FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Sprites

Laboratório de Computatdores

2021/22

Pedro Brandão

1

References

- These slides are based on:
  - Lectures notes from LCom 20/21 by Pedro Souto
    - Uses resources from João Cardoso (jcard@fe.up.pt), namely the Sprite code
  - João Cardoso, Notas sobre Sprites

With adaptations/additions by Pedro Brandão

- Images/Photos are licensed under CC BY-SA, CC BY-NC-ND or CC BY-SA-NC
  - Results from MS Office search

LCom 21/22 - Sprites

U.PORTO

2

2

LCom 21/22 - Sprites

U.PORTO

Objectives

- Isn't Sprite only a myth?
- Does a sprite move?
- Does a sprite animate (move in place)?

3

3

LCom 21/22 - Sprites

U.PORTO

Sprite

- Sprite "Two-dimensional image that is integrated into a larger scene. […] Originally, the term *sprite* referred to fixed-sized objects composited together, by hardware, with a background. Use of the term has since become more general." (Sprite (CG) Wikipedia)
- Allows the integration of independent pixmaps into a scene
- Allows image animation without altering the background – thus a sprite can be considered an overlay image

4

4

LCom 21/22 - Sprites

U.PORTO

Animation vs movement

- Movement: re-position the sprite from on a different (x,y)
- Animation: alter the sprite's image to make an animation
  - May also have movement associated (animate and move from (x1,y1) to (x2,y2)
  - Present a sequence of pixmaps

5

5

LCom 21/22 - Sprites

U.PORTO

A Sprite

```c
typedef struct {
  int x, y;            // current position
  int width, height;   // dimensions
  int xspeed, yspeed;  // current speed
  char *map;           // the pixmap
} Sprite;

/**
 * Creates a new sprite with pixmap "pic",
 * with specified position (within the
 * screen limits) and speed;
 * Does not draw the sprite on the screen.
 *
 * @param pic lines of strings, same
 * as xpm_map_t (has const protectors)
 * @return NULL on invalid pixmap.
 */
Sprite *create_sprite(const char *pic[],
          int x, int y, int xspeed, int yspeed) {
  // allocate space for the "object"
  Sprite *sp = (Sprite *) malloc(sizeof(Sprite));
  xpm_image_t img;
  if (sp == NULL)
    return NULL;
  // read the sprite pixmap
  sp->map = xpm_load(pic, XPM_INDEXED, &img);
  if (sp->map == NULL) {
    free(sp);
    return NULL;
  }
  sp->width = img.width;
  sp->height = img.height;
  //...
  return sp;
}

void destroy_sprite(Sprite *sp) {
  if (sp == NULL)
    return;
  if (sp->map)
    free(sp->map);
  free(sp);
  // caller should put the pointer to NULL
}
```

6

6

LCom 21/22 - Sprites

Pixmap and transparency and collision

- The pixmap uses black (or some unused color) for the background, which is assumed to be transparent
- When copying the colour bytes to video mem (or buffer)
  - If the colour to copy is the transparent one ➔ do not copy, leave video mem as is
  - Otherwise,
    - If the pixel to write to has a non-background colour (or the colour of another object that this one can collide with) ➔collision
    - Otherwise, copy colour bytes to video mem

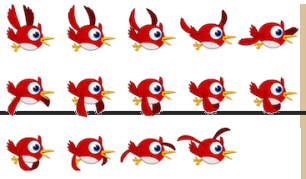
U.PORTO

7

7

LCom 21/22 - Sprites

An animated Sprite



```
typedef struct {
  Sprite *sp; // standard sprite
  int aspeed; // no. frames per pixmap
  int cur_aspeed; // no. frames left to next change
  int num_fig; // number of pixmaps
  int cur_fig; // current pixmap
  char **map; // array of pointers to pixmaps
} AnimSprite;

/**
 * Creates a AnimSprite given a set of xpm pics
 * @param no_pic number of xpm pictures
 * @param pics pointer to a cllection of images (lines of strings)
 * @return
 */
AnimSprite *create_animSprite(uint8_t no_pic, char *pic[][]);
```

U.PORTO

8

8

LCom 21/22 - Sprites

U.PORTO

Animation Speed

- Animation speed is measured as number of "frames" per pixmap
- Change the pixmap after the aspeed frames have been shown for current pixmap

9

9

LCom 21/22 - Sprites

U.PORTO

create_animSprite

```
AnimSprite *create_animSprite(uint8_t no_pic, char *pics[][]) {
  AnimSprite *asp = malloc(sizeof(AnimSprite));
  // create a standard sprite with first pixmap
  asp->sp = create_sprite(pics[0], 0, 0, 0, 0);
  // allocate array of pointers to pixmaps
  asp->map = malloc((no_pic) * sizeof(char *));
  // initialize the first pixmap
  asp->map[0] = asp->sp->map;

  // load he other pics
  for (int i = 1; i < no_pic; i++) {
    char **tmp = pics[i];
    xpm_image_t img;
    asp->map[i] = xpm_load(tmp, XPM_INDEXED, &img);
    if (asp->map[i] == NULL || img.width != asp->sp->width || img.height != asp-> sp->height) {
      // failure: release allocated memory
      for (int j = 1; j <= i; j++) // free (NULL) is OK
        free(asp->map[i]);
      free(asp->map);
      destroy_sprite(asp->sp);
      free(asp);
      return NULL;
    }
  }
  return asp;
}
```

10

10

LCom 21/22 - Sprites

U.PORTO

Summary

- A sprite structure
- An animated sprite structure
- Transparency and collision

11

11

LCom 21/22 - Sprites

FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

U.PORTO

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Questions/ Comments

12

12