Nanouche Hamza
Pomès Hadrien

DO-178 Report

Software Requirement Data

System Requirements

SR.1 Game should be displayed on the touch screen.
SR.2 The snake should move following direction input by player on touchscreen.
SR.3 Apples should appear at random position.

High Level Requirements

HLR.1.1 Current position of each elements should be updated and displayed on the screen.
HLR.1.2 Player's input on the touchscreen should be recorded.

HLR.2.1 The snake should always move.
HLR.2.2 When snake's head collides with an apple, the apple should disappear.
HLR.2.3 When snake eats, it grows by 1.

HLR.3.1 When snake crosses screen bounds, it should appear at opposite side.
HLR.3.2 Snake should die when it collides with itself.

Software Architecture

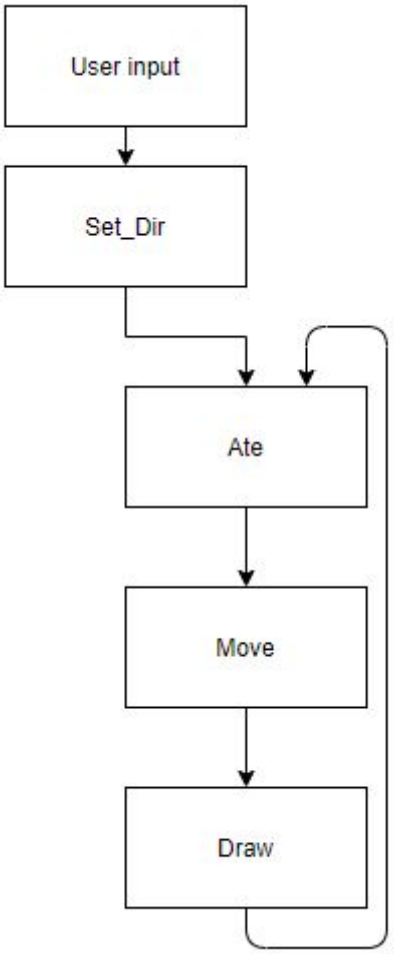

Figure 1. *Overall architecture*

Low Level Requirements

LLR.1.1.1 Draw is called at every steps.
LLR.1.1.2 Draw displays snake cells and apples.

LLR.2.1.1 Move and Ate are called regularly.
LLR.2.1.2 When Move is called, snake goes toward last direction imputed.
LLR.2.1.3 When an apple is eaten, Move should add one cell to snake's tail.

LLR.2.2.1 A new apple is created when Ate returns true.
LLR.2.2.2 New apple is created at random position

LLR.3.1.1 Snake should be created with length of 2.
LLR.3.1.2 Snake should be created at predetermined position.

Design Description and trace data

Snake : Contains position and direction of the player.

- Create_Snake: Create new snake with length of two at fixed position (LLR.3.3.1, LLR.3.3.2).
- Move: Updates snake's position and length (LLR.1.2.1, LLR.1.2.2).
- Is_Alive: Checks if snake is biting its tail.

Ate : Returns True when an apple is ate and creates a new one randomly when necessary (LLR.2.2.1, LLR.2.2.2).

Draw : Displays snake and apple at current state of the game (LLR.1.1.2).

Move, Ate and Draw are placed in an endless loop with an iteration counter (LLR.1.1.1, LLR.2.1.1).

Source code : https://github.com/Pomsss/ada_snake