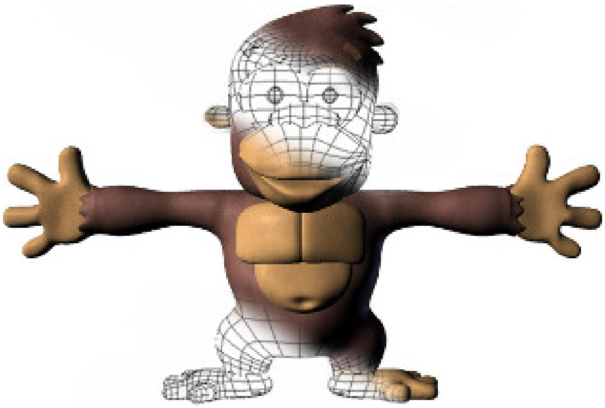
B2 - C Graphical Programming

B-MUL-200

my_defender

Who said TDs are out fashioned ?



1.3

my_defender

binary name: my_defender
repository name: MUL_my_defender_$ACADEMIC_YEAR
repository rights: ramassage-tek
language: C

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

This project is a Tower Defense game. A Good example of this kind of games is Revenge of the titans
Your main challenge for this game will be to create nice and smooth user interface and menus.

Your game must follow the following rules:

- The player is a builder who must defend his castle,
- Enemy waves will regularly appear from one side of the playground,
- The player must buy and place buildings on the playground grid to block/kill enemies,
- Buildings can be offensive (towers attacking) or defensive (basic walls),
- When the castle is reached by an enemy, it takes damages,
- If the castle reaches 0 hit points the player loose.

Having a pleasant user interface is one of the details that makes a good quality game, this project is the occasion for you to try your best on that topic.

You should look for information about how to create interface in video game.

Requirements

Mandatory

The following features are **mandatory** (if your project is missing one of them it will not be evaluated further):

- the window can be closed using events,
- the game manages the input from the mouse click and keyboard,
- the game contains animated sprites rendered thanks to sprite sheets,
- animations in your program are frame rate independent,
- animations and movements in your program are timed by clocks.

Must

The game **must** have:

- a starting menu with at least two buttons, one to launch a game, and one to quit the game,
- an `escape` key to pause the game when launched,
- a menu when the game is paused with at least three buttons, one to resume the game, one to go to the starting menu and the one to leave the game,
- at least 4 different types of buildings (e.g. walls, slowing towers, damaging towers, etc...),
- a building menu showing the different available buildings and their price,
- at least 2 different sound effects, and one looping background music.

> The starting menu and the game **must** be two different scenes.

Should

The game **should** have:

- a windows that sticks between 800x600 pixels and 1920x1080 pixels.
- a "How To play" menu, explaining how to play your game.
- a stored scoreboard.
- a scoreboard displayed at the end of a game, or thanks to a scoreboard button in the starting menu.
- buttons with at least three visual states: idle, hover, and clicked.
- a way to skip eventual cut scenes or an animated intro.
- different types of enemies with different speed and hit points.

Could

The game **could**:

- let the user upgrade its buildings,
- load buildings from files,
- take buildings files as command line argument,
- take buildings files from a menu inside the game,

- have a skill tree, unlock different types of buildings,
- have a "settings" menu that contains sound options and/or screen size options.

The size of your repository (including the assets) must be as small as possible.
Think of the format and the encoding of your resource files (sounds, music, images,...).
An average maximal size might be 15MB, all included.
Any repository exceeding this limit might not be evaluated at all.

AUTHORIZED FUNCTIONS

All the functions from the CSFML and the math library are allowed.
From the libc, here is the full list of authorized functions:

malloc
free
memset
(s)rand
getline
(f)open
(f)read
(f)close
(f)write
opendir
readdir
closedir

Any unspecified functions are de facto banned.

Damn! How can you have a random seed without using time??