Game (super: file)

1. Casting (sub: file)
   1. \_\_pycache\_\_???
   2. object.py
      1. A visible, moveable # that participates in the game. The responsibility of object is to keep track of its appearance, and position in space.
      2. Attributes:
         1. Player (#)
         2. Movement
         3. Velocity
         4. Constructs new objects
         5. Gets Position
         6. Get velocity
         7. Move
         8. Set position
         9. Set color
         10. Sets velocity
   3. point.py
      1. Gems, Rocks, Player, and Score
      2. Attributes:
         1. Rewards points
         2. Take away points.
         3. Score display
   4. Cast.py
      1. A collection of actors. The responsibility of a cast is to keep track of a rocks, gems, and the player.
      2. Attributes:
         1. Actors (dict): {key: group\_name, value: a list of actors}
         2. Construct new actor
         3. Add object
         4. Get object
         5. Remove object
2. Directing (sub: file)
   1. \_\_pycache\_\_???
   2. Director.py
      1. Person who directs the game. The responsibility of a director is to control the sequence of play.
      2. Attributes:
         * 1. Keyboard service: Getting directional input.
           2. Video\_service: for providing video output.
         1. Start Game
            1. Starts game using cast. Runs the main game loop.
            2. Args:

Cast the objects

* + - 1. Get inputs

Gets dirctional input from the keyboard.

* + - 1. Updates

Calculates collisions with objects

1. Services (sub: file)
   1. \_\_pycache\_\_???
   2. Keyboard Services.py
      1. Detects player input.
      2. The responsibility of a keyboard service is to detect player key presses and translate them into a point representing a direction.
         1. Init

Constructs a new keyboardsrevice using the specified cell size

* + - 1. Get direction
  1. Videoservice.py
     1. Outputs in the game state. The responsibility of the classs of objects is to draw the game state on the screen.
        1. \_\_init\_\_
           1. Constructss a new VideeoService using the specified debug mode.
        2. Close window
           1. Closes the window and releases all computing resources.
        3. Clear Buffer
        4. Draw object (player)
        5. Draw objects (gems and rocks)
        6. Flush buffer
           1. Copies the buffer content to the screen. This method should be called at the end of the game’s output phase.
        7. Get cell size
           1. Gets the video screen’s cell size
           2. Return

Grid: the video screen’s cell size

* + - * 1. Get height

Gets the video screen’s height.

Returns: Grid the video screens height

* + - * 1. Get width

Gets the video screens’s sidth.

Returns: grid: the video screen’s width

* + - * 1. Is window open

Whether or not the window is closed by the user.

Returns: bool: True if the window is closing; false if otherwise.

* + - * 1. Open window

Opens a new window with the provided title

Title (string): The title of the window.

* + - * 1. Draw grid

Draws a grid on the screen.

1. Shared (sub file)
   1. \_\_Pycache\_\_
   2. Color.py
      1. Colors of objects
   3. Points.py
      1. Assign different values for colored objects.
2. \_\_Main\_\_.py