System Architecture

The system architecture of team 18s program is comprised of 4 unique subsystems that interact with each other to create a fully functional program. Firstly, the program must have a proper GUI, which will be handled by the User Interface subsystem. The GUI will have to interact with the processing system as input from the user must be read into the database for the program. Based on the contents in this database and the current inputs from the user, the display subsystem will handle animations and changing display contents based on the state of the program.

Decomposition

The four main subsystems of the program are User Interface, Processing, Database, and Display. Each subsystem is necessary for this program to effectively meet the design goals of the team.

-User interface

-Processing

-Database

-Display

The User Interface will be the subsystem responsible for handling user input and therefore it will be the only system that user will be directly interacting with. The other systems will be invoked based on the selections from the user interface and therefore it will have strong coupling. The user interface will also have strong cohesion as most of the objects in the user interface are for choosing options and therefor single purpose.

The processing subsystem will be responsible for computing operations based on the user input and program state. It will require input from the user interface and relay output to all other subsystems and therefore has strong coupling. As each computation is only single purpose it will have strong cohesion.

The database subsystem will be responsible for reading and writing, to and from memory. It must access data for the user and write data based on the users input with the user interface. This subsystem has weak coupling as many changes must be made to the database and it is important that this does not impact other stored data. It will also have strong cohesion as most of the data is protected and not easily accessible.

The display system will be responsible for drawing objects on the monitor and ensure animations are properly displayed. This will be affected by the database and processing and will therefore have strong coupling. It also has strong cohesion as each process will display a unique item separate of each other.

Subsystem services

The subsystems that make up the program are each responsible for their own processes necessary for complete functionality. They can work completely independently, but it is also crucial that each one is able to interact with each other and complete tasks that involves processes from each system. However, each system still has its unique role in the entirety of the program.