Emergent Architecture Design

Group 5

Team Members

Liam Clark	4303423
Mitchell Hoppenbrouwer	4243889
Martin Koole	4217500
Ike Rijsdijk	4294106
Jorai Rijsdijk	4282264

1. Introduction

This document will have a high level description of software and hardware components of our games project.

1.1. Design goals

- **Responsiveness**
  The input of the player should be processed within a reasonable time. If it takes too long, the players won't have the feeling of satisfaction when creating their art. It will cause frustration, so it is an important design goal. The game needs to feel responsive.
- **Easy use**
  Players should be able to play the game easily. It doesn't require any hardware on the player's end. It has no learning curve, the player should be able to play the game instinctively.
- **Audience**
  Our audience consist of people in public places with spare time on their hands. They can be of all ages. We want the design the game to have people take their time. We don't want to rush them.

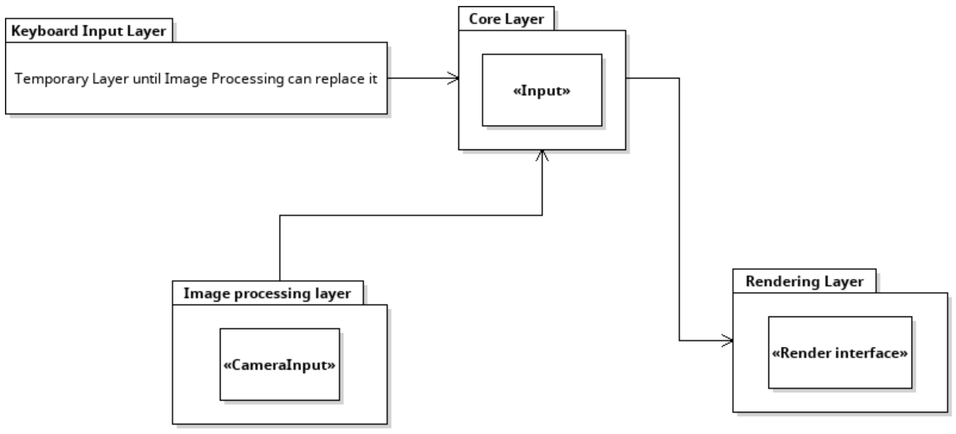
2. Software architecture views

This chapter discusses the architecture of the system. The system is first decomposed into smaller subsystems and the dependencies between the subsystems are explained. In the second paragraph the relation between the hardware and software of the system is elaborated. The third paragraph illustrates the data management of the system.

2.1. Subsystem decomposition (sub-systems and dependencies between them)

The subsystems are simple. We want them to do a job and do it well. The core layer and the rendering layer are intertwined by LibGDX, but separate enough to see as different layers.

- **Image Processing Layer**
  All player input comes from processed camera input of the player this layer should parse the input to an understandable format for the core layer.
- **Keyboard Input Layer**
  Instead of using image processing, a keyboard can be used to paint. This will be used to test the game while the image processing is being developed. When image processing is ready it will replace this layer.
- **Core Layer**
  The input from either the image processing layer or the keyboard input layer is converted to movement of the player. These player movements will be send to the rendering layer. The core layer does contain a package called input, this package converts the input from the image processing layer into gameplay.
- **Rendering Layer**
  The canvas of the painting needs to be rendered and changes need to be applied every cycle.



2.2 Hardware/software mapping

- **Camera**
  The camera observes the player in the playing field. It sends all its output to the image processing system.
- **Image Processing System**
  The image processing system takes the output from the camera and processes it to detect players entering or leaving, detect brushstrokes being made by players and their requests to change the current colour. It sends the information over a LAN network to the core system. The image processing cost a lot, which is why it is on a different system.
- **Core System**
  The core system consists of the core layer and the rendering layer. It takes the image processing output and performs the detected actions in the game. It then renders the image and sends it to the beamer. While the keyboard input layer is being used as a replacement for the image processing layer, it will be on the core

system. Since the cost of the keyboard input is less compared to image processing, this is not a problem.

- **Beamer**
  The beamer receives the image from the core system and displays it.