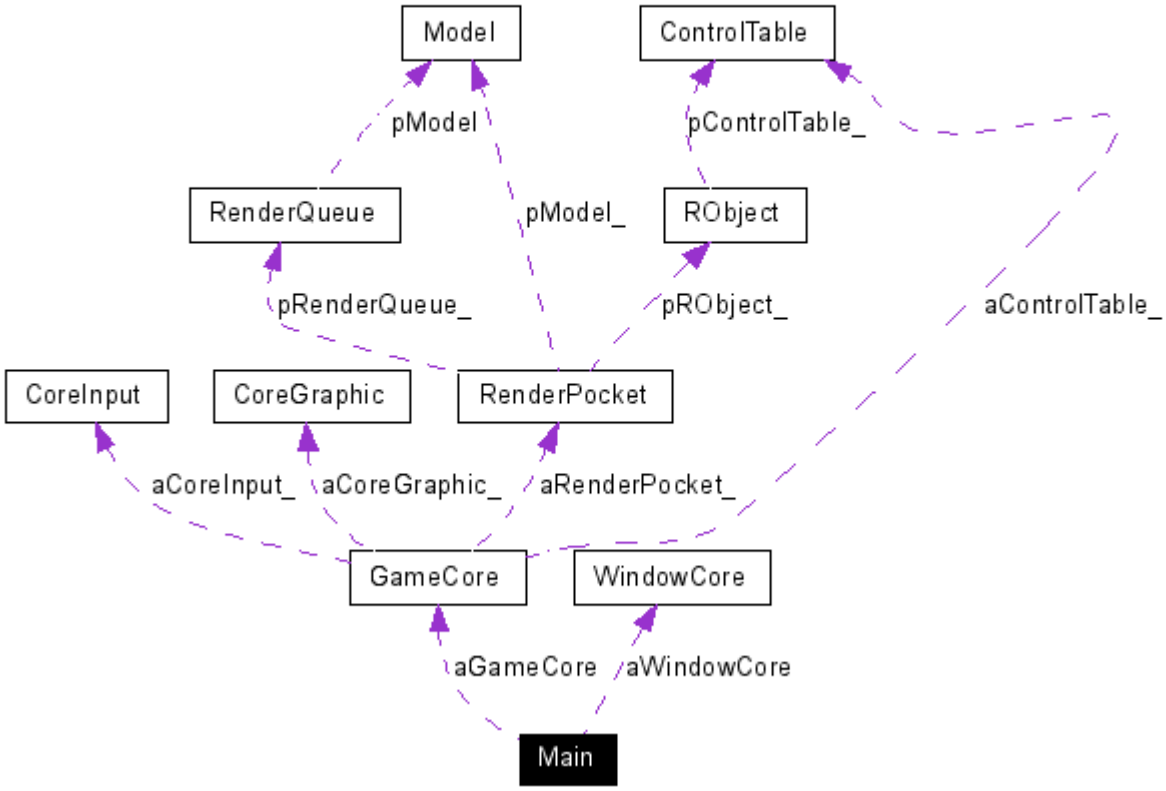
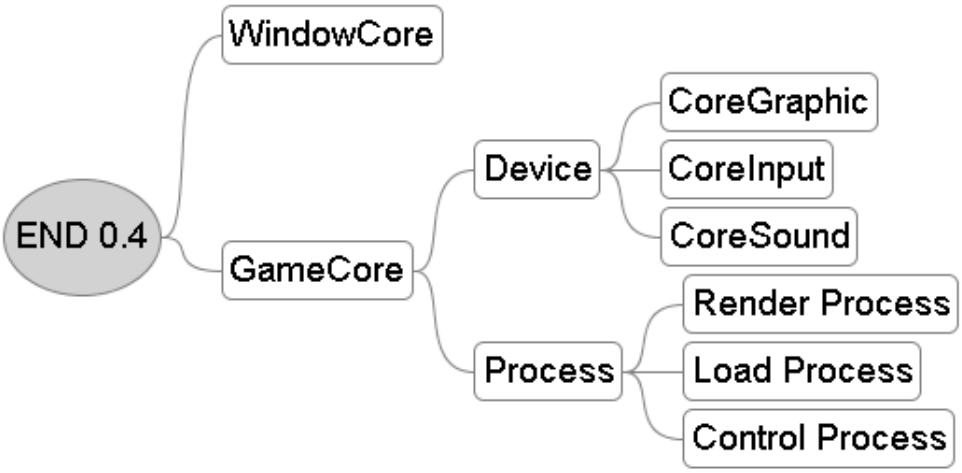
END -Engine for Dimensions

END 0.4 Main Outline



END엔진은 크게 장치 클래스와 수행 클래스로 나눌수 있는데,
장치클래스는 CoreInput, CoreGraphic, CoreSound가 있고
수행클래스는 WindowCore, GameCore가 있습니다.

윈도우와 관련된 수행은 WindowCore클래스로 하며
Load/Rendering/Control등 게임에 중추적인 수행은
GameCor클래스에서 게임 장치클래스를 이용하여 수행합니다.

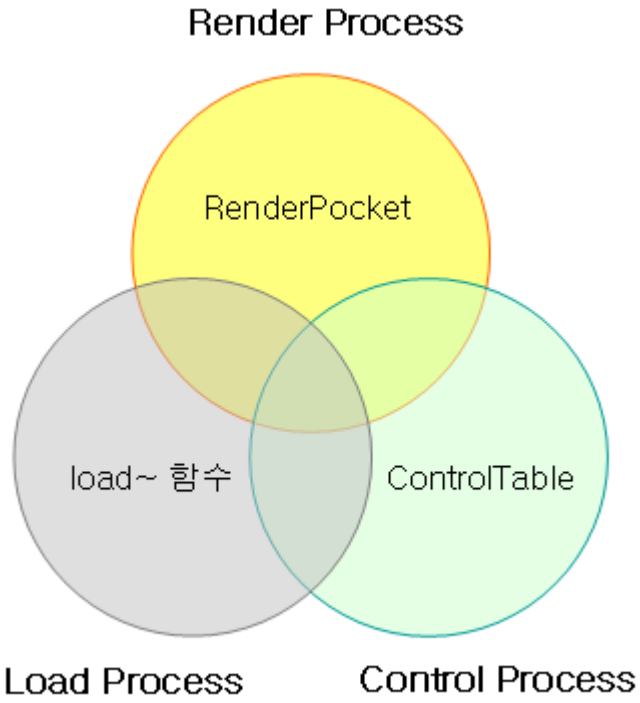


주요 장치클래스로는 CoreGraphic, CoreInput, CoreSound이 있습니다. CoreGraphic은 0.3버전과 많이 달라져서 기본적인 Model Drawing만 가능하게 경량화 시켰고 엔진의 기본객체인 RObjec 렌더링 프로세스는 GameCore에서 다룹니다.

0.2 버전에서는 GameBase에서 장치를 일괄생성했지만 0.4버전에서는 장치마다 독립적으로 생성되며, 엔진의 실질적인 프로세스를 담당하는 GameCore에 대해 독립성을 유지해서 다른 프로젝트에서도 쉽게 사용이 가능합니다.

장치클래스의 사용에 대해서는 홈페이지(http://end.kldp.net)에 튜토리얼이 있는데, 변화가 적은 CoreInput같은 경우는 0.3버전도 유용합니다. (반면 CoreGraphic은 그렇지 않습니다.)

0.2버전과 구조적으로 다른것중 하나가 LayerInfo 클래스가 없어졌다는것이 있습니다. 대신에 각각의 장치는 필요로 하는 객체들을 직접 init계열의 멤버함수를 통해 제공받습니다.



0.2버전의 다른 큰 차이점은 GameState 가 없어졌다는데 있습니다. GameState로 게임의 상태를 바꾸는 대신에 스크립트를 제어함으로써 바꾸는게 더 낫다고 판단해서 GameCore 중앙제어 방식으로 바꾸었습니다.

GameCore는 0.2버전에서의 GameCore와 성격이 많이 다른데, 0.2버전이 State를 제어하기 위한 클래스였다면, 0.4버전에서는 실제적인 수행을 하는 클래스 입니다.

GameCore클래스는 크게 Render Process, Load Process, Control Process 기능의 세부분으로 나눌 수 있습니다.

렌더링의 수행은 모델별로 RObject를 나눈 컨테이너를 포함하는 RenderPocket이 담당합니다. END는 기본적으로 XML에 담긴 스크립트를 중심으로 수행되는 엔진이기 때문에 로딩에는 기본적인 객체로딩뿐만 아니라 스크립트 로딩도 존재합니다.

스크립트가 로딩되면 스크립트로 부터 객체생성이나 자원로딩을 하고 만일 스크립트에 command operation이 포함되어 있다면 command에 맞는 함수를 map에 담긴 함수포인터를 호출하는 방식으로 수행합니다.

GameCore에서는 단지 (부모클래스의)인터페이스를 통해 함수를 호출하면 실질적인 drawing이나 process는 각각의 자식객체가 담당합니다. process와 draw를 하는 객체들은 모두 RObject로 부터 파생되면 RObject는 순수가상함수를 가지는 추상클래스 입니다.

```
//Robject Class
class RObject {
	public:
		RObject() : bShow(FALSE), fX(0), fY(0), fZ(0) {}
		virtual ~Robject(){}
		//순수 가상함수 - 실제 수행을 위한 인터페이스
		virtual void draw(Model* _pModel) = 0;
		virtual void process() = 0;
		virtual void setRObject(CoreGraphic*,CoreInput*,
				OggPlayer*, ControlTable*) = 0;
	public:
		bool  bShow;
		float fX;
		float fY;
		float fZ;

	protected:
		//·렌더링을 위한 mat자료형
		D3DXMATRIXA16 matWorld_;
		//수행결정을 위한 ControlTable
		ControlTable* pControlTable_;
};
```

실제적으로 렌더링이 되는 객체이기 때문에 mat을 protected로 포함하고 있으며, 빈번하게 접근이 되어야 할것들은 함수를 이용하기 보다는 직접적으로 public으로 두었습니다.