



## File export and integrating with chisel part 2

### Sandpit/touchmap/touchmap.py

```
def mydoom3 (param, tap):
    pygame.display.update () # flush all graphic changes
    pygame.time.delay (toggle_delay * 2) # pause
    pygame.quit () # shutdown pygame
    dmap () # run chisel and dmap doom3
    exec_doom_map () # now run doom3
    quit () # quit python
```

### Sandpit/touchmap/touchmap.py

```
def dmap ():
    os.system ("d3 +dmap tiny.map +quit")

def exec_doom_map ():
    os.system ("d3 +map tiny.map")
```

## File export and integrating with chisel part 2

- maybe you can improve the code so that it checks whether it needs to export the map
- the whole area of check pointing and saving maps has not been addressed
  - you might want to consider this aspect of the touchmap

## Adding a monster tick

### Sandpit/touchmap/touchmap.py

```
blank_t, wall_t, door_t, spawn_t, hell_t, tick_t = range
```

## Adding a monster tick

### Sandpit/touchmap/touchmap.py

```
class button:
    ...
    def to_blank (self):
        self._tile.set_images (blank_list ("wally", cell_
    def to_wall (self):
        self._tile.set_images (wall_list ("v", cell_size)
    def to_door (self):
        self._tile.set_images (door_list ("v", cell_size)
    def to_hell (self):
        self._tile.set_images (private_list ("hellknight"
    def to_tick (self):
        self._tile.set_images (private_list ("tick"))
    ...
```

## Adding a monster tick

Sandpit/touchmap/touchmap.py

```
def tick (name, tap):
    global next_tile
    pygame.display.update ()
    if tap == 1:
        next_tile = tick_t

def assets ():
    return [touchgui.image_tile (private_list ("hellknight"),
                                touchgui.posX (0.95), touchgui.pos
                                100, 100, hellknight),
            touchgui.image_tile (private_list ("tick"),
                                touchgui.posX (0.95), touchgui.pos
                                100, 100, tick)]
```

## Adding a monster tick

Sandpit/touchmap/touchmap.py

```
def create_tick (button):
    global next_tile
    mouse = pygame.mouse.get_pos ()
    x, y = get_cell (mouse)
    button.to_tick ()
    include_asset ('T', "monster monster_demon_tick")
    cell_array.set_contents (x + xoffset, y + yoffset, "T")
    next_tile = wall_t

function_create = {blank_t:create_blank,
                  wall_t:create_wall,
                  door_t:create_door,
                  spawn_t:create_spawn,
                  hell_t:create_hell,
                  tick_t:create_tick}
```

## Adding a monster tick

Sandpit/touchmap/touchmap.py

```
def cellback (param, tap):
    global clicked, cell_array, button_array, double_tapp
    clicked = True
    ...
    elif old == "|":
        # door -> blank
        button.to_blank ()
        cell_array.set_contents (x + xoffset, y + yoffset)
    elif old in ["H", "S", "T"]:
        # remove asset
        button.to_blank ()
        cell_array.set_contents (x + xoffset, y + yoffset)
        exclude_asset (old)
    ...
```