Design Patterns for Solilo



Nigel Landwehr Johann - 500685221

Bart Dikmans - 500681415

Jan-Willem Jozic - 500623980

Inhoudsopgave

[GoF 3](#_Toc451247598)

[1. Decorator 3](#_Toc451247599)

[2. Singleton 4](#_Toc451247600)

[3. Iterator 5](#_Toc451247601)

[4. State 6](#_Toc451247602)

[5. Object Pool 7](#_Toc451247603)

[6. Private Class Data 8](#_Toc451247604)

[7. Facade 9](#_Toc451247605)

[Non GoF 10](#_Toc451247606)

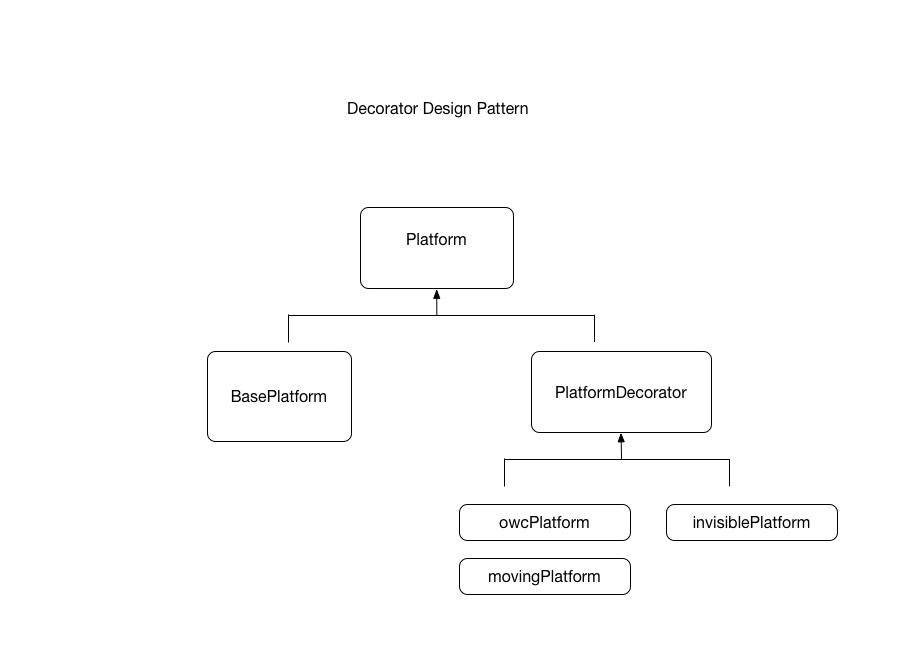
[Sequencing 10](#_Toc451247607)

[Decoupling 11](#_Toc451247608)

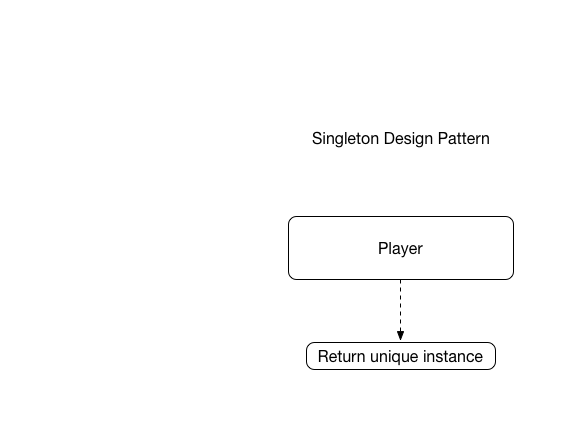
[Cyclic Executive 12](#_Toc451247609)

# GoF

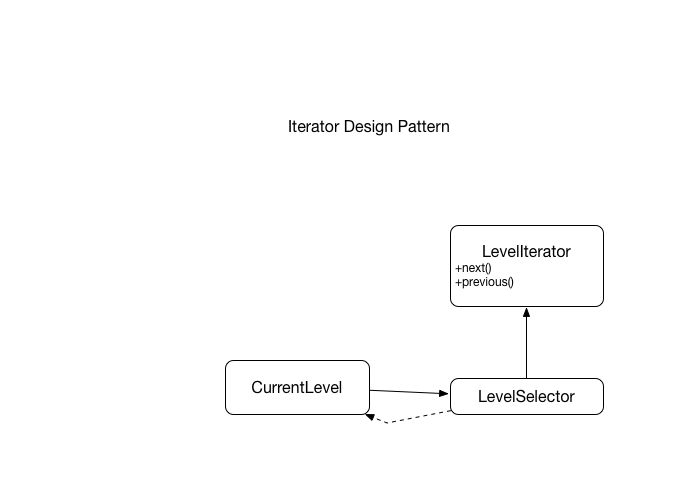
## Decorator



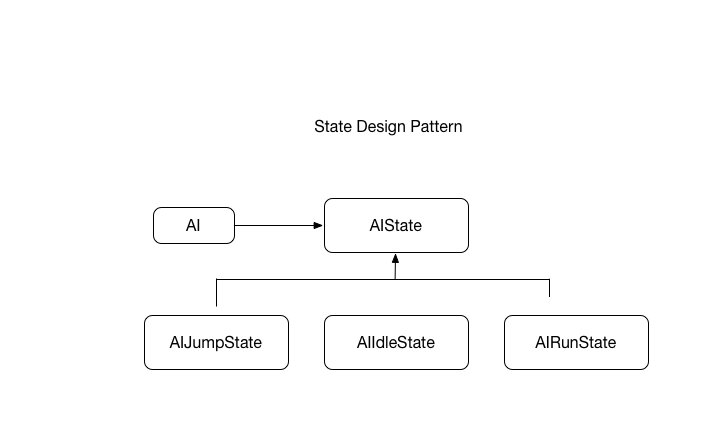
## Singleton



## Iterator

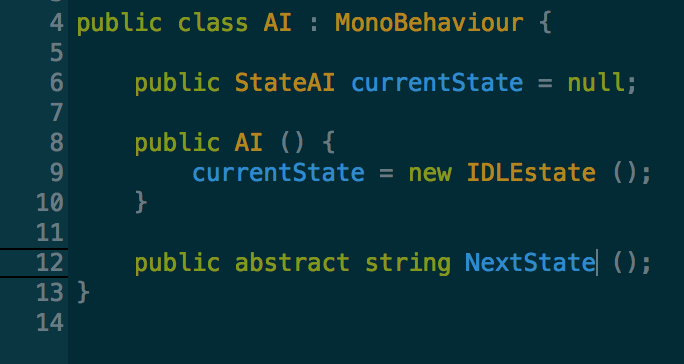


## State

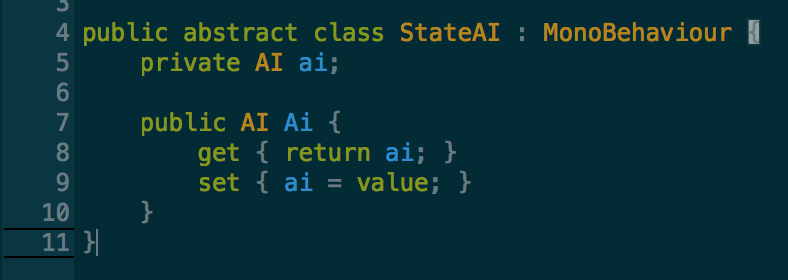


**AI Class**

Contains no state related information, only connected to the object which keeps track of the AI state.

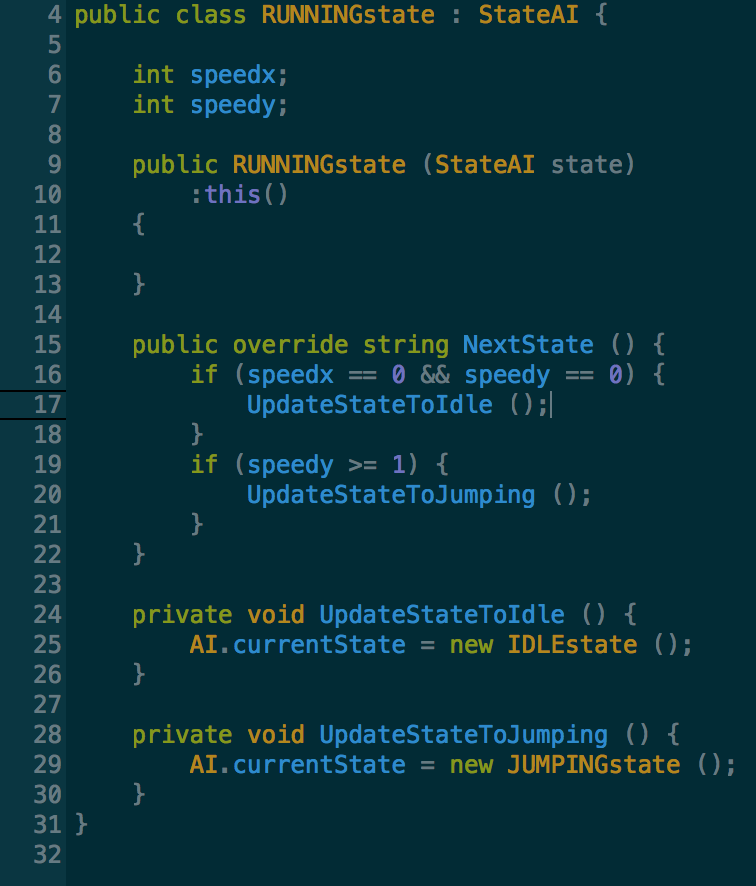


**StateAI Class**

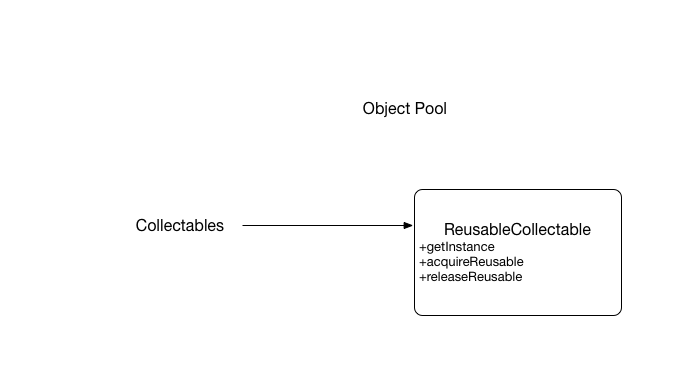
Handles the state

**RUNNINGstate Class**

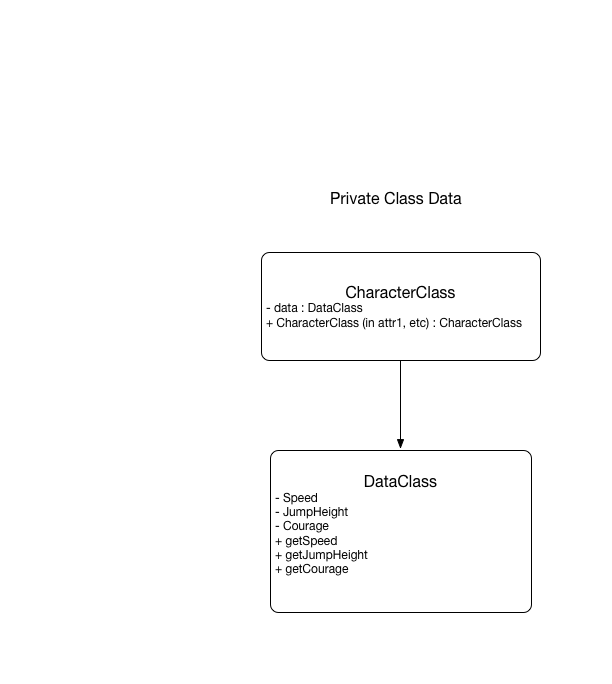
Here’s an state (one of the states the AI could have, such as IDLE, JUMPING, etc)



## Object Pool

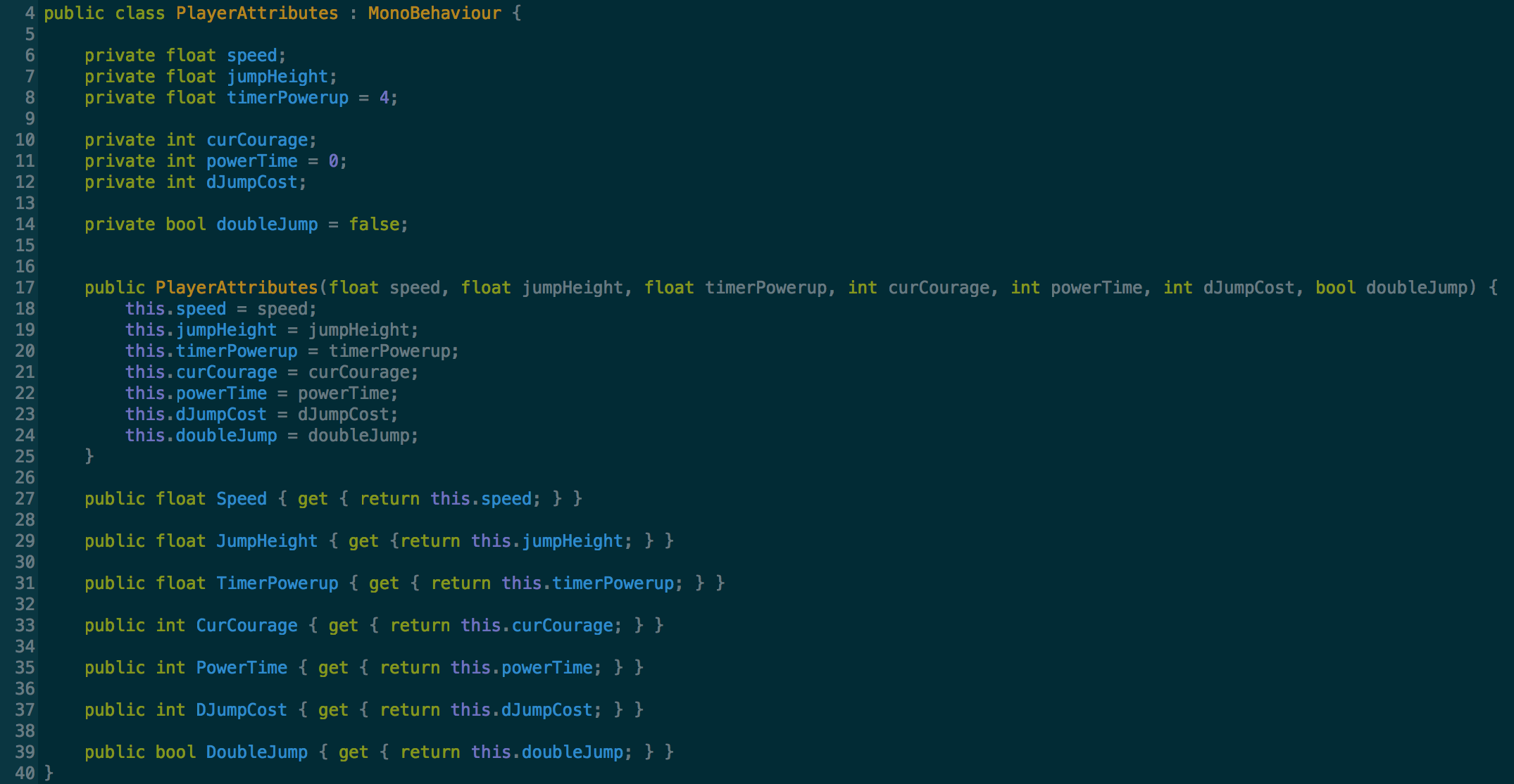


## Private Class Data

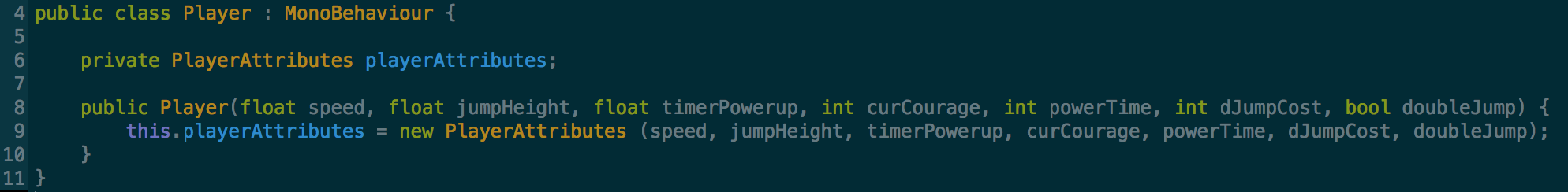


**PlayerAttributes**

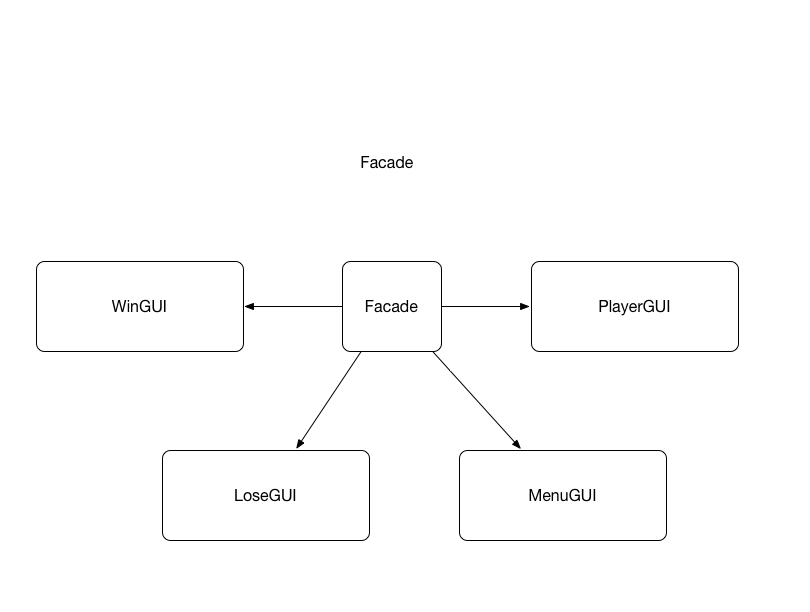
Reducing coupling.



**Player**

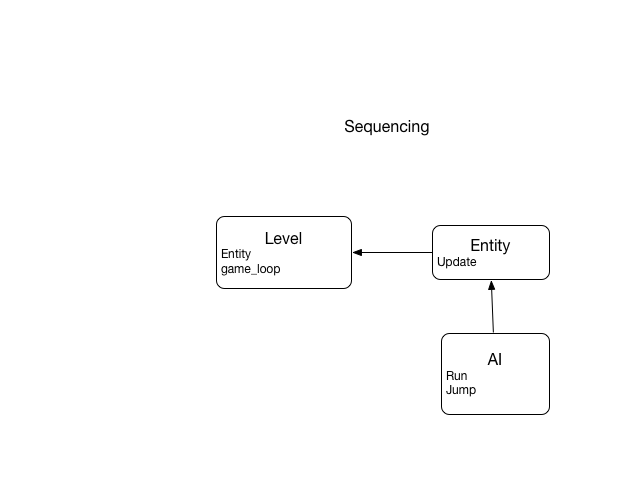


## Facade

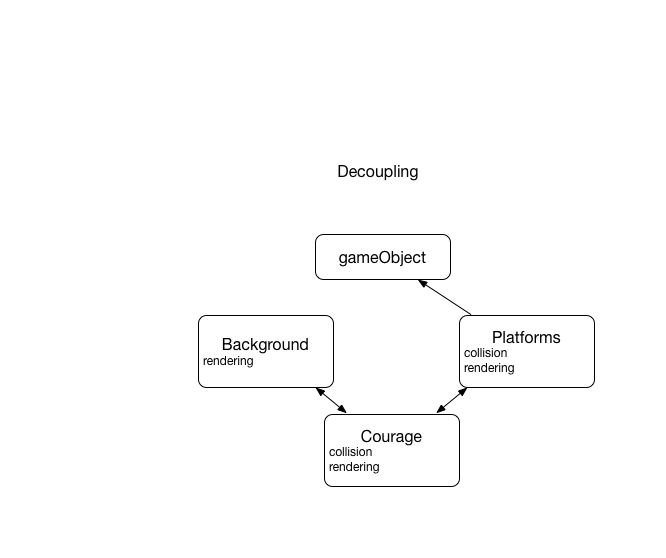


# Non GoF

## Sequencing



## Decoupling



## Cyclic Executive

Intent:

To control the execution of tasks in a defined sequence [B01].

Problem and Solution:

In the Cyclic Executive (CE) pattern, the tasks are executed in a planned sequence and it is therefore simple to implement. In addition, it is highly efficient for monotonous tasking problems. However, CE has a few disadvantages, its inability to react to different events being one. Another big flaw is that if the system gets delayed there is no way of knowing which task will miss its deadline [B01].