Touchmap-0.2 using object orientation

Sandpit/touchmap-0.2/touchmap.py

```
class button:
    def __init__ (self, x, y, size):
        self._x = x
        self._y = y
        self._size = size
        self._tile = touchgui.image_tile (blank_list ("wallv", size),
                                           x, y,
                                           size, size, cellback)
    def to_blank (self):
        self._tile.set_images (blank_list ("wallv", cell_size))
    def to_wall (self):
        self._tile.set_images (wall_list ("v", cell_size))
    def to_spawn (self):
        self._tile = touchgui.text_tile (dark_grey, light_grey, white, mid_grey,
                                         's', self._size,
                                         self._x, self._y,
                                         self._size, self._size,
                                         worldspawn, "worldspawn")
    def get_tile (self):
        return self._tile
```

Touchmap-0.2 using object orientation

- **Sandpit/touchmap-0.2/touchmap.py**

```
blank_t, wall_t, door_t, spawn_t = range (4)  # enumerated types
next_tile = wall_t

function_create = {blank_t:create_blank,
                   wall_t:create_wall,
                   door_t:create_door,
                   spawn_t:create_spawn}
```

- note the `next_tile` is now a variable which contains an enumerated value
  - denoting the cell type to be created on the grid

- notice the dictionary of enumerated values associated with functions

Touchmap-0.2 using object orientation

Sandpit/touchmap-0.2/touchmap.py

```
def cellback (param, tap):
    global clicked, cell_array, button_array, double_tapped_cell
    clicked = True
    mouse = pygame.mouse.get_pos ()
    x, y = get_cell (mouse)
    old = cell_array.get (x + xoffset, y + yoffset)
    button = button_array.get (x + xoffset, y + yoffset)
    if old == " ":
        # blank -> next_tile
        function_create[next_tile] (button)
        cell_array.set_contents (x + xoffset, y + yoffset, "v")
    elif old == "v":
        # wall -> door
        button.to_door ()
        cell_array.set_contents (x + xoffset, y + yoffset, "|")
    elif old == "|":
        # door -> blank
        button.to_blank ()
        cell_array.set_contents (x + xoffset, y + yoffset, " ")
```

Touchmap-0.2 using object orientation

Sandpit/touchmap-0.2/touchmap.py

```
def create_blank (button):
    button.to_blank ()

def create_wall (button):
    button.to_wall ()

def create_door (button):
    button.to_door ()

def create_spawn (button):
    global next_tile
    button.to_spawn ()
    next_tile = wall_t
```

Hints on how to implement the header for the output map

- we need a mechanism to allow the program to remember which assets have been used
- then just prior to writing the file the program needs to iterate over the list and emit the appropriate header info