Lily Wang and Igor Geller
6.111 Final Project Proposal
November 2, 2007
TA mentor: Brian

# Interactive Tetris

## 1. Introduction

We propose to build an interactive game of Tetris where two players using separate labkits and terminals will be able to affect the difficulty of their opponent's game by clearing more rows on their own game.

Shown below is the block diagram of the modules involved in this project.

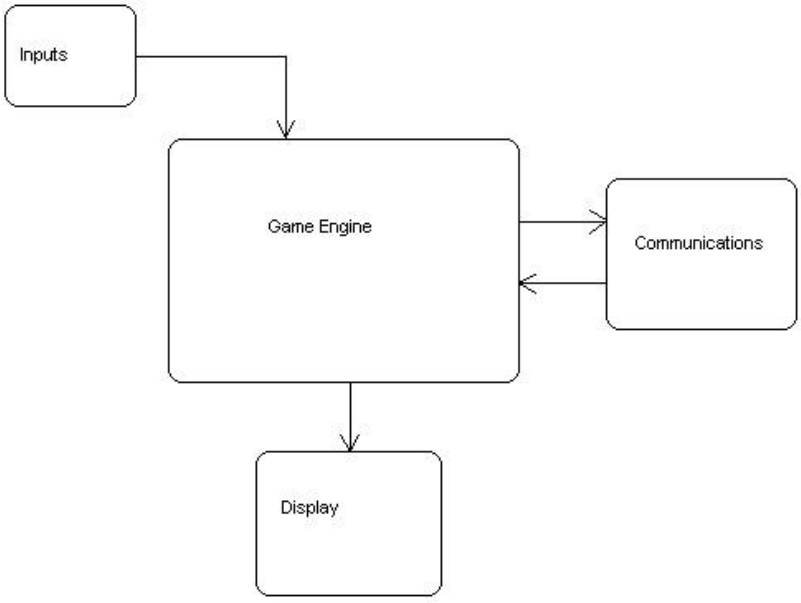


Figure 1. Simple block diagram of the modules in the proposed system.

## 2. Module Descriptions

### 2.1. The Inputs Module

This module interfaces user inputs from the keyboard to the game engine. Pressing keys on the keyboard will serve as interaction for each player, which will be worked out in the logic of the game engine.

### 2.2. The Game Engine Module

The game engine will consist of three parts:

Part 1: movement of an active shape defined in a 4-by-4 square based on the inputs to the module.

Part 2: Transfer of the "active" status from a given shape to a newly generated one at the top of the screen.
Part 3: Updating the stored values after the completed row or a signal from communication module.

Description: The game engine keeps the current status of the game stored in the BRAM. The color of a given square (x, y) can be found in the address {x, y} of the BRAM. There are *X_shape_location* and *Y_shape_location* registers that keep track of where the "active" piece is. The two registers represent the top-left coordinate of where the piece is located in a virtual 4-by-4 grid. The different shapes and different rotations will be kept in the seven shape-states, each one having four sub-states. (The default rotation will be clockwise).

### 2.3. The Communications Module

This module handles the serial port communications between the labkits of two Tetris opponents. It takes as input a signal *Num_Rows_Cleared* from the game engine module which it sends in a packet over the serial port to the opponent's labkit, causing his/her game speed to increase by a number proportional to *Num_Rows_Cleared.*

### 2.4. The Display Module

This module takes as input the BRAM holding the colors of each square in the game grid. The display is refreshed at every update of the game. To access the color of each square, for each x from 1 to 10 and for each y from 1 to 20, the module reads from the address resulting from the concatenation of the x and y coordinates. The output is a pixel at the appropriate square of the corresponding color.

## 3. Testing

The inputs module will be tested by hooking up signals to the LED display on the labkit, and checking that appropriate keyboard functions light up the correct LEDs.
We will test the display module by feeding in as input a test BRAM filled with hardcoded data, and checking to see if it displays as expected on the screen.
After finishing the display module, we will be able to plug it directly into the game engine module for testing.
The communications module will be tested by feeding the packets from the serial port in a loop into itself, and checking the results with what is expected.

## 4. Division of Labor

Igor will be responsible for the inputs and game engine modules, and Lily will complete the display and communications modules.