DV1435 High-Level Design

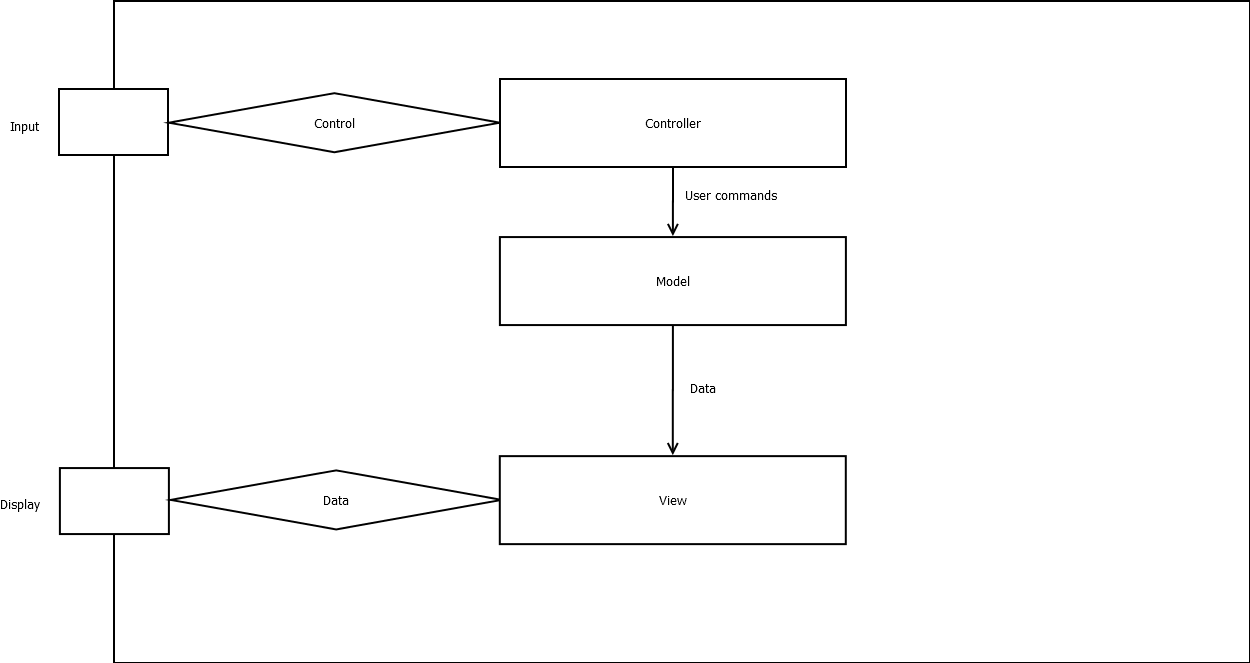
Thomas Sievert, Martin Säll, Lars Woxberg ,Kim Restad &Fredrik Johannesson

# Project description

Pacman::Reloaded is a 3D rendition of the timeless classic Pacman.

The game starts with a title screen, where there are four options: Play the game, view the highscore, view the credits, and quit the game. The game has a set number of stages for Pacman to go through. When he has finished them all, he is sent back to the first stage, on a slightly harder difficulty. Thus, the game goes on indefinitely, or until Pacman dies. When the game is over, the player might be registered to the highscore list. [INSERT: Target and development environment]

# Architecture



The system will contain three distinct packages: Model, View and Controller.

Input is collected from the keyboard and sent to the Controller package. The Controller translates the input to user commands and passes them on to the Model package. The game logic is inside the Model package, and is updated according to the user commands. Further data is sent to the View package, to render Pacman in our 3D environment. This structure is our basic game loop.

# Description of components

[INSERT: Architecture overview]

* Core
  + Content manager  
    The system that handles resources. Derived from a template singleton class. Makes sure that resources are only loaded once, and that they are deleted properly.
* Resources  
  Most resources are wrappers around third party components, to facilitate a more uniform resource management.
  + 3D models  
    Objects in the 3D environment. A wrapper around D3D vertex buffers.
  + Textures  
    Textures for 3D models. A wrapper around D3D shader resources.
  + Materials  
    Ditto.
  + Sounds  
    The sounds to be played in the game. A wrapper around FMOD resources.
  + Music  
    Ditto.
  + Levels  
    The only self-fashioned resource, storing the data necessary to build a level with all its components. Reads from a .png image file and converts it to a resource.
* Low-level renderer  
  Basically an abstraction upon the D3D device, ensuring basic functionality.
  + Materials & shaders
  + Sprite batch  
    A collection of sprites to be drawn.
  + Deferred rendering
  + Camera
  + Lighting
  + Viewports
  + Debug rendering
  + Fonts
* Visual effects  
  The visual effects of the project, making heavy use of the low-level renderer and 3D model, texture and material resources.
  + Particles
  + Reflections
  + Shadows
  + Environment mapping
* Animation  
  Ditto.
  + Morph animation
* Audio  
  The audio component makes use of the sound and music resources.
  + Playback
  + Position based playback
* Front end  
  The component whose responsibility it is to present everything to the screen.
  + HUD
  + GUI
  + Cinematics
  + Scene
* Gameplay foundations  
  The component that makes use of visual effects and the graphic resources.
  + Static world elements
  + Dynamic world elements
* Game-specific subsystems
  + Camera  
    Passing information to the low-level renderer's camera, this component is used to navigate the in-game camera.
    - Chase camera
    - Debug camera
    - Cinematics camera
  + AI  
    This component describes the behavior of the ghosts, and makes use of the level data structure to accomplish that.
    - Path finding
    - Targeting
    - State machine
    - [INSERT: AI state diagram]
  + Logic
    - Collision
    - World grid
    - State machine
    - [INSERT: Pacman state diagram]
  + Movement
    - State

[INSERT: Class diagrams]

# /\* Work breakdown structure

We have detected the following activities, and assigned hours to them respectively:

* L1:
  + Set up templates (10 h)
  + Work breakdown structure (2.5 h)
  + Meetings (15 h)
  + Architecture diagrams (10 h)
  + Weekly reports (5 h)
  + High design report (10 h)
* L2:
  + Class diagrams (30 h)
  + Architecture modeling (50 h)
  + Package diagrams (5 h)
  + State machine diagrams (10 h)
  + Meetings (22.5 h)
  + Weekly reports (3 h)
  + Report (20 h)
  + Prototype (160 h)
* L3:
  + State machine diagram (5 h)
  + Meetings (22.5 h)
  + Weekly reports (3 h)
  + Report (20 h)
  + Prototype (250 h)
* L4:
  + Meetings (7.5 h)
  + Weekly reports (1 h)
  + Report (50 h)
  + Buffer (42 h)

The amount of hours planned above is the sum of every group member’s time spent on the task. \*/