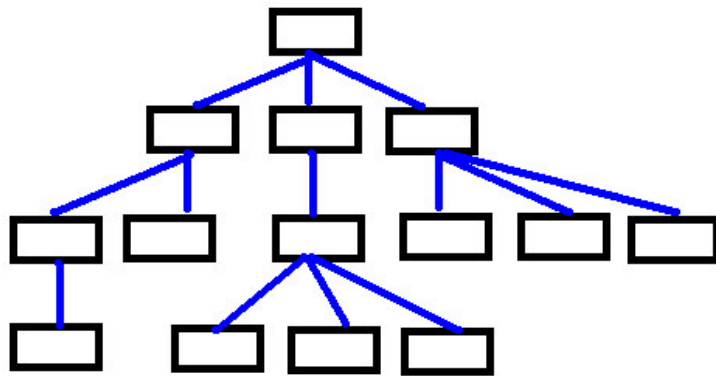
Object Oriented game analysis has already been proposed as one possible technique for defining the general game architecture:

- Identify all the 'things' in the game
- Identify their functionality, communications, events and data

- Remap these to tasks for programmers, artists, musicians and level designers.

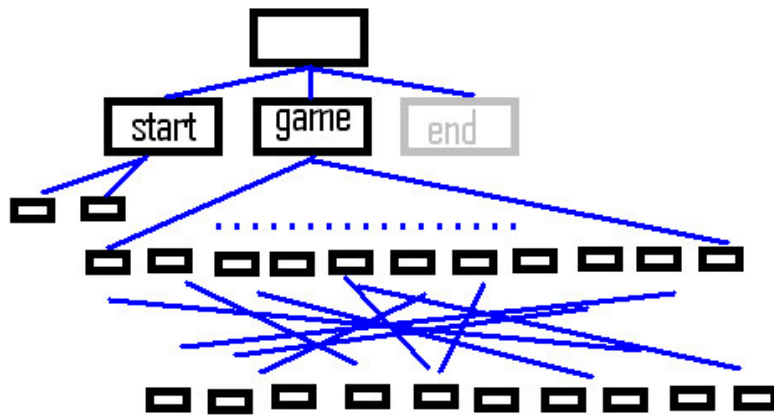
It is quite common for the final architecture to be *flat and wide*, rather than *evenly distributed*:

traditional typical computer program:



low fan-out
 medium coupling
 high cohesion

Typical game computer program:



High fan-out/in
Between levels:
 high coupling
 low cohesion
Within levels:
 low coupling
 high cohesion

A database is also often required:

- To define the level tiling and functionality -

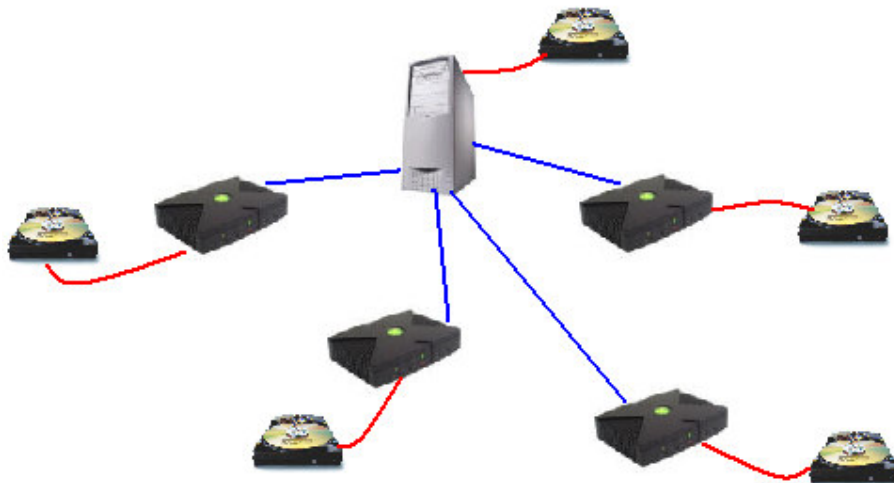


Ward floor, pot plant

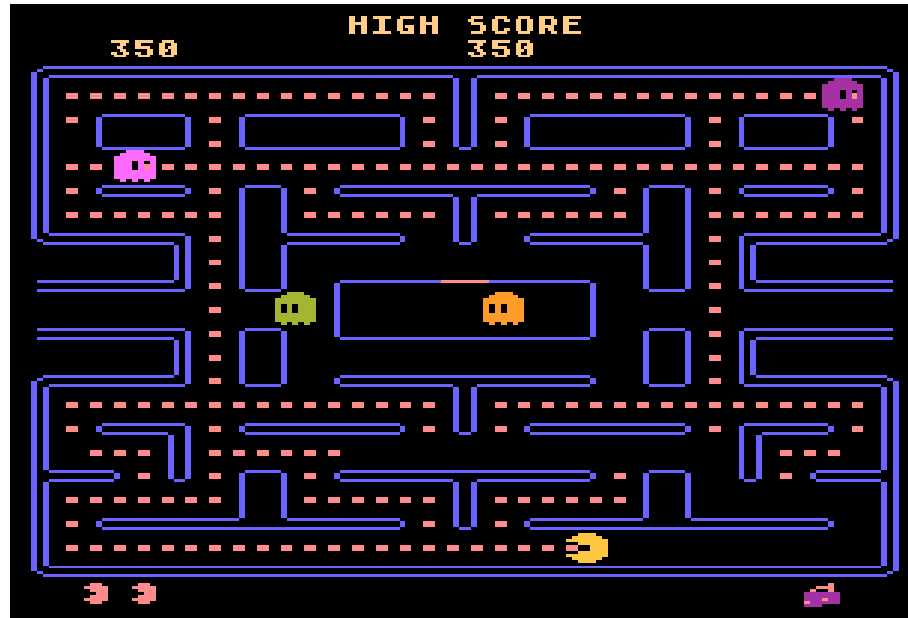
Ward door, closed

Vending machine (iff
corridor tile)

- To hold general game data - lives, scores, event history, player/avatar locations, avatar statuses, ...
- Location and division of persistent and non-persistent data -



e.g. Pacman



component *things* -

images -

sounds -

basic programme architecture -

database -