4 Time and Interactivity

4.1 Global Time and Timers

4.2 Local Time and Time Containers

4.3 Degrees of Interactivity

Literature:

cocos2d-x.org

R. Engelbert: Cocos2d-x Beginner's Guide, 2nd ed.,
Packt Publishing 2015

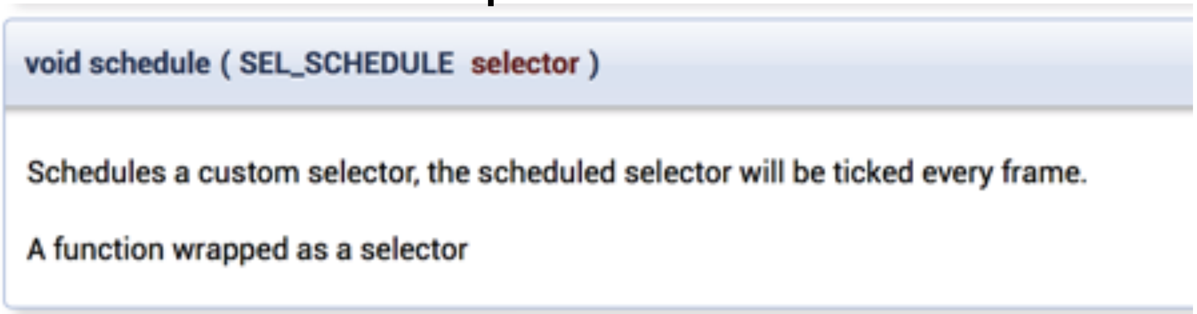
QUIZ!

- Looking at the SlideShow0 example, did we use active or passive waiting between slide transitions?
- Do you have an idea how to implement the other variant?

Trying to Understand Schedulers…

- void SlideShowScene::update (float dt) { … }
  - Parameter `dt` means …?

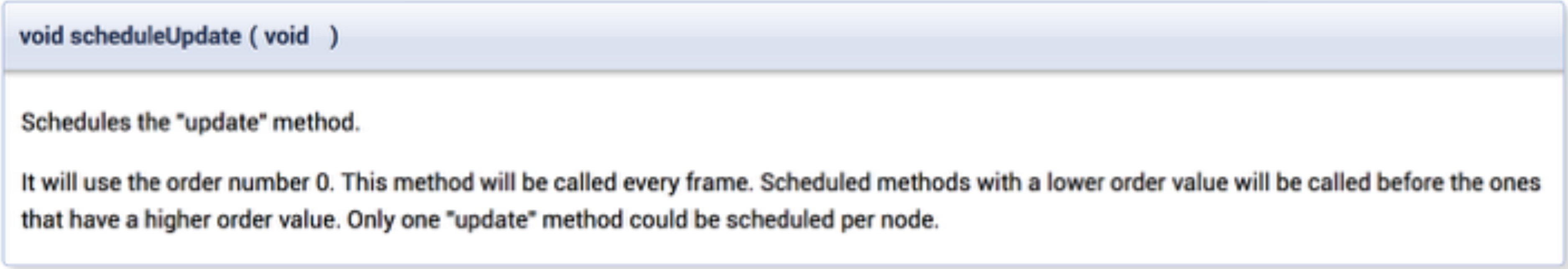
Cocos2d-x API specification:



void schedule (SEL_SCHEDULE selector)

Schedules a custom selector, the scheduled selector will be ticked every frame.

A function wrapped as a selector



void scheduleUpdate (void)

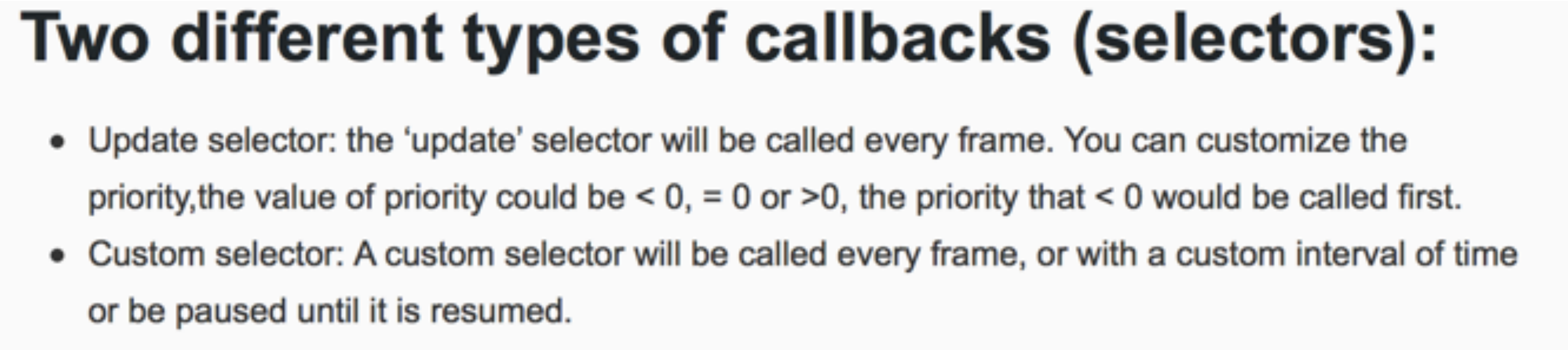
Schedules the "update" method.

It will use the order number 0. This method will be called every frame. Scheduled methods with a lower order value will be called before the ones that have a higher order value. Only one "update" method could be scheduled per node.

What is a selector?

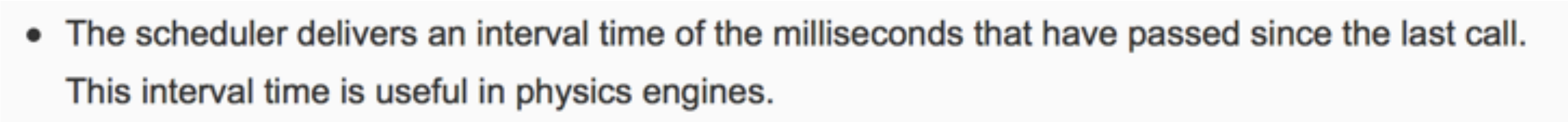
Sparse Documentation…

From: http://www.cocos2d-x.org/wiki/Scheduler



Two different types of callbacks (selectors):

- Update selector: the ‘update’ selector will be called every frame. You can customize the priority,the value of priority could be < 0, = 0 or >0, the priority that < 0 would be called first.
- Custom selector: A custom selector will be called every frame, or with a custom interval of time or be paused until it is resumed.



- The scheduler delivers an interval time of the milliseconds that have passed since the last call. This interval time is useful in physics engines.

Calculating Time Differences

- In `init()`:

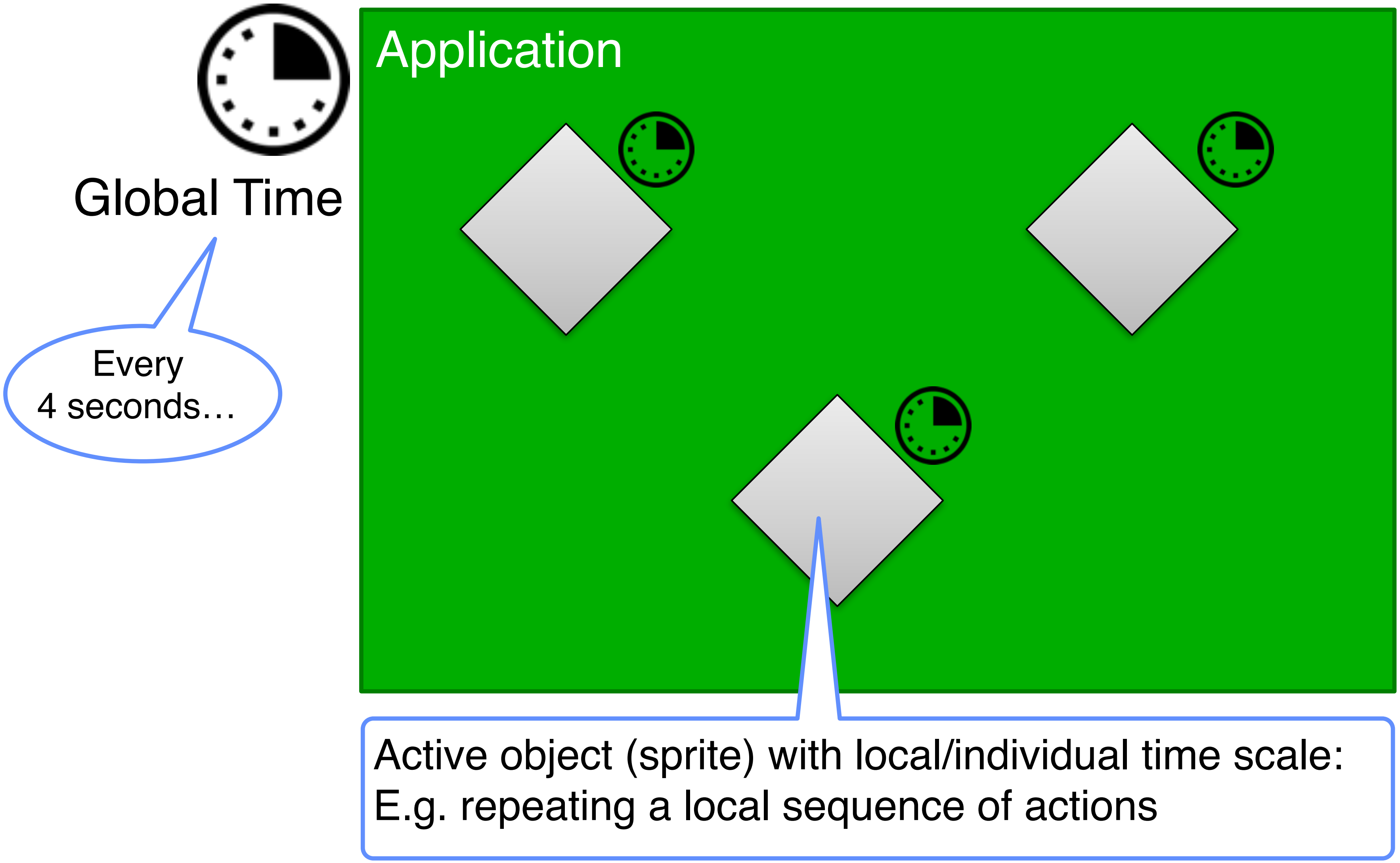
```
// schedule the update function to be called every frame
 this->scheduleUpdate();
```

- In `update(float dt)`:

```
accTime += dt;
if (accTime > INTERVAL) {
     slideIndex = (slideIndex+1) % NUM_PICS;
     imageSprite->setTexture(picFiles[slideIndex]);
     accTime = 0.;
 }
```

- How to declare and initialize `accTime`?

Global vs. Local Time Reference



Every
4 seconds…

Active object (sprite) with local/individual time scale:
E.g. repeating a local sequence of actions

4 Time and Interactivity

4.1 Global Time and Timers

4.2 Local Time and Time Containers

4.3 Degrees of Interactivity

Literature:

cocos2d-x.org

R. Engelbert: Cocos2d-x Beginner's Guide, 2nd ed.,
Packt Publishing 2015

http://www.gamefromscratch.com/post/2014/10/11/Cocos2d-x-Tutorial-Series-Game-loops-Updates-and-Action-Handling.aspx

Local Time in Cocos2d: Actions

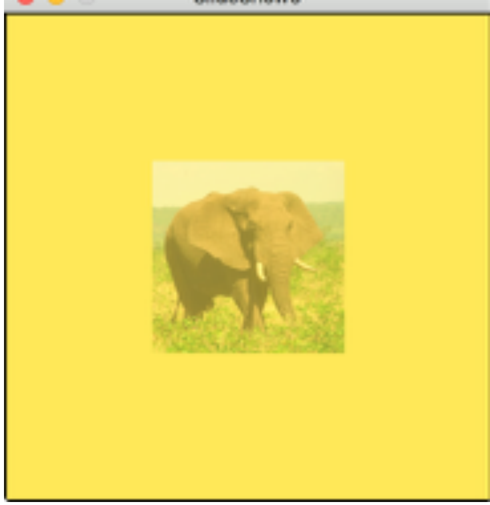
- ***Sprite:*** Graphical object which is dynamically modified (is moved, changes)
- ***Action:*** Activity executed locally for a sprite, once or repeatedly

- Cocos2d-x: `sprite->runAction(action)`
  - Controlled by ActionManager object, e.g. to pause and resume actions
- Building blocks for actions:
  - Pre-defined actions, e.g. fade-in, fade-out
  - Calling a custom function
- Actions are time-constrained
  - Either well-defined execution time
  - Or immediate (no relevant execution time)

Time Containers

- ***Time Container:*** Unit of composition for actions taking place in time
- Composition of time containers/actions:
  - Do *action2* (immediately) after *action1* (sequence)
  - Do *action1* and *action2* in parallel
  - Repeat an action or a sequence of actions (limited or unlimited)
  - Compare e.g. <seq>, <par>, repeatCount in SMIL!
- Cocos2d-x:
  - `Sequence::create(… Actions …)`
  - `Spawn::create(… Actions …)`
  - `RepeatForever(Action)`
- Please note:
  - Actions are executed asynchronously (through separate scheduler)
  - Repeat is only applicable to time-constrained actions

Example: Actions with Local Time



```
auto fi = FadeIn::create(3.0f);
auto fo = FadeOut::create(3.0f);
auto sd = ScaleTo::create(3.0f, 0.1f);
auto su = ScaleTo::create(3.0f, 1.0f);
auto fisu = Spawn::createWithTwoActions(fi, su);
auto fosd = Spawn::createWithTwoActions(fo, sd);
auto inc = CallFunc::create([this](){
    picIndex = (picIndex+1) % NUM_PICS;
    imageSprite->setTexture(picFiles[picIndex]);
});
imageSprite->runAction(
     RepeatForever::create(Sequence::create(fisu, fosd, inc, NULL)));
```

Questions:

- What does this actually achieve?
- What is the syntactical construct used in inc?

4 Time and Interactivity

4.1 Global Time and Timers

4.2 Local Time and Time Containers

4.3 Degrees of Interactivity

Literature:

cocos2d-x.org

Aleem, T. A. (1998). A Taxonomy of Multimedia Interactivity. (Doctoral Dissertation, The Union Institute, 1998). *Digital Dissertations, AAT 9919729.*

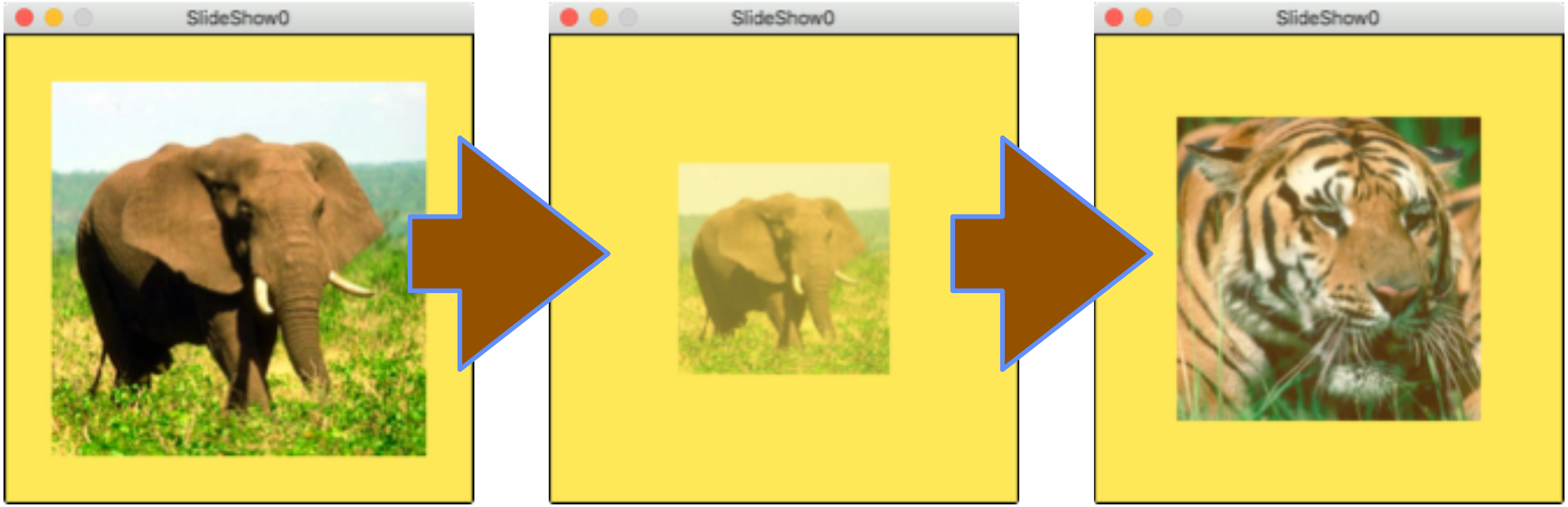
Heller, R. S., & Martin, C. D. (1999). Multimedia Taxonomy for Design and Evaluation. In B. Furht (Ed.), Handbook of Multimedia Computing (pp. 3-16). Boca Raton, Florida: CRC Press.

Interactivity

- Degrees of interactivity (based on T.A. Aleem 1998):
  - Passive, Reactive, Proactive, Directive
- Application to multimedia (Heller et al. 2001) - Examples:

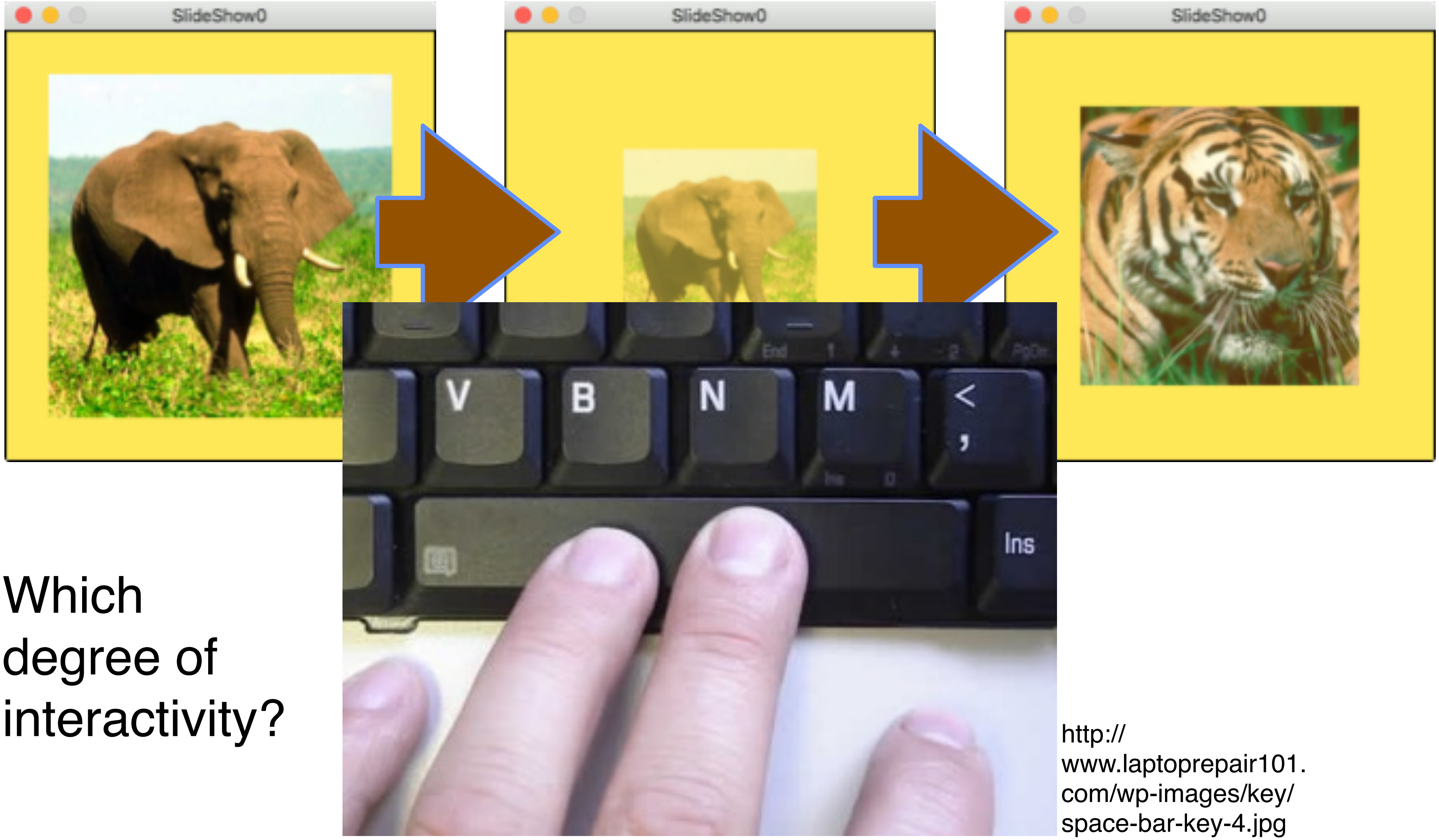
Media type ↓	***Passive***	***Reactive***	***Proactive***	***Directive***
Text	Sequential presentation	Page turner, Linear spacing	Browsing, Hypertext	Word processing
Graphics	Sequential presentation	Predefined changes (choice between graphics)	Change of colors, sizes, shapes, …	Drawing graphics
Sound	Sequential presentation	Predefined changes (sound clip, volume)	Selection of track, fast forward, loop	Creation of sounds
Motion	Sequential presentation	Predefined changes (path, target of motion)	Start, stop, pause, forwd, reverse	Creation of animations

Example 1: Automatic Slide Show



Which degree of interactivity?

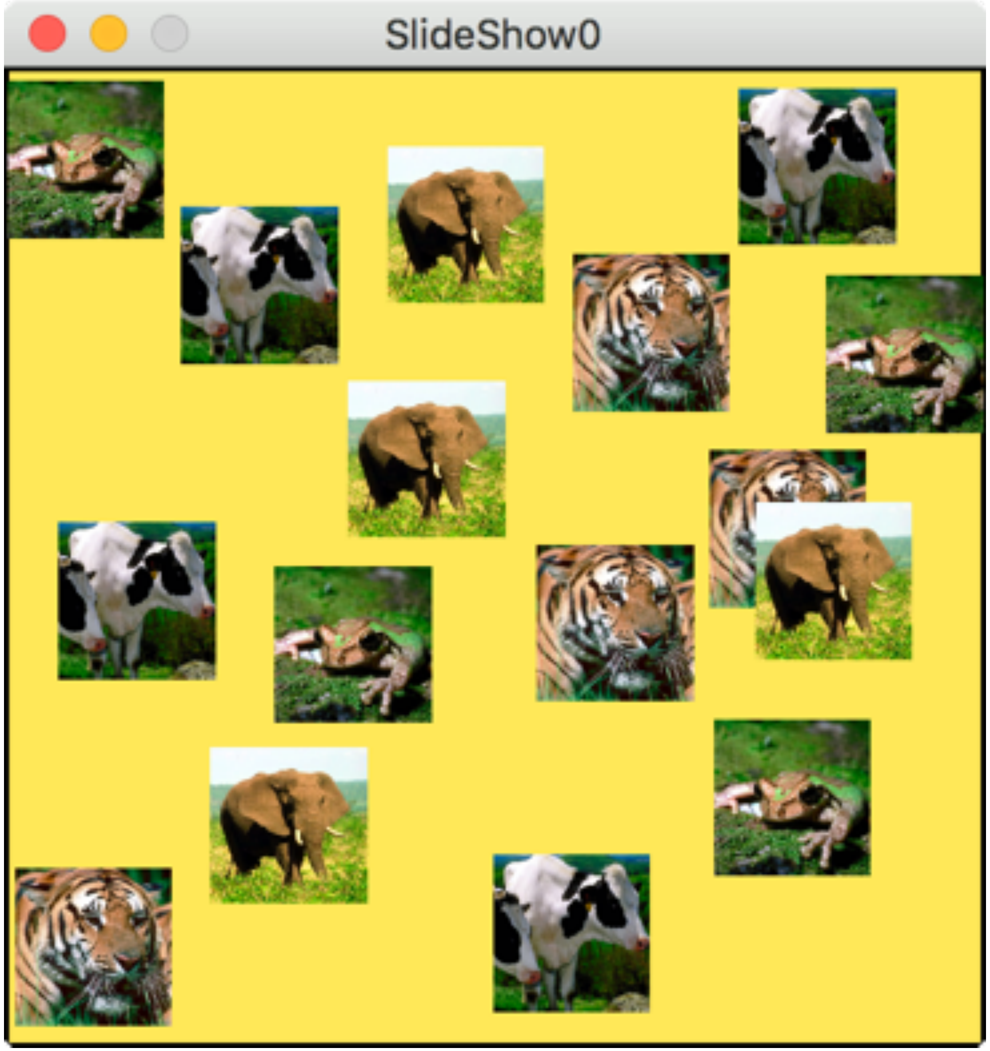
Example 2: Start & Stop



Which degree of interactivity?

http://www.laptoprepair101.com/wp-images/key/space-bar-key-4.jpg

Example 3: Arranging Pictures / Sprites



Which degree of interactivity?