SuperHotFire Agreement Page

Members: Esaac Ahn

Christopher Farfan Centeno

Terry Humber

Coding Style

Layout

Two consecutive blank lines between functions.

One blank line before and after function definitions.

Indent with tab to indicate level of nesting, and align related subexpressions.

Try to keep your lines between 70 and 80 characters.

Add a space after commas in function calls and semicolons in for loops.

Function curly braces start on the same line.

Commenting

We will only use single line commenting: // not /\* \*/

Single line comments should be added on top, and not in line.

Every function should have a purpose statement.

Naming Convention

Capitalize next words: nextWord

Types start with Capitals: MyClass

Functions and variables start with lower case: myMethod

Constants are all capital: const int PI = 3.14159265358979323;

m\_ prefix used for private members in classes: m\_data;

Project Structure

Docs

All group and project related documents go in here (GDD, TDD, project documentation)

Assets

Images, models, soundtracks etc. go in here

Builds

All builds go in here (Game Builds, data files required for game to run)

ImNotAWrapper

Our Game Engine, all source code will be found here

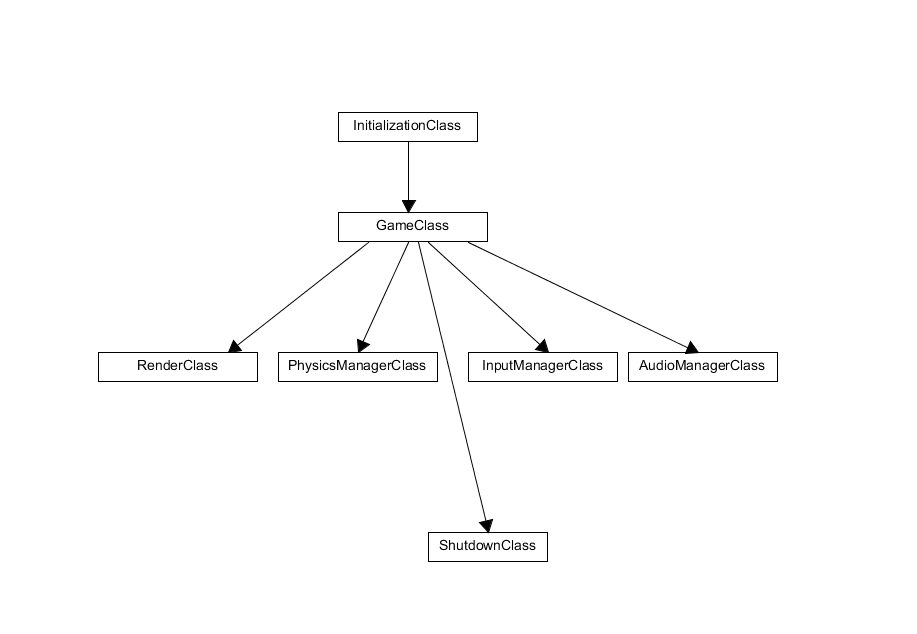
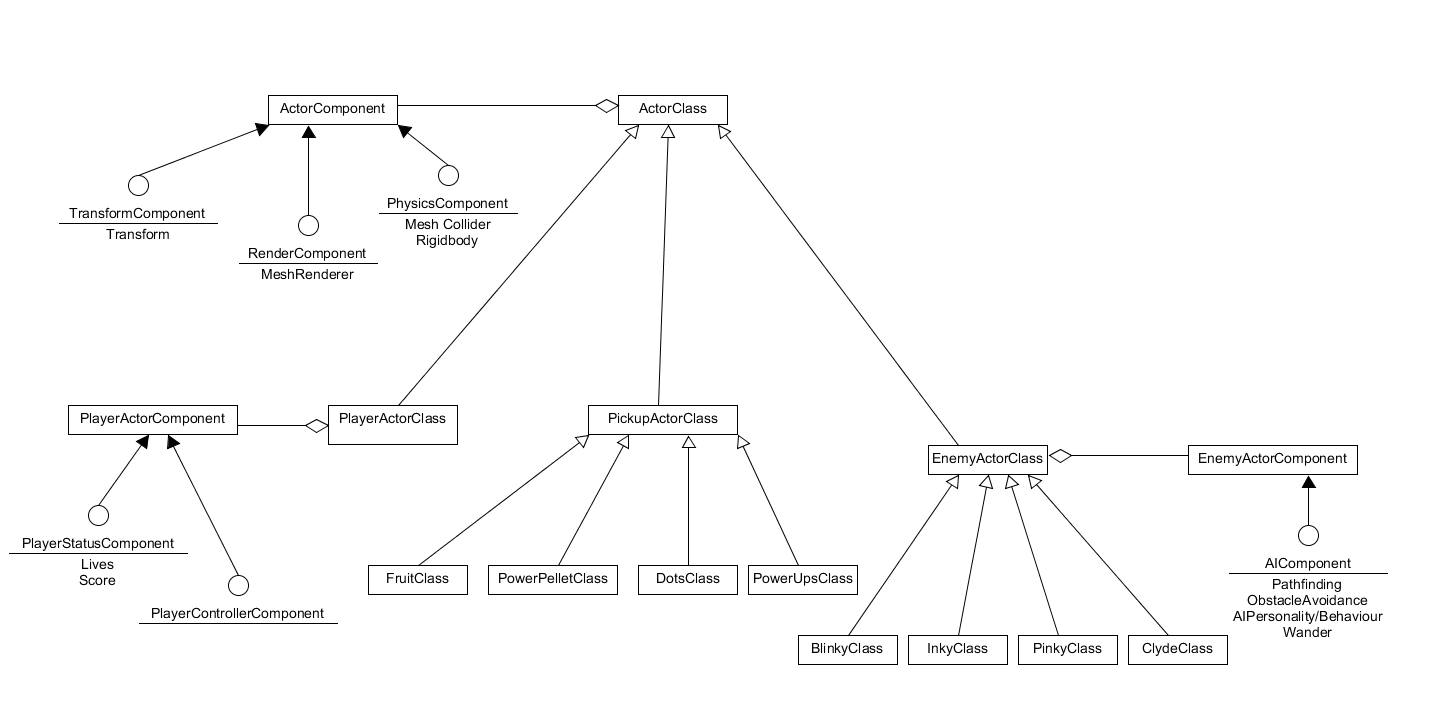
Temp

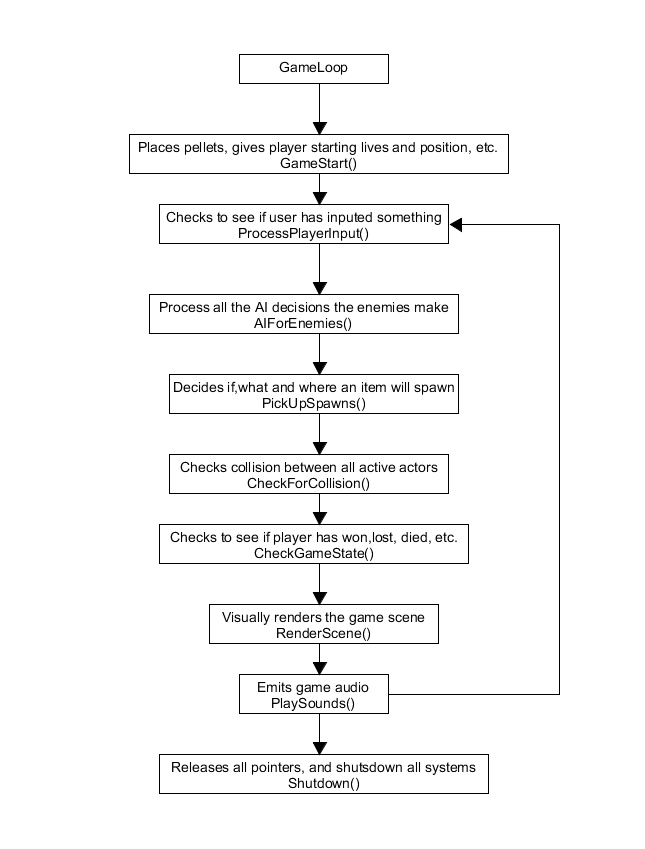
All temp files created by builds are placed here

Minimum PC Requirements

* CPU: Intel i3+ or equivalent
* RAM: 1 GB
* OS: Windows XP
* VIDEO CARD: Nvidia 8800, Radeon HD35xx, Intel HD 4000+
* FREE DISK SPACE: 100 MB

ImNotAWrapper Component, Engine and Loop Architecture





ImNotAWrapper Questions

1 What limitations do you think your game engine would have? Try to explain what these limitations are. (Approx. 50-100 words)

- Physics engine - Graphics engine - Input engine - Audio engine - Logic Engine

At this point answering this question is fairly difficult because we don’t have much of an engine built yet. However that said we are aiming to make a simple game engine that doesn’t use multithreading so are engine would be limited in the kinds of games we could make. Everything will be updated in the same loop so more complicated games that require child processes would not be possible until the engine is updated. We will also be limited in quality and polish due to being confined to completing this engine this semester. We are not expecting to produce a AAA title from this engine this year but we do expect a complete a fully functional engine that can produce a game from start to finish.

2 How can you adapt your game engine in order to accept a different game genre? You may use diagrams, UML, etc. to help illustrate your example. (if your game genre was FPS pick a different genre for instance RPG and describe how your engine would be different). (Approx. 200-300 words)

Our engine will not have many components because it will not be necessary for our game. If we wanted to make changes to our game engine to support an fps game for example, we’d have to make changes to every component in our game engine due to it requiring them. Our physics engine would have to calculate projectile movement as well as player movement. Most FPS games just use raycasting but if we wanted our FPS game to have realistic projectile movement, we’d need our physics engine to calculate things like travel time, ricochets, penetration and deflection angles. The graphics engine would not need to be changed. The audio engine would need 3D sound which is essential to FPS games as it helps the player locate direction using sounds from shots, footsteps, vehicles etc. The engine for ‘The Pac’ is single threaded which will only allow for AI to update at the same time. This would need to be changed for FPS games using AI as they would have need for individual updating, hence requiring multi thread.