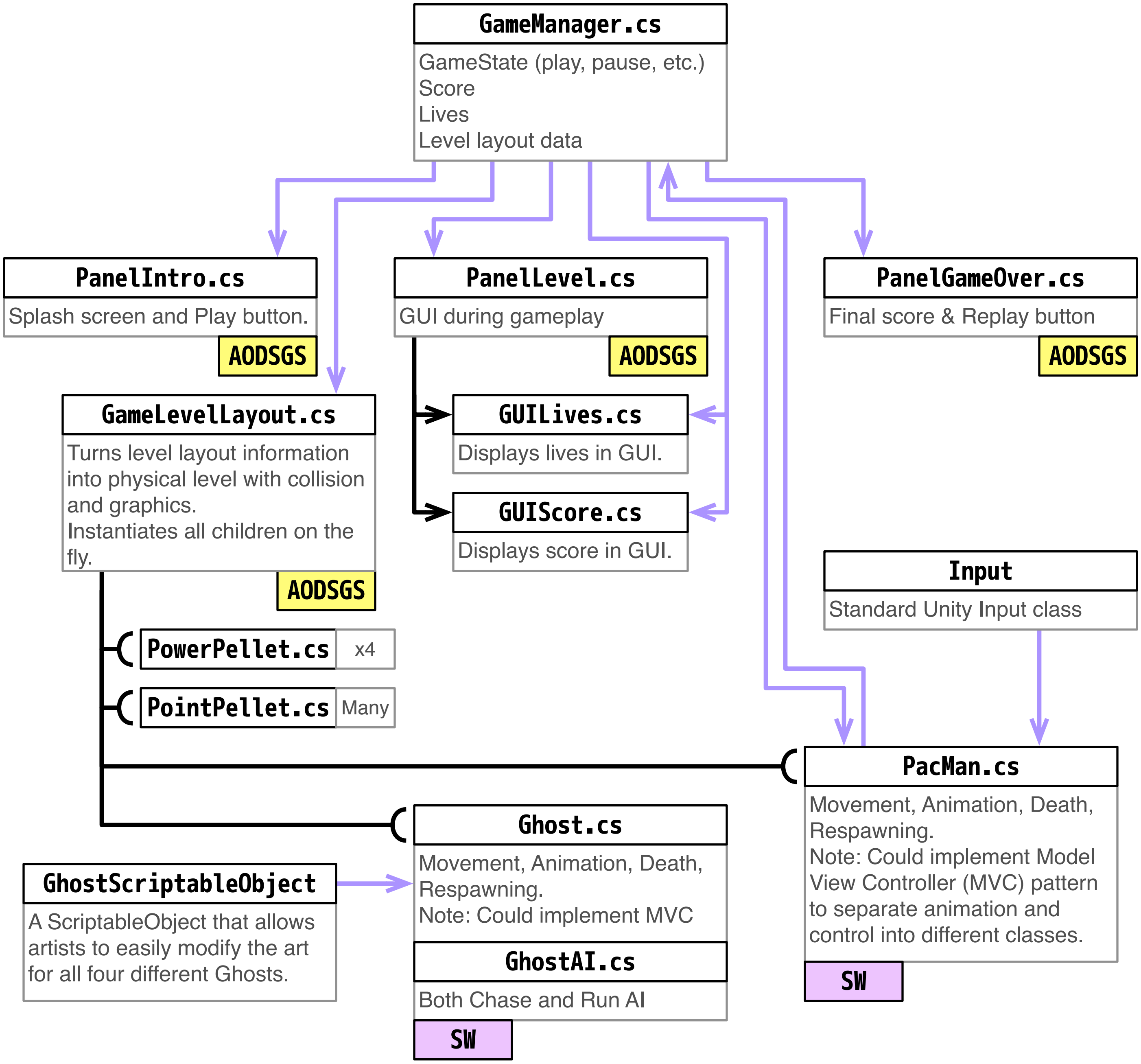
Example: Component & Classes Diagram (CCD)
Pac-Man

Requirements

This CCD covers some of the most basic mechanics from the classic arcade game *Pac-Man*. Many of the elements of the actual game are omitted so that this can fit on a single page. In your *AsteraX* work, make sure that you include every element in the Requirements Doc.

This CCD covers the following game elements/mechanics:

- **Pac-Man** – Player controlled character that moves orthogonally on a 2D plane
  - Pac-Man is constantly moving. When in a hallway, the player can reverse direction. When at a crossroads, the player can choose any of the four directions.
  - If a Ghost touches Pac-Man, it will kill him. When Pac-Man dies, he loses a Life, the game pauses for a couple of seconds, Pac-Man is reset to the center of the level, and the Ghosts are also reset to their starting positions.
  - When Pac-Man consumes a Power Pill, he can chase and eat ghosts for a limited time.
  - Eating a Ghost earns Pac-Man 100 points.
- **Ghosts** – Each of 4 Ghosts has different art.
  - Ghosts attempt to chase Pac-Man but often make random decisions at crossroads.
  - Ghosts will run from Pac-Man when has consumed a Power Pill.
- **Power Pellet** – When consumed, it provides Pac-Man with the limited ability to consume Ghosts.
  - Eating a Power Pill earns Pac-Man 50 points.
- **Point Pellets** – Line all the hallways. Each adds 10 points to Pac-Man's score.
- **GUI** – The Graphical User Interface needs to show:
  - The number of lives Pac-Man has remaining.
  - The score that the player has achieved in this game.
- **Screen Wrap** – Both Pac-Man and the Ghosts will wrap around the screen if they exit out the hallway on either side.



GameManager.cs

GameState (play, pause, etc.)
Score
Lives
Level layout data

PanelIntro.cs

Splash screen and Play button.

AODSGS

PanelLevel.cs

GUI during gameplay

AODSGS

PanelGameOver.cs

Final score & Replay button

AODSGS

GameLevelLayout.cs

Turns level layout information into physical level with collision and graphics.
Instantiates all children on the fly.

AODSGS

GUILives.cs

Displays lives in GUI.

GUIScore.cs

Displays score in GUI.

Input

Standard Unity Input class

PowerPellet.cs x4

PointPellet.cs Many

PacMan.cs

Movement, Animation, Death, Respawning.
Note: Could implement Model View Controller (MVC) pattern to separate animation and control into different classes.

SW

Ghost.cs

Movement, Animation, Death, Respawning.
Note: Could implement MVC

GhostAI.cs

Both Chase and Run AI

SW

GhostScriptableObject

A ScriptableObject that allows artists to easily modify the art for all four different Ghosts.

Notes:

- Solid line denotes a parent/child relationship in Hierarchy pane.
    - Solid line with semi-circle on the end are GameObjects instantiated as needed.
- A light purple arrow denotes a class pulling info from or utilizing another class.
- Boxes stacked together or touching each other are multiple Components on the same GameObject.
- I have left out some extremely simple scripts like those for the buttons on PanelIntro and PanelGameOver.

Common Components:

AODSGS	ActiveOnlyDuringSomeGameStates.cs – Makes the GameObject active/inactive based on GameState.
SW	ScreenWrap.cs – Allows GameObject to wrap around screen horizontally.