

Challenge - Pairs Game

Emmanuel Miras	Zoe Wall
40168970@napier.ac.uk	40182161@napier.ac.uk
Edinburgh Napier University - Fundamentals of Parallel Systems (SET09109)	

1 Requirement 2

Within the original implementation the Player Manager process acted as a server and the Controller Manager process acted as a client. In order to implement the required changes, the roles had to be switched. This proved to be quite straightforward.

2 Requirement 3

2.1 Process Network

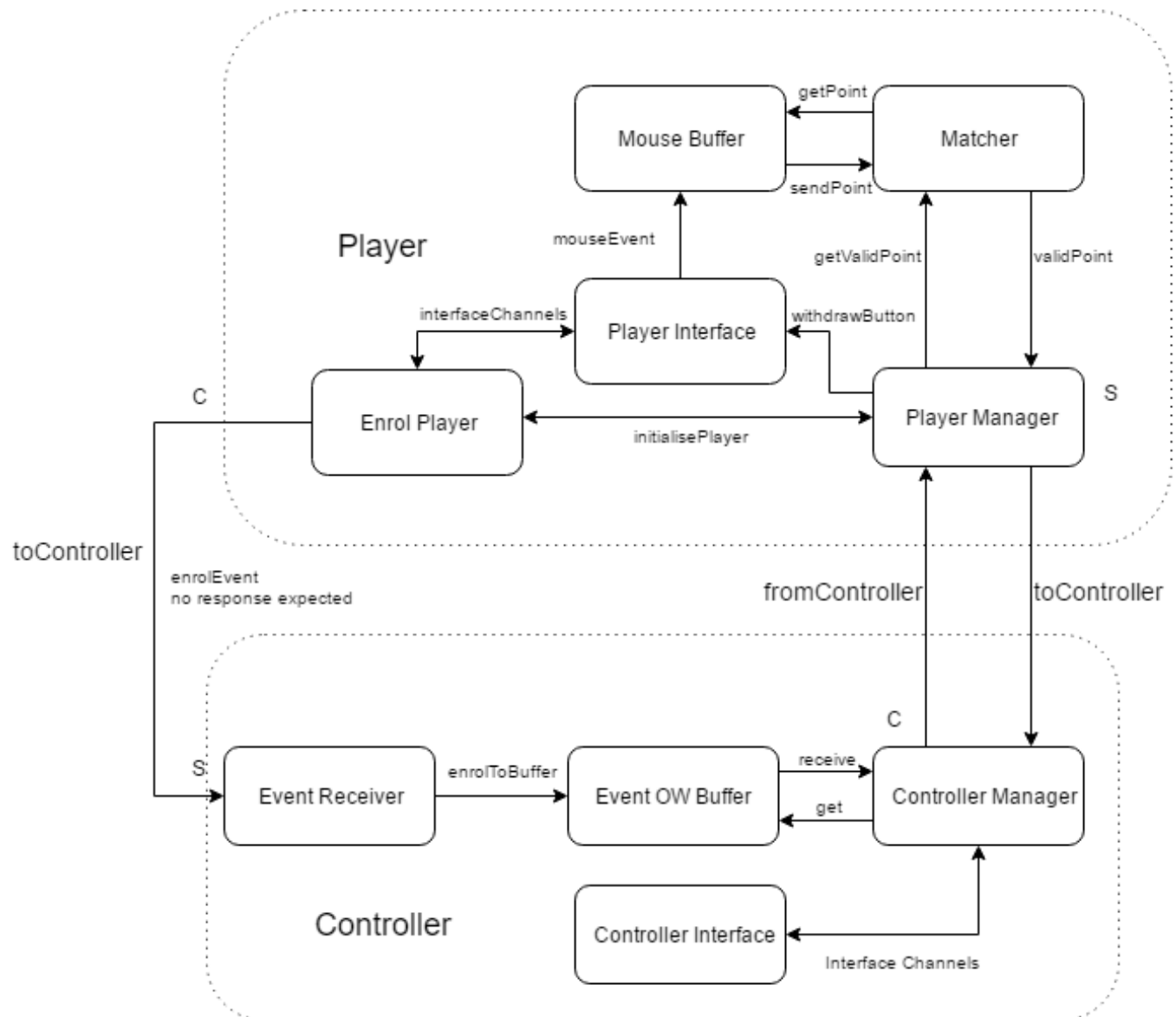


Figure 1: **Process Network** - diagram to show channel connections in the Pairs Game network.

2.2 Channel Interaction

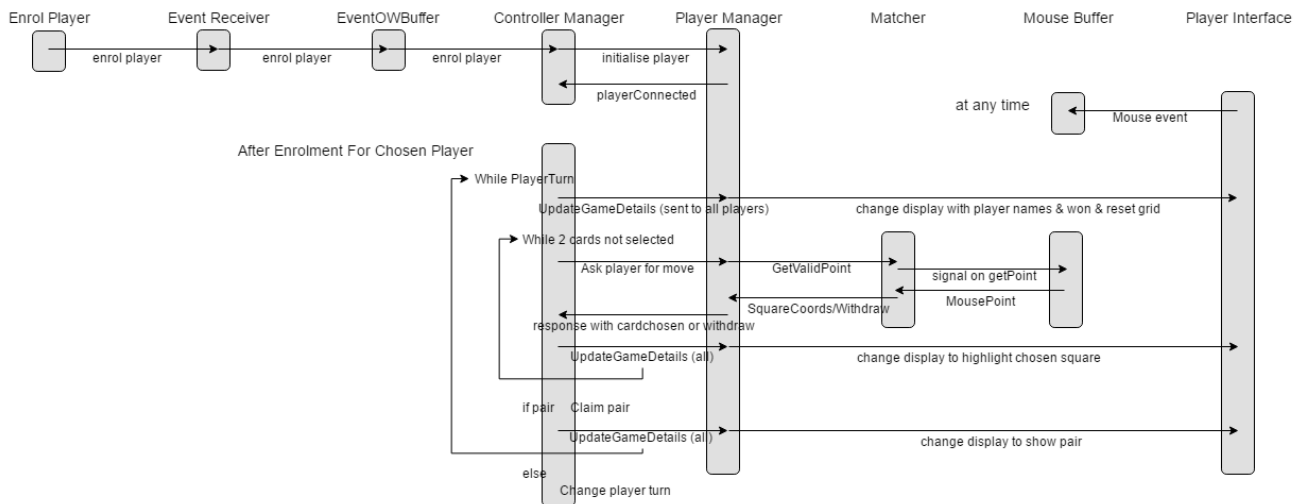


Figure 2: **Channel Interaction Sequence** - diagram to show the sequence of interactions between channels.

2.3 Explanation

The above diagrams detail a modified version of the Pairs Game system to add the functionality of taking turns and dynamically updating every players interface to match the game state.

Turns To implement player turns, the Controller Manager must keep track of every player initialised within a list. To initialise a player, a modified EnrolPlayer object, which contains the location of the player sending the request and the player id, is copied to the Controller Manager. Once the player has been enrolled they are added to a list of current players, which the Controller Manager can iterate through to request for card selections and send a map of square coordinates for updating the game state.

Updates The main change to the communications structure is to modify the behaviour of both the Player Manager and Controller Manager. The proposed solution is to have multiple channel inputs to the Player Manager that are allocated dynamically once the player has been enrolled. In doing so, each of the Player Managers can act as pure servers. This means that the Controller Manager can issue requests to every Player Manager to update their interfaces without the risk of dead-lock. With the current structure, this causes an issue with the enrolment of players as they must initiate the connection to the Controller Manager therefore acting as clients. To solve this, the enrolment logic from the Player Manager is abstracted into another process which can act as a client. An enrolment request can therefore be copied through an event handling system using an overwriting buffer to the Controller Manager client that obeys the client-server pattern.

3 Requirement 3

3.1 Group H

The game was fully operational as intended

3.2 Group J