Part 2 Doc

Contents

[Coding 1](#_Toc527115247)

[Indentation 1](#_Toc527115248)

[Vertical Alignment 1](#_Toc527115249)

[Tabs 1](#_Toc527115250)

[Directory 2](#_Toc527115251)

[Minimum PC requirements 2](#_Toc527115252)

[Component Architecture 2](#_Toc527115253)

[Description of Main Loop 3](#_Toc527115254)

[Main loop: 3](#_Toc527115255)

[Update: 3](#_Toc527115256)

# Coding

## Indentation

We will use Allman style indentation ex.

**while** (x == y)

{

something();

somethingelse();

}

## Vertical Alignment

We will not do anything special for code alignment

## Tabs

We will use tabs that cascades ex

**class** **MyClass** {

int foobar(

int qux, *// first parameter*

int quux); *// second parameter*

int foobar2(

int qux, *// first parameter*

int quux, *// second parameter*

int quuux); *// third parameter*

};

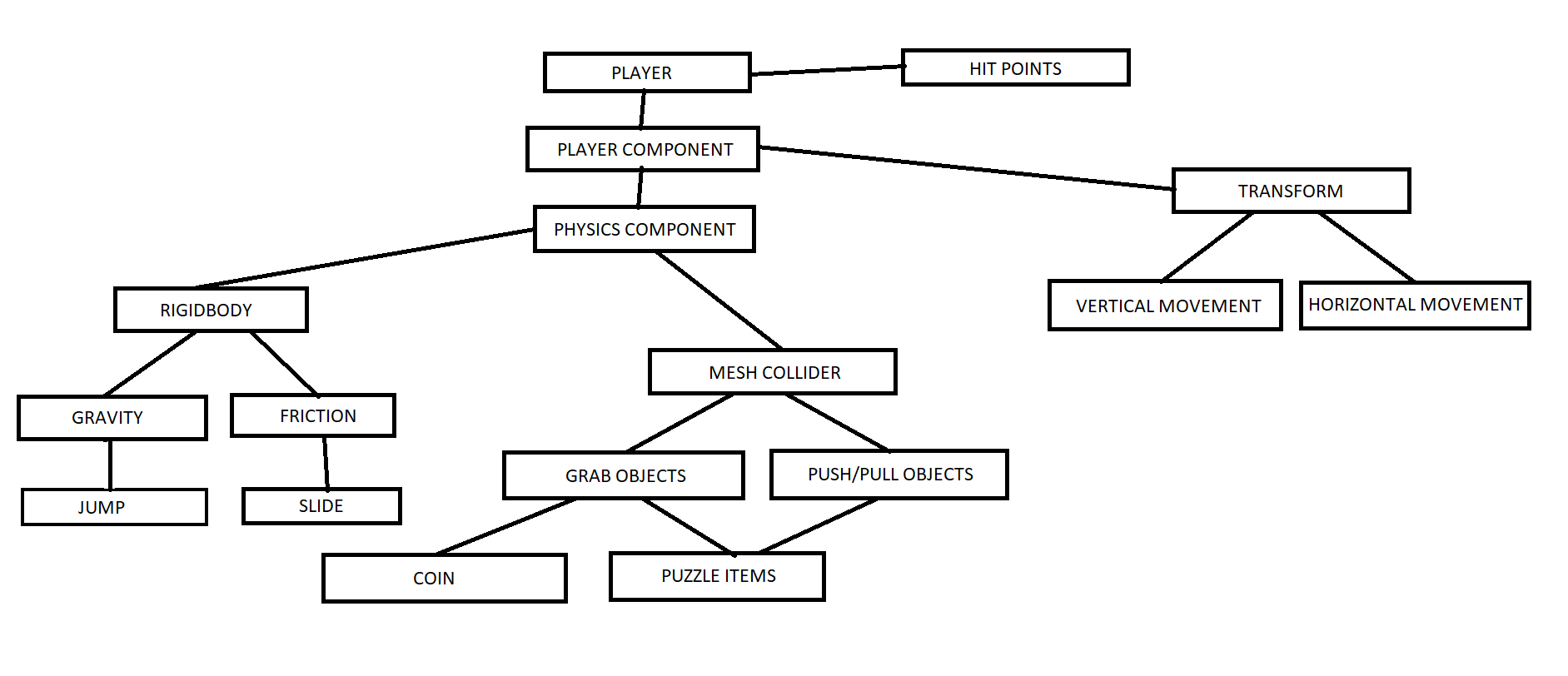
# Directory

We will put work in progress into the Sandbox branch of our git hub, when every thing is finalised it will be moved to the main branch. Also during specific Assignments (like this one) we will have specific branches for each assignment.

# Minimum PC requirements

* **Operating System:** Windows 7 SP1, Windows 8/8.1 (64bit)
* **Processor:** Intel Core2 Quad Q9300 @ 2.5 GHz or AMD Athlon II X4 620 @ 2.6 GHz (or better)
* **RAM:** 4GB.
* **Video card:** NVIDIA GeForce GTX260 or AMD Radeon HD4870 (512MB VRAM with Shader Model 4.0 or higher)
* **DirectX**: **DirectX** June 2010 Redistributable.

# Component Architecture



The Player class owns all of the component classes. The hit point component is derived from the player class, which details the hit points left until the player disappears. The player component derived from the player class, which leads to the physics component, and the transform component. Physics component has the rigidbody aspect to enable physics. The physics that are outlined in the rigidbody class are gravity and friction. Physics component owns the mesh collider so that collision can be detected. Mesh collider can enable you to manipulate puzzle objects/items by grabbing/pushing/pulling them. The main character can grab the end game coin. The player can use the gravity aspect to jump, and use the friction aspect to slide/move the cube around. The transform component features vertical and horizontal movement as a basis.

# Description of Main Loop

## Main loop:

Load audio, video, and text files.

Detect system hardware, and set graphics settings based on them.

Load/check input if controllers present.

Detect audio peripherals and set output.

Allocate memory.

Allocate cache and flow for CPU.

Allocate buffers for CPU, GPU, and Audio.

Create window.

Draw/render graphics on screen.

Launch audio listener.

Launch input listener.

Launch video interface.

Enable necessary sound.

Set graphic settings.

Connect to server if multiplayer game.

Load menu UI and sound.

## Update:

Check player input

Check connection/send packets

Update AI

Update terrain data

Update texture data

Load textures/models as they come by