

CSC258 Project Proposal

	First Name	Last Name	Student No.	Email
Member 1	Tiago	Ferreira	1004305245	t.ferreira@mail.utoronto.ca
Member 2	Pan	Chen	1005430776	pan.chen@mail..utoronto.ca

What is the title of your project?

Whack a Mole

Provide a one-paragraph description of your project.

This project is a game. The game will fire LED's at random times each level and the user must click the corresponding button to "whack the LED". The game will feature a fixed speed and a fixed number of moles. The goal is to whack as many LED's as you can. If you take to long, then you will not receive the point. Your score is then displayed on a HEX-display and represents the amount of LED's you "whacked".



Project Proposal



DETAILS

SUBMISSIONS

GRADE

ADD A COMMENT

PAST DUE DATE

This cannot be accepted as the final project as the amount of workload is not comparable with 3 labs. You might want to define a new project or re-consider the game such that you use keyboard, VGA, random generation of moles (LEDs), etc.

**Hossein Shafieirad**

Jul. 16, 2019 9:06 p.m.

**Online eFile upload**

Handed I... Jul. 12, 2019 8:19 p.m.



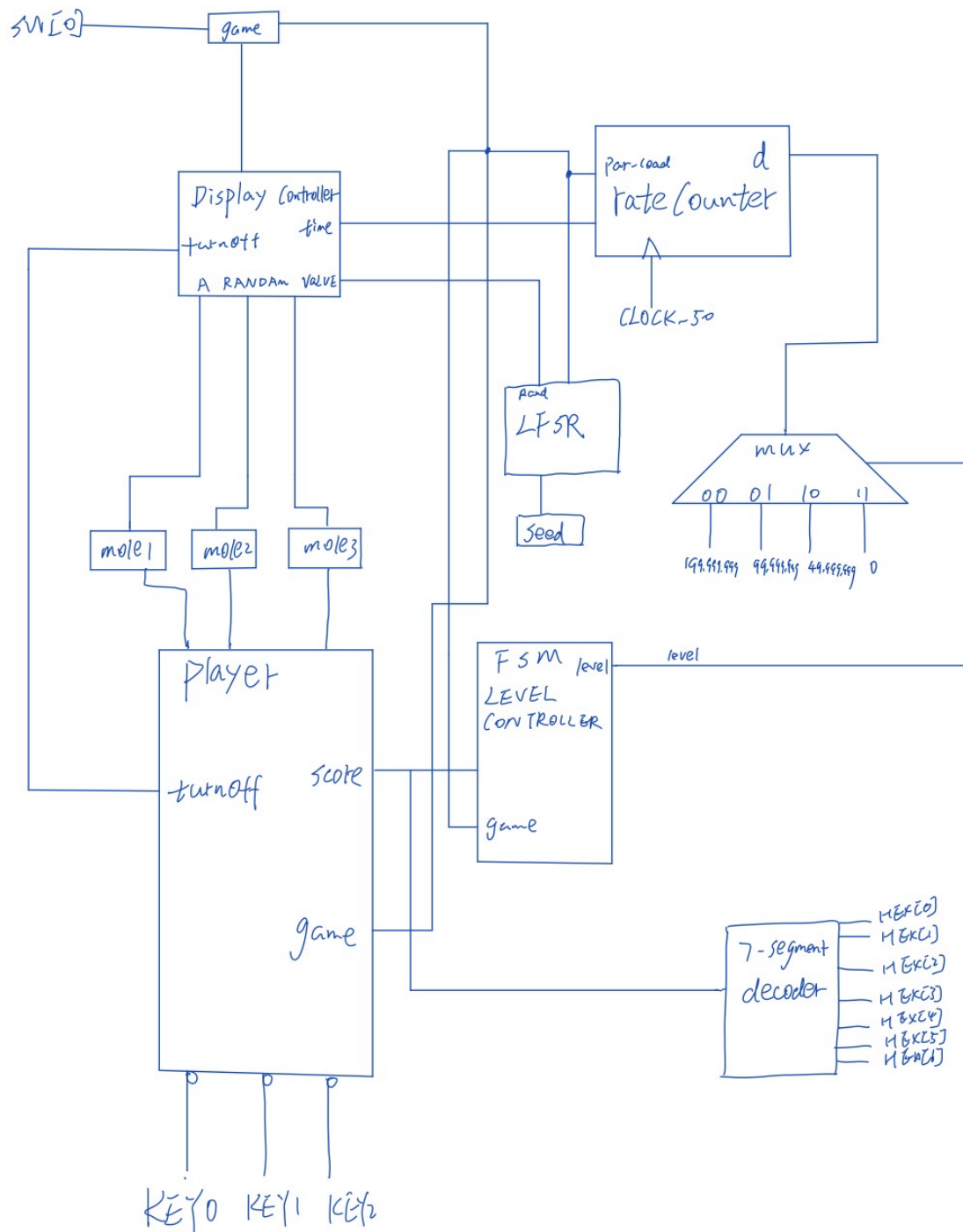
Dashboard



6

Project Description:

Design Schematic (For the core part (except VGA part)):



Design Specifications:

1. A fixed number of LEDs (moles) will be on for a fixed length of time for each state. If the player pushes the corresponding key before the LED automatically turns off, the player earns a certain number of points.
2. rateCounter from lab5 will be used to control when the LEDs should turn off.
3. When player clicks the “start” signal, the game will start.
4. After the player earns a certain number of points, he will enter the next state, with higher speed. (There are four different states.)
5. **Player will use A S D in the keyboard as their ‘Whacker’.**
6. **The game (the moles, the score ...) will be displayed on the screen of the lab computer by VGA.**

Pseudo-code of algorithm

For each LED's turning on / off controller

 If start

 Choose a random LED to display

 While (rateCounter != a_fixed_frequency)

 LED keeps on

 LED turns off.

For player gaining scores controller

If LED[0] turns on and KEY[0]:

 Score += this_state_point

 turnoff = 1

If LED[1] turns on and KEY[1]:

 Score += this_state_point

 turnoff = 1

If LED[2] turns on and KEY[2]:

 Score += this_state_point

 turnoff = 1

If LED[3] turns on and KEY[3]:

 Score += this_state_point

 turnoff = 1

Frequency change controller (like a FSM):

if score == level1Max or level2Max or level3Max:

 update = 1

else

```
    update = 0
State 1
If update:
    Go to State 2
State 2
If update:
    Go to State 3
State 3
If update:
    Go to State 4
```

What will you accomplish for the first milestone?

We will first implement the module (Display Controller) that chooses a random LED to display. We will use the Linear-feedback shift register to do this part of work.

Also, for the first milestone, we will add feature to this module which turns off the LED after a certain time even if the player doesn't push the corresponding key. (In this case, the player doesn't earn any point). We will use the rateCounter from lab5 to do it.

And we also will implement another called Player Controller. To better focus on the feature that once a player push one key (**in the third milestone, we will make it one of A S D in the keyboard**), the corresponding LED will turn off. And the LED is on before, the player will earn a certain point, we will mock two LEDs, one (A) is on and the other (B) is off. And we will test if the A turns off when player push KEY[0] and earn point as well as if B keeps off status and the player doesn't earn point after pushing KEY[1]. And in the next milestone, we will connect the LED of Player Controller with the LED of the Display Controller.

What will you accomplish for the second milestone?

As said above, for this milestone, we will connect the LED of Player Controller with the LED of the Display Controller. (This is for test, and in the third milestone, we will let the 'moles' connected to the VGA instead of LED for a better experience)

And we will create a higher level called game controller which uses the ports from FPGA board directly. We will use SW[0] for Go, which indicates that the game starts, CLOCK_50 for frequency calculation, KEY[0], KEY[1], KEY[2], KEY[3] for the player's choice, and LEDR[0], LEDR[1], LEDR[2], LEDR[3] to stand for the four 'moles'.

And we will use a HEX to show the player's score.

And we will improve the player's module by adding a rule that if the player cheats by clicking multiple 'buttons' at the same time, he won't earn any point.

What will you accomplish for the third milestone?

After the first two milestones, we will be proud to see that our project works. The flow is that now the game only supports one level, that is, the LED display time is always the same no matter how many points the player has already obtained.

So, for this third milestone, we will use the knowledge of FSM to create a Level Controller. The Level Controller takes player's score as input, and depending on whether the score has reached a certain value, the state may move to the next state, with shorter time for the LED's displaying. And the output of the FSM, level, will be the input to a 4-to-1 MUX whose possible outputs are 199,999,999, 99,999,999, 49,999,999 and 0. And the output of mux will be the input of rateCounter in order to change the frequency.

And for this third milestone, we will in the end try to use keyboard and computer screen to control our game for a

better player's experience. We will let the image of a cartoonized mole display on the screen.

How does this project relate to material covered in CSC258?

This project will use multiple topics covered in CSC258. It will require sequential circuits to design. It will represent a FSM with multiple states. It will feature a HEX display for score output. It will use clock frequency for random LED Generation and other things. It will use other modules and topics learned in this class. **And it will use VGA knowledge to put our game on the screen and use a keyboard to control game.** Finally, it will be programmed in Verilog which is a big part of the course.

What's cool about this project (to CSC258 students and non-CSC258 students)?

To CSC258 students it is cool to them because it is a unique idea that we have not seen in past 258 projects. It will require unique knowledge of what was learned in this class, such as rateCounter, VGA, how to use a keyboard to control signals To non-CSC258 students it is cool to them because they will not understand the process to actually make the game so they will be amazed and wonder how it is possible to actually make something like this using hardware.

Why does the idea of working on this appeal to you personally?

We used to have whack-a-mole when we were younger kid and this project gets its idea from that. It is interesting to us to see how we can algorithmically implement this using circuit and hardware knowledge.