(1)
Installer GL4D

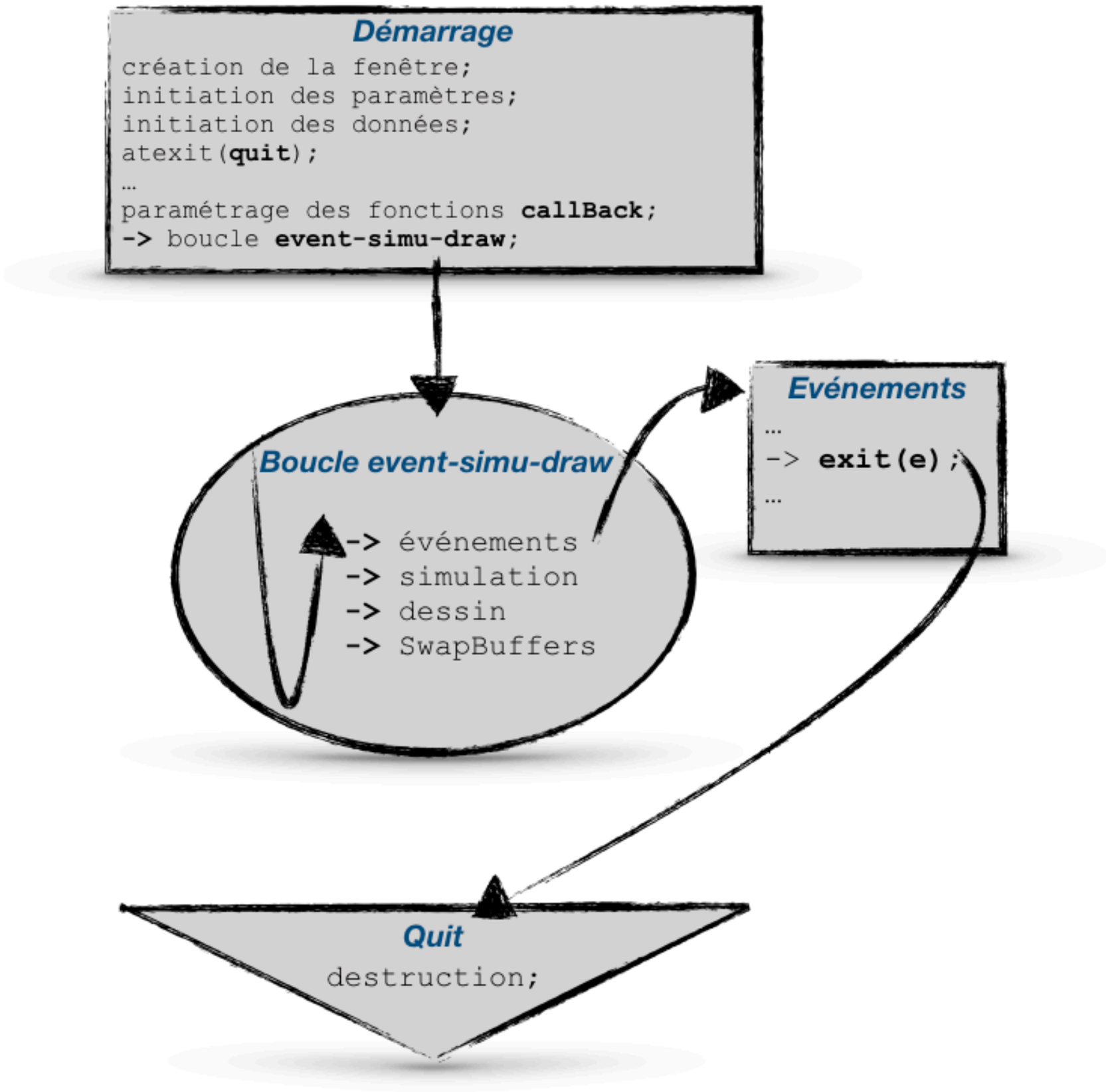
Tout est là https://gl4d.api8.fr/FR/
(On se donne une heure max)

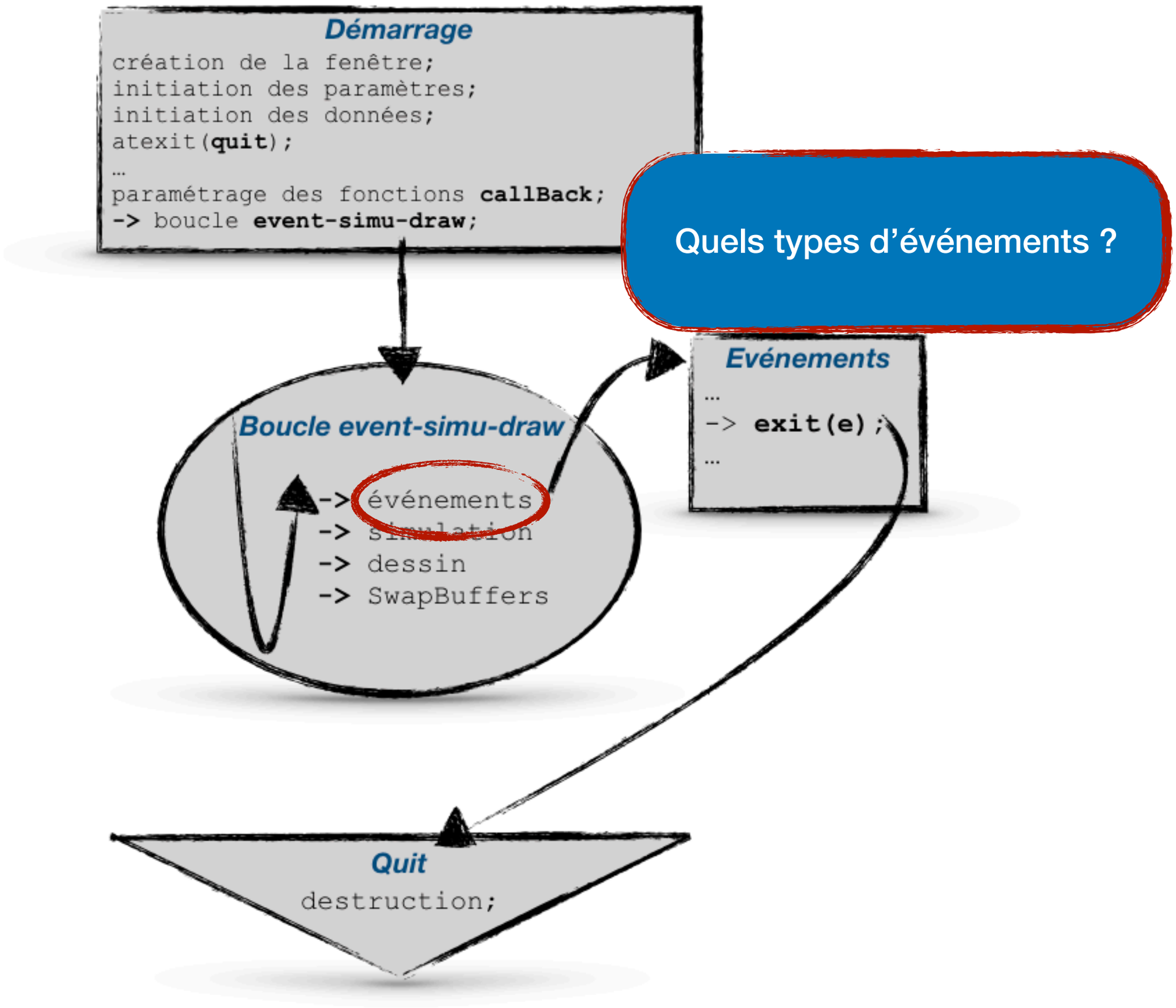
 —

(2)
Introduction à l'interaction

Appliquée à GL4Dummies

Voir la doc de référence à l'adresse https://gl4d.api8.fr/doxygen/html/index.html


Démarrage
création de la fenêtre;
initiation des paramètres;
initiation des données;
atexit(quit);
…
paramétrage des fonctions callBack;
-> boucle event-simu-draw;
Boucle event-simu-draw
-> événements
-> simulation
-> dessin
-> SwapBuffers
Evénements
…
-> exit(e);
…
Quit
destruction;


Démarrage
création de la fenêtre;
initiation des paramètres;
initiation des données;
atexit(quit);
…
paramétrage des fonctions callBack;
-> boucle event-simu-draw;
Quels types d'événements ?
Boucle event-simu-draw
-> événements
-> simulation
-> dessin
-> SwapBuffers
Evénements
…
-> exit(e);
…
Quit
destruction;

Quels types d’événements ?

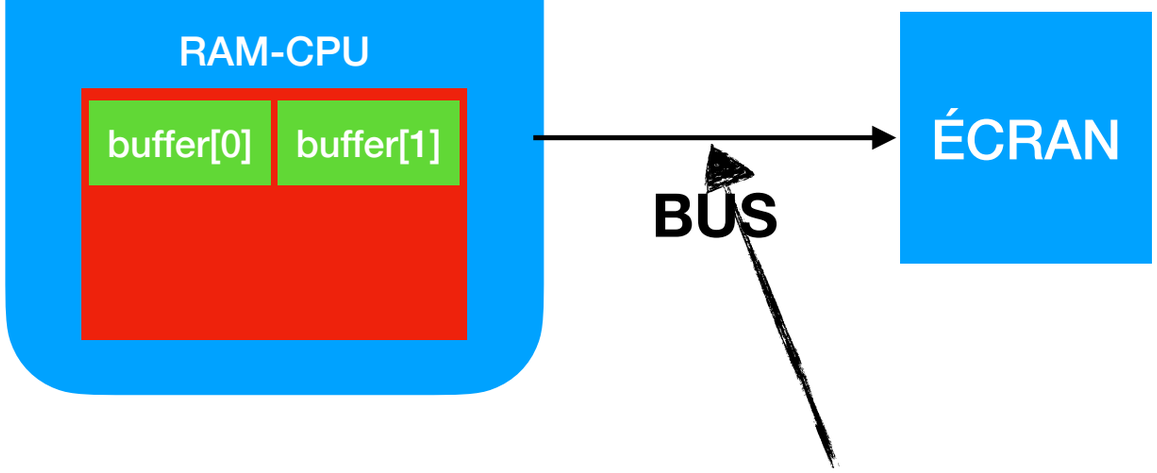
- Interactions clavier (plusieurs types)
- Interactions souris (plusieurs types)
- Actions sur la fenêtre :
    - retailler, perte de focus, fermeture, …
- Autres interactions possibles (extérieurs) :
    - Divers capteurs, caméra RGB, caméra RGBD, Motion Leap, …

Quels types d'événements ? (GL4Dummies)

- Interactions clavier (plusieurs types)
  - ➡ gl4duwKeyDownFunc(void(*)(int keycode) **mafonction**) => void **mafonction**(int code_de_latouche) { … }
  - ➡ gl4duwKeyUpFunc(void(*)(int keycode) **mafonction**) => void **mafonction**(int code_de_latouche) { … }
- Interactions souris (plusieurs types)
  - ➡ gl4duwMouseFunc(void(*)(int button, int state, int x, int y) **mafonction**) => void **mafonction**(int button, int state, int, x, int y) { … }
  - ➡ gl4duwMotionFunc(void(*)(int x, int y) **mafonction**) => void **mafonction**(int, x, int y) { … }
  - ➡ gl4duwPassiveMotionFunc(void(*)(int x, int y) **mafonction**) => void **mafonction**(int, x, int y) { … }
- Actions sur la fenêtre :
  - retailler, perte de focus, fermeture, …
    - ➡ gl4duw…
- Autres interactions possibles (extérieurs) :
  - Divers capteurs, caméra RGB, caméra RGBD, Motion Leap, …
    - ➡ **manuellement** dans gl4duwIdleFunc(void(*)(void) **mafonction**) => void **mafonction**(void) { … }

La parenthèse SwapBuffers (double buffering)
Améliorer la fluidité de l'application graphique

- **Deux buffers existent : buffer[0] et buffer[1]**
- **Mettre i, une variable d'indexation, à 0**
- **Début boucle**
- Calculer image et Écrire dans buffer[i]
- Quand terminé : demander le transfert depuis buffer[i] vers écran
  - Pendant ce temps le CPU est libre (non bloquant)
- Le CPU est donc libre de commencer à calculer l'image suivante
  - il modifie i en **i = (i + 1)%2** pour le faire dans l'autre buffer
    (si le transfert de ce dernier - démarré lors du précédent tour - est bien terminé, c'est généralement le cas)
- **Recommencer**



(3)
Écrivons ensemble une fonction qui dessine des cercles

Rappel de l'équation d'un cercle : $(x - x_0)^2 + (y - y_0)^2 = r^2$
où (x_0, y_0) est le centre du cercle et r son rayon

(AIDE AU SLIDE SUIVANT)

POINT DE DÉPART
https://github.com/noalien/GL4Dummies/tree/master/samples/sc_00_00_blank-1.0

il suffit de calculer ce tronçon
et reproduire le reste par symétrie

3

2

4

1

0

?

$$R \times cos(\frac{\pi}{4}) = \left\lceil \frac{R}{\sqrt{2}} \right\rceil$$

5

8

6

7

R est le rayon
du cercle

 —