DO-178

Axel BANAL - Antoine LEBEURY

1er février 2019

Table des matières

1 Software Requirement Data

1.1 System Requirements

SR.1 The game should be displayed on the touch screen, and the player's ship should move to where the user touches

SR.2 The player's ship should fire missiles and destroy enemies with it

SR.3 Enemies should appear randomly at the top of the screen and progress toward the bottom

1.2 High Level Requirements

HLR.1.1 The current position of the player's ship and enemies should be updated and displayed at all time

HLR.1.2 The player input on the touchscreen should be recorded

HLR.2.1 The player's ship should fire missiles continuously

HLR.2.2 Missile should continuously move toward the top of the screen

HLR.2.3 When missiles collide with an enemy, they both should be destroy

HLR.3.1 Enemies should appear randomly regularly

HLR.3.2 Enemies should appear at the top of the screen, with their positions randomize between the most left and the most right column

HLR.3.3 Enemies should continuously move toward the bottom of the screen

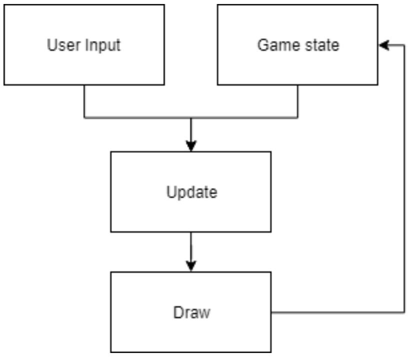

FIGURE 1 – Overall architecture

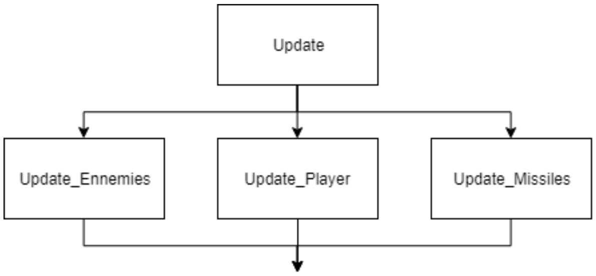

FIGURE 2 – Update architecture

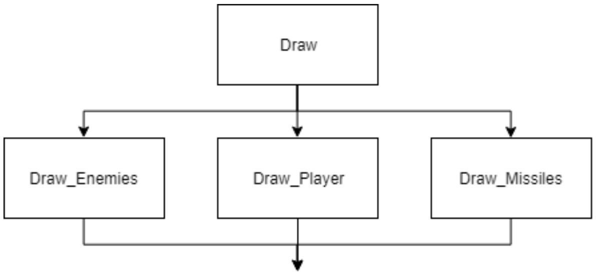

FIGURE 3 – Draw architecture

1.3 Software Architecture

1.4 Low Level Requirements

LLR.1.1.1 Update and Draw are called at every step

LLR.1.1.2 Draw does not display DEAD elements

LLR.1.1.3 Draw displays ALIVE and DMG_DEALT enemies differently

LLR.1.1.4 The player's ship is created ALIVE

LLR.1.2.1 When calling Update, the player's ship moves to the last position the player touched on the screen

LLR.2.1.1 A missile is created when the timer for missiles is over and the maximum number of missiles is not reached

LLR.2.1.2 Missiles are created on the player's ship position

LLR.2.1.3 Missiles are created ALIVE

LLR.2.2.1 Update_Missile is called with a fixed frequency

LLR.2.2.2 Update_Missile moves every ALIVE missiles toward the top of the screen

LLR.2.3.1 When update is called, if a missile and an enemy collide, the missile becomes DEAD and the enemy DMG_DEALT

LLR.3.1.1 An enemy is created when the timer for enemies is over and the maximum number of enemies is not reached

LLR.3.1.2 Enemies are created ALIVE

LLR.3.2.1 Enemies are created at the top of the screen. Their X position is fixed, but their Y position is random within the screen

LLR.3.3.1 Update_Enemies is called with a fixed frequency

LLR.3.3.2 Update_Enemies move every ALIVE enemies toward the bottom of the screen

LLR.3.3.3 Update_Enemies change DMG_DEALT enemies to DEAD ones

2 Design Description and trace data

Spaceship : Contains the position of the player.

- move : changes the player's position.

Ennmie : Contains the position and state of an enemy.

- appear_enn : makes an enemy ALIVE (LLR.3.1.2, LLR.3.2.1)
- move_enn : changes an enemy position towards the bottom of the screen and updates DMG_DEALT ones to DEAD ones (LLR.3.3.2, LLR.3.3.3)

Missile : Contains the position and state of a missile

- appear_mis : makes a missile ALIVE (LLR.2.1.2, LLR.2.1.3)
- move_mis : moves a missile toward the top of the screen (LLR.2.2.2)

Update : Updates every position of ALIVE elements and checks for collisions between missiles and enemies (LLR.1.2.1, LLR.2.1.1, LLR.2.2.1, LLR.2.3.1, LLR.3.1.1, LLR.3.3.1)

Draw : Displays the current state of the game (LLR.1.1.2, LLR.1.1.3)

Update and Draw are placed in an endless loop with a timer (LLR.1.1.1)

The player's ship is initialized at the bottom of the screen and ALIVE (LLR.1.1.4)