DV1435 Detailed 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. Pacman::Reloaded is developed on Windows for Windows PCs with Direct3D10 compatibility.

# Architecture overview

C:\Temp\thsi\Pacman\Docs\UML\ArchitectureOverviewDiagram.png

# Description of components

# C:\Temp\thsi\Pacman\Docs\UML\ClassOverviewDiagram.png

* Framework  
  *Handles the basic setup of the game, such as creating a window, initializing Direct3D, and provides an abstraction to their services.*
  + Game  
    *Wraps around the game loop. Is meant to be extended by the main game class.*
  + D3DContext  
    *Abstracts Direct3D functionality.*
  + ApplicationWindow  
    *Abstracts Win32 functionality for handling windows.*
  + Timer  
    *A class for measuring time with high precision.*
  + VertexBuffer  
    *An abstraction for the data structure used to send information to the GPU.*
  + Effect  
    *An abstraction to handle effect (.fx) files, i.e. shaders.*
    - Pass  
      *A subcomponent to the Technique class. One Technique may consist of many passes.*
    - Technique  
      *A subcomponent to the Effect class. One Effect may consist of many techniques.*
* Resources  
  *Manages all resources we do not wish to load twice.*
  + ResourceManager  
    *Holds references to resources of a specific type. This is a templated singleton class. Thanks to this, resources are easily available in any part of the project, should they be needed.*
  + Sound2D  
    *Plays a sound.*
  + Sound3D  
    *Plays a sound from a certain position in a 3D scene.*
  + ModelObj  
    *A 3D model loaded from a model file.*
  + Sprite  
    *Manages a 2D HUD image.*
* Helper  
  *Classes with generic use, needed throughout the system.*
  + Animation  
    *A collection of 3D models, using morph animation to interpolate between key frames.*
  + ParticleSystem  
    *A system of particles used to create various effects.*
  + Camera  
    *A camera used to look at a 3D scene.*
* View  
  *Classes that are concerned with outputting data to the user.*
  + PacmanGame  
    *Main game class, runs the actual game loop.*
  + GameEventSubscriber  
    *An interface used in the model package to notify the view of certain game-specific events.*
  + Screen  
    *The base class for the different game screens.*
    - MenuScreen  
      *The main menu, where the game starts.*
    - HighscoreScreen  
      *The highscore screen, meant to present the highest scores of past players. Can also insert a new entry into the list of scores.*
    - CreditsScreen  
      *A screen mentioning the development team.*
    - IngameScreen  
      *The screen where most of the game happens. Draws the scene and updates the model.*
  + GUI  
    *Components that make up the graphical user interface in the screens.*
    - Menu  
      *A list of selectable options.*
    - Table  
      *A table of data.*
  + Scene  
    *The component that draws the actual game. Collects data from the model and presents it graphically.*
  + Ground  
    *Draws the tiles of the level, walls and floors.*
  + Ghost  
    *Draws a ghost, with an animation/model and a particle system.*
  + Pacman  
    *Draws Pacman, with an animation.*
  + ChaseCamera  
    *A camera controller, chases Pacman's position, and can be turned to look either forwards or backwards.*
* Model  
  *Contains the classes concerning the logic of a Pacman game session.*
  + GameplayHandler  
    *Handles one session, moving through all levels until game over.*
  + ModelDataInterface  
    *Exposes data to the view, necessary to render the game.*
  + LevelHandler  
    *Handles the progression through the levels.*
  + Level  
    *Contains the data of one level. Is able to load itself from a .png file.*
  + Player  
    *Handles the movement and logic of Pacman.*
  + Fruit  
    *Handles the fruit bonuses that sometimes appear in the game, with a timer.*
  + Ghost  
    *Handles the logic of the enemies in the game. Contains an AI object, which can be used to change the behavior with the strategy pattern.*
  + AI  
    *Determines the behavior of a ghost. The following AIs will be implemented, one for each ghost:*
    - Blinky  
      *Moves towards Pacman's actual position.*
    - Pinky  
      *Attempts to move in front of Pacman.*
    - Inky  
      *Tries to move to its mirror position in regards to Pacman's path.*
    - Clyde  
      *Moves towards Pacman, but flees when he gets too close.*