**Sign-off Sheet**

***Early (Bonus) Sign-off***: **Friday 9/9/22** ***Report due***: **Wednesday 9/14/22**

**Student 1:** Cristobal Rincon Rogers

**Student 2:** Lili Loughlin

| ***Task Max points*** | | | ***Max points*** | ***TA’s assessment*** |
| --- | --- | --- | --- | --- |
|  | ***PRE-LAB Complete at start of lab*** |  | 10 | Student 1    Student 2 |
| (Students graded individually) BOTH PRE-LABS MUST BE ATTACHED |
| SPACE INVADERS on LCD | | | 5 |  |
| Game start with count down (on \* key) | | | 10 |  |
| Display descending aliens | | | 10 |  |
| Check keypad and remove dead aliens | | | 10 |  |
| Have graphical icons display “aliens” instead of alpha-numeric characters | | | ***5*** |  |
| Play complete game (i.e. display aliens, check keypad, update aliens positions, increase speed and number of aliens) | | | 10 |  |
| Proper player humiliation on error & reset to welcome screen | | | 5 |  |
| Answer to TA Questions  (up to 5 points per student and up to 10 points per team) | | | 10 | Student 1    Student 2 |
| Adding buzzer “sound track” with at least 5 pitches that plat on game events.  (Buzzer on Game Over is not enough!) | | | ***5*** |  |
| Report | | | 30 |  |
| ***Total points***  ***(with bonuses)*** | | | 100  ***110*** |  |

# Table of Contents

[**Introduction**](#_kpo5wbga09mj) **3**

[**Discussion/Questions**](#_cuzhjvd4fl8l) **4**

[**Conclusion**](#_ove2ntguj9h8) **4**

# 

# Introduction

Created in the late 1970’s, one of the earliest and most popular video arcade games was Space Invaders. The objective of Space Invaders was simple, shoot the descending aliens before they land (or they shoot you). The original game displayed rows of aliens who “fell” down the screen unless the player shot them. As soon as the player cleared all the threatening aliens, more would appear, and these ones would be descended even more quickly.

Our objective in this lab was to program and debug our custom MSP430F5529 Launchpad-based Lab Board using Code Composer Studio (CCS) 10.4. Additionally, we implemented a “Space Invaders” like game where randomly generated numbers appeared on the lab board’s onboard LCD screen. Once numbers appeared, users would have a fixed amount of time to supply input by pressing one of the numbers on the keypad. If the input was valid (values between 1 and 5) then it would remove it from the LCD screen and move the aliens closer to the “base” or bottom of the LCD screen. If the aliens made it to the “base” the user would be shown a loser screen and have to start from the beginning.

# 

# Discussion/Questions

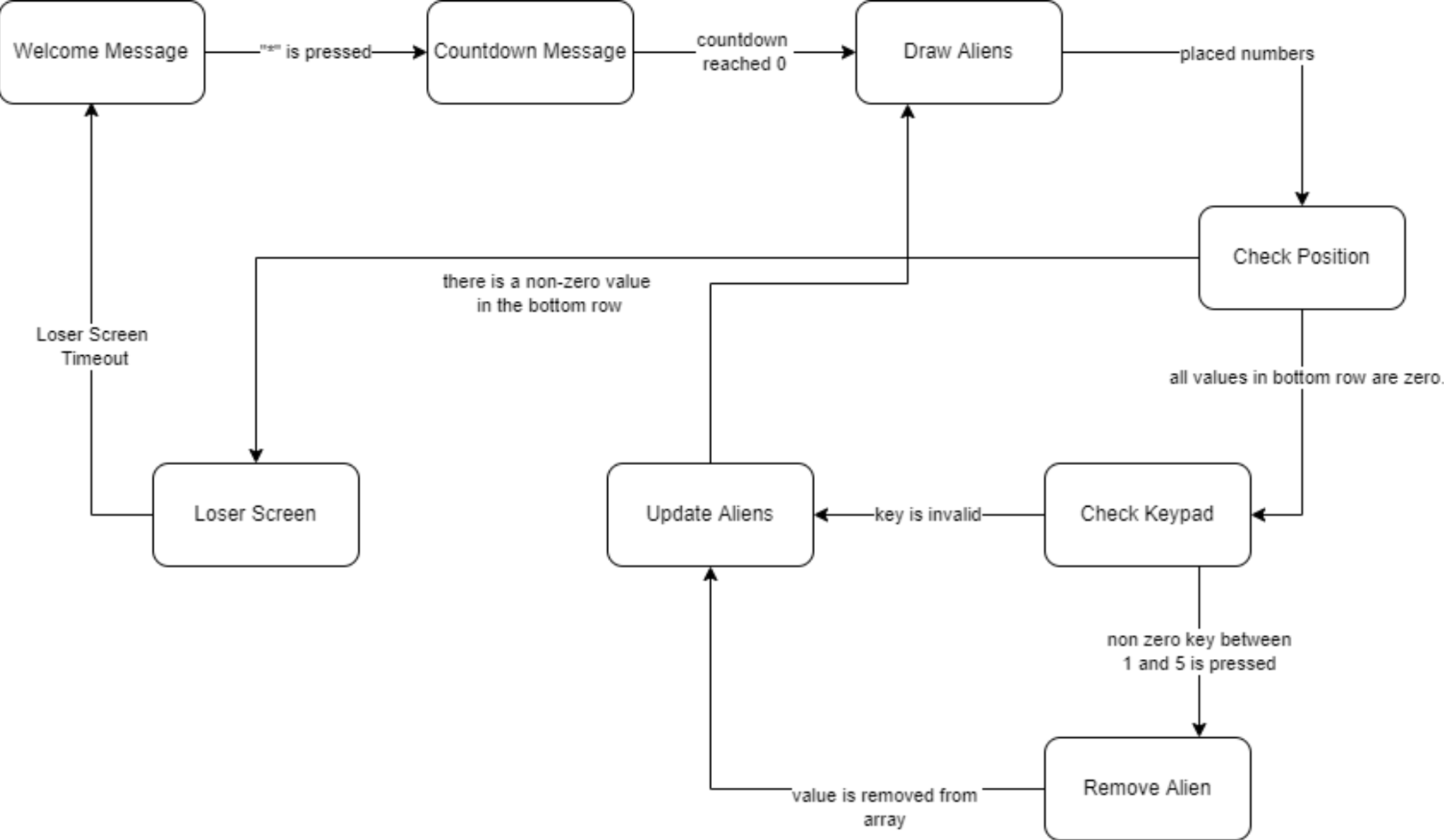


Figure 1: State Diagram for Space Invaders like Program.

Our state machine used seven states in total: *welcome*, *countdown*, *drawAliens*, *checkPos*, *checkKey*, *updateAliens*, *loser*. In the *welcome* state, the string “SPACE INVADERS Press \* to START” is displayed on the screen. If the \* button is pressed, the current state is updated to *countdown*, which utilizes the void function countdownScreen(); to display the values 3, 2, then 1 before switching to the *drawAliens* state. This state is split into two sections: generating numbers and placing said numbers on the screen. Because the numbers should be unpredictable, we use a for loop to assign the column values in the first row of the aliens array randomly. To actually place the aliens, we use a nested for loop to assign X, Y pixel coordinates to all non-zero values contained in the aliens array to output to the LCD screen. The *checkPos* state is needed to check if the bottom row contains any non-zero integers; if this is the case, the user is directed to the loser screen. If not, the state switches to *checkKey*. This checks if the correct corresponding key has been pressed. Then, in the *updateAliens* state, using a nested for loop, the rows are shifted down one and a new random set of values appears at the top row.

# Conclusion

In conclusion, in this lab we implemented a “Space Invaders” like game where randomly generated numbers appeared on the lab board’s onboard LCD screen. Once numbers appear, users press one of the numbers on the keypad. If the input was valid (values between 1 and 5) then it would remove it from the LCD screen and move the aliens closer to the “base” or bottom of the LCD screen. If the aliens made it to the “base” the user would be shown a loser screen and have to start from the beginning.

# 

# Appendix

# 

Figure 2: Table used to convert from decimal numbers to ASCII characters.