Ecrire un jeu en 2d avec

Cocos2d

Composants

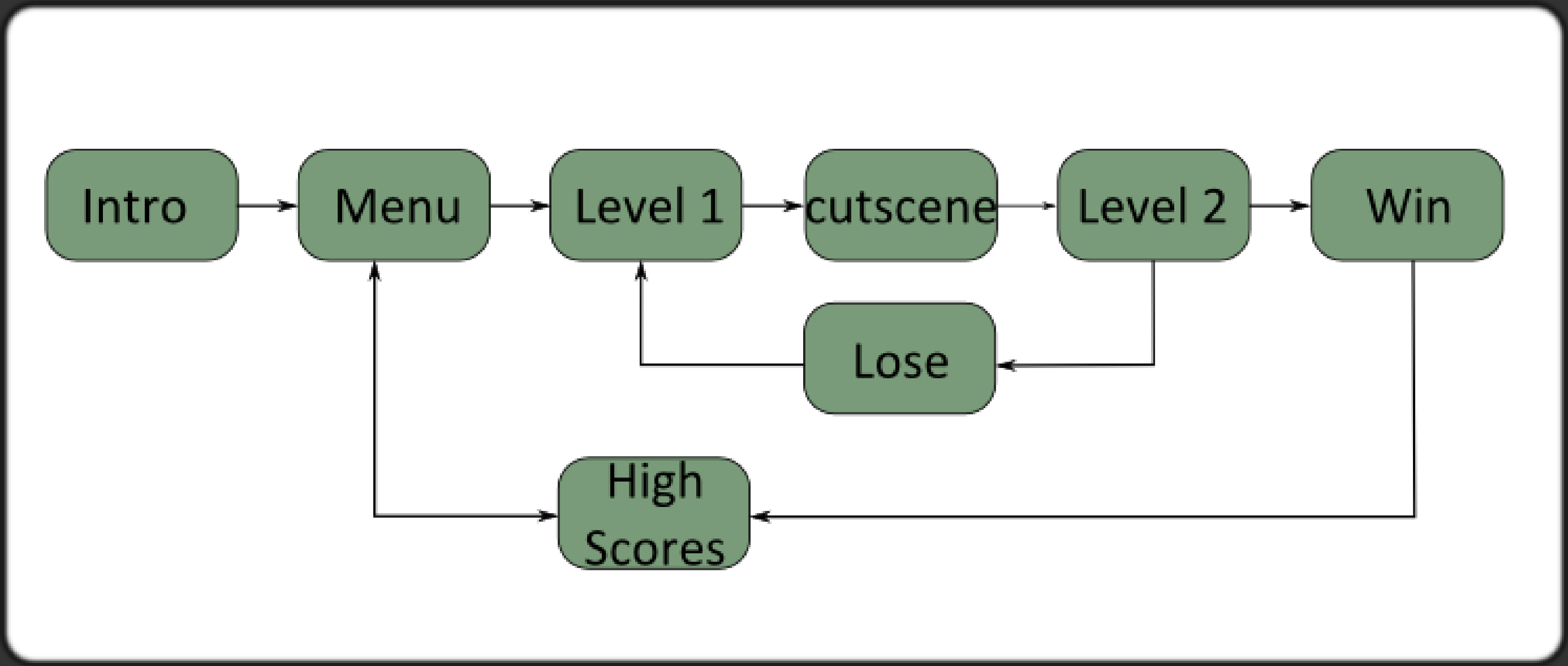
- Director
- Scenes
- Layers
- Sprites
- Events
- Resources

Director

```
window = director.init(
    width=800,
    height=600,
    do_not_scale=True,                              resizable=True)
director.run(somescene)
```

Scenes

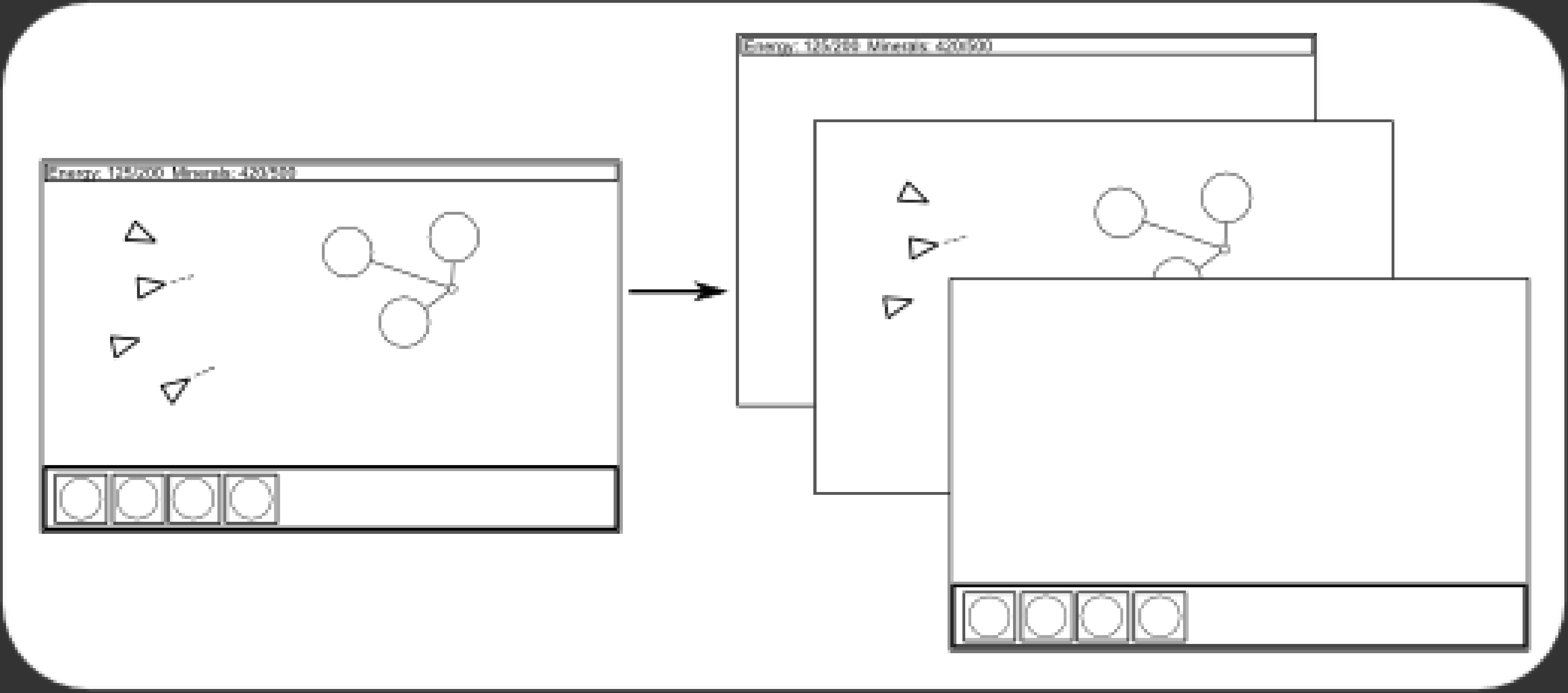
- Organiser votre jeu avec cocos.scene.Scene



- utilisation cocos.scene.Scene(somelayer)

Layers

```
scroller = cocos.layer.ScrollingManager(
    viewport=director.window)
```


Energy: 126/200 Minerals: 420/500
Energy: 126/200 Minerals: 420/500

Sprites

```
class Asteroid(cocos.sprite.Sprite):
    def __init__(self):
        super(Asteroid, self).__init__('asteroid.png')
```

Fourni

- transformations (rotation, move)
- Peut contenir des sprite qui héritent des transformations du parent

Mais Aussi

```
from pyglet.gl import (gluNewQuadric, glColor4f, glPushMatrix,
    glPopMatrix, glTranslatef, glRotatef, gluDisk)


class Circle(cocos.batch.BatchableNode):
[...]
    def draw(self, *args, **kwargs):
        glColor4f(*self.color)
        glPushMatrix()
        glTranslatef(self.x, self.y, -self.z)
        glRotatef(self.rotation, 0, 0, 0.1)
[...]
        gluQuadricDrawStyle(self.quad, self.style)
        gluDisk(self.quad, inner, self.radius, self.resolution, self.loops)
        glPopMatrix()
```

Events

```
class EventEmitter(pyglet.event.EventDispatcher):
    def do_something(self):
        self.dispatch_event(
            'on_purchase_order', **kwargs)

class EventListener(cocos.layer.Layer):
    is_event_handler = True
    def on_purchase_order(self, **kwargs):
        pass
EventEmitter.register_event_type('on_purchase_order')
emitter = EventEmitter()
listener = EventListerner()
emitter.push_handlers(listener)
```

Resources

```
pyglet.resource.path.append('@pyco.data')
pyglet.font.add_directory(
    pkg_resources.resource_filename('pyco', 'data'))
pyglet.resource.reindex()
```

Bonus

- interactive console (CTRL-i)
- toggle full-screen (CTRL-f)

Liens

- cocos2d.org
  - Licence BSD
  - Works on Mac, Linux, Windows
  - Pure Python
- bitbucket.org/faide/pyco
  - Licence MIT
  - Pure Python