Water

+water_start: image, int

+water_tile_width: image, int

+tile_x_amount: image, int

+draw(): image

---

draw()

Pre:
get water png

Post:
draw water png onto screen
within the int limits set

Editor

+land_tiles: image

+switch: image

+get_current_cell(): int

+check_neighbors(): bool

---

get_current_cell()

Pre:
Get origin (0,0)

Post:
return column and row position

check_neighbors()

Pre:
get canvas data

Post:
return whether there is a cell
neighbor present

AnimatedTile

+ size: int

+x: int

+y: int

+path: str

+animate(): image

+update(): int

Level

+enemy_layout: csv

+player_layout: csv

+player_setup(): int

+enemy_collision_reverse(): int

Sky

+top: image, int

+bottom: image, int

+middle: image, int

+draw(): image

---

player_setup()

Pre:
level data (csv)

Post:
loads player start and end
positions if csv files are present

enemy_collision_reverse():

Pre:
level data (csv)

Post:
updates enemy start and end
positions via constraints if csv files
are present

---

draw()

Pre:
get sky png files for top,
bottom, and middle of screen

Post:
draw sky png's onto screen
within the top, bottom, and
middle int limits set

---

animate()

Pre:
get group of png files

Post:
cycle through each png file at
specified speed to get
animated image

update()

Pre:
get animated png files

Post:
update animated png files
onto screen at a specific
location

Camera_Group

+horizontal_pos: int

+player: image

+custom_draw(): image

+draw_horizon(): image

---

customer_draw()

Pre:
get graphics (png)

Post:
loads png files onto game screen

draw_horizon()

Pre:
horizontal position

Post:
takes horizontal position and fills
surface onto game screen

Level:
The level class is its own separate
.py file that has access to all other
.py files. This level class allows us
to create multiple different levels
and controls image blitting and
various update/draw functions for
visuals to be seen on screen

Sky:
Sky class is apart of the
decorations.py file that allows
level.py to access images and
orientation of background to blit
onto the screen

Water:
Water class is apart of the
decorations.py file that allows
level.py to access images and
orientation of background to blit
onto the screen. It also hold
various constraints regarding how
far the water image can span
across the screen

Editor: The editor class is its own
separate .py file that has access
to all other .py files. This editor
class allows us to custom create
different designs within the game
itself- without having to resort to
third party level designing
softwares

Camera_Group:
The camera_group class is in the
level.py file and it contributes to the
camera tracker which follows the
main character as it moves
throughout the game and various
levels

Animated_Tiles:
This class allows for the creation of
animated tiles through the use of
png files. Animated_Tiles accesses
the file path and cycles through
each image at a specified pace.
The image is then drawn onto the
screen and animation can be
seen.

Water

+water_start = 950

+water_tile_width: 100

+tile_x_amount: 1200

Editor

+land_tiles: landtille1.png

+switch: self.land_tiles

AnimatedTile

+ size: 300x500

+x: 300

+y: 200

+path: src/graphics/at.png

Level

+enemy_layout = enemies.csv

+player_layout = player.csv

Sky

+top: sky_top.png, (950,16)

+bottom: sky_bottom.png, (950,16)

+middle: sky_middle.png, (950,16)

Camera_GROUP

+horizontal_pos: = 300

+player = 1.png