CSC258 Project Proposal

	First Name	Last Name	Student No.	Email
Member 1	Tianjiao	He	1003920215	judith.he@mail.utoronto.ca
Member 2	Zhuosi	He	1003436203	zhuosi.he@mail.utoronto.ca

What is the title of your project?

Volcano scientific drome operation

Provide a one-paragraph description of your project.

This project is a game of controlling planes by players to avoid obstacles along the way until they crash into obstacles. At the beginning of the game, the obstacles would show up randomly at a slow rate, and the gap allowing the plane to go through is wide. The width of the gap is decided randomly behind the scene. As the game continues, the gap will show up with an increasing speed using rate divider.
There would be 2 types of obstacles. One is static, tube-like mountains, and the other is lava drops moving in horizontal directions coming towards the plane. The scene of the game is moving simultaneously rolling towards the left. The player can operate the plane through key0(downwards) and key1(upwards) on the DE1-SoC board to dodge the tubes and lava. The Hex lights on the board represents the score for the current player. The time that the player remains alive represents the score that the current player has. Each player can have 3 lives. Once they crash with some obstacles, they lose one life, and they will lose the game when they lose all of their lives. The final score for the player represents the length of time for the player to remain alive.

Project Description:

(This is where you describe your project in detail. You can use the Design Case Studies slides as a reference on how to create the following components for your project. All of these components are not compulsory for you to have but most projects usually have these components. They are: high level pseudo code, state diagram, datapath-and-control-CU block diagram, input/output block diagram [Example: https://www.nandland.com/goboard/images/project10-pong-block-diagram.png], truth tables etc. These components should be designed and described to show your understanding of your design i.e. how many bits is each input/output, what is the max number your counter can count up to, how many counters/shift registers you need to use etc. A good idea is to get an initial draft of this done and show to your TA in the next lab or during office hours to get feedback.)

Pseudo-code

ALU -> generate random numbers through modular calculation.

-> controlling which state the game is in

-> control the appearance of lava drops, item and the width of the gap between 2 tubes

(optional) -> light up the led according to the note of music

Rate divider -> control the rate of screen rolling and the gap's width for lava drop coming out

Score Counter(asynchronous) -> track current player's score, +10 if user dodge the obstacles

FSM -> track state of the game

   Note: if the pixel of the plane crash with the pixel of the obstacles, the player will lose one life

Time Counter(item) -> track how much time the unbeatable state lasts for. (2s)

Time Counter(game_time) -> statistic on the total game time

Speed_changing Counter -> once the speed_changing counter reaches its limit(59 seconds in normal, 29 in hard), the speed of the game will be doubled and the counter (the counter in rate divider would be halved)will go back to 0. Then it will start another cycle.

Counter(synchronous) -> tracking how many lives are left. By default, full life is 3. Each crash would subtract 1 from it. When it reaches 0, it is the game over state.

PIN Assignments (input/output)

KEY[0] -> plane going down

KEY[1] -> plane going up

SW[0] -> resetn

SW[1] -> difficulty control(default:0 -> normal; 1->hard)
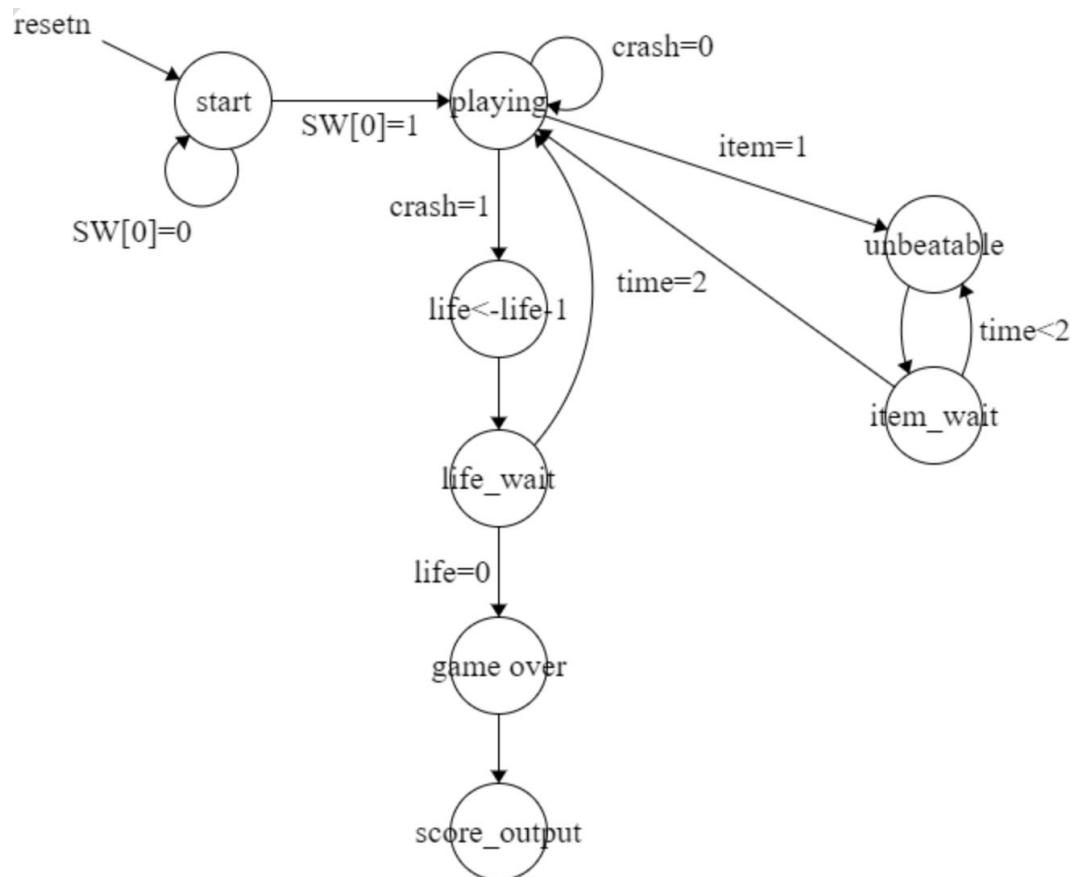
HEX0,1,2,3-> show score

HEX5-> show the life remaining

VGA -> show the game scene

(optional) LEDR[9:0] -> showing the note for music

1. General workflow

What will you accomplish for the first milestone?

(Advice here: Pretend that you're designing Lab 8 around your project idea, in keeping with the difficulty level of the previous labs. Try to be **specific and detailed** in describing the components that you will complete. Don't say that you'll "think about" or "plan" or "design" something.
Bad example 1: We build the graphical interface.
Good example 1: We display moving spaceships on VGA.
Bad example 2: We write code for the PS2 keyboard interface.
Good example 2: We make the PS2 keyboard work and show the key inputs on the HEX display.
Assume your project can be developed in three independent parts, what you write in the space below should outline the components of the first part. Make sure to describe a full lab's worth of work, including the evidence of your work that you will provide to the TAs to justify getting the full marks for this milestone.)

1. We display the tube-like mountains without the randomized generated height in VGA and the lava drop is always coming from the top most level on the screen.
2. We can control the plane up and down by pressing KEY[0](downwards) and KEY[1](upwards).
3. A constant rate divider to roll the screen of gameplay to the left at constant speed.

4. Displaying the score on hex lights on the board.
5. Implement the register to do statistics on game time
6. Define the FSM to determine whether the current player loses his/her life or not.
7. Complete the logic that players will lose life when plane and obstacles crash together.

What will you accomplish for the second milestone?

(similar advice as above, but for the second part of your project. Remember to specify what inputs and outputs will be used for each milestone. If your project is a visual game for example, what will appear on the screen for each milestone, for example, static colored boxes in one milestone and moving boxes in the next one etc.)

1. Implement the random number generation algorithm by adding modular operations to ALU
2. Control the probability of appearance of lava drops and item that triggers unbeatable state
   a. if the difficulty is hard (SW[1] =1) -> the rolling speed would be changing every 30s, and the lava drop would appear more frequently (33%). On the contrary, bonus item would appear more rarely (8%)
   b. if the difficulty is normal (SW[1]=0)-> the rolling speed would be changing every 60s, and the lava drop would appear at 20% probability. On the contrary, bonus item would appear at 15% probability.
3. We will make the rate divider working properly with the FSM. Once the speed_changing counter reaches its limit, the rate divider will double the speed of clock cycle of the FSM.

What will you accomplish for the third milestone?

(don't say "everything" just because this is the final milestone; describe the final components instead, and exactly what the TAs should expect to see)

1. Testing the edge cases whether the gap of tubes are reasonable
2. The 8-bit music would be played and the notes would be shined on LEDRs
3. The VGA should display the scene properly
4. The hexes would display the score in decimal (ie, just showing 0-9)
5. Show the demo to TA.

How does this project relate to material covered in CSC258?

This project covers: ALU, rate divider, counter(timer), FSM ,datapath and control block, and VGA display. These are the fundamental structures appeared in labs and lectures. On top of that, the project also involves designing concepts and ideas revealed in the lectures and labs.

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

We would recreate our popular old-time classical game in hardware level and add some fancy new features we like to it. Besides, it is a good overall summative practice of all components and designs taught in classes and labs. Hence we could understand the working theories and the ideas behind those components and designs deeper and better.

On the other hand, this project could attract non-CSC258 students to take this course.

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

I(Tianjiao He) saw a movie called Ekipazh that described an internship pilot used his excellent skill to save the passengers on the plane even though they met a volcanic eruption. The exciting story motivates me to write the flight saving game where player can help the power-lost plane to fly over mountainous regions.