Construct an Advanced Computer Program for a Specified Task

by

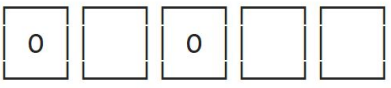
Travis Byrman

Program Test Plan

Testing of User Inputs

Input to Test	*Test Data to Use*	*Expected Outcome*
Coin selection, Grid size, Number of coins.	“2”	Input Accepted - normal data
	“ 2”	Input Rejected - abnormal data
	“[blank]”	Input Rejected - abnormal data
	“[spaces]”	Input Rejected - abnormal data
	“[tt1]”	Input Rejected - abnormal data
Coin movement onto another coin	“2”	Input Rejected - prints “Invalid move, try again”
	“[blank]”	Input Rejected - abnormal data
	“[spaces]”	Input Rejected - abnormal data
	“[tt1]”	Input Rejected - abnormal data
Player name	“Travis”	Input Accepted - normal data
	“[blank]”	Input Rejected - abnormal data
	“[spaces]”	Input Rejected - abnormal data
	“Ben12***&”	Input Accepted - normal data
Answering Yes/No question	“Yes”	Input Accepted - normal data
	“No”	Input Accepted - normal data
	“yEs”	Input Accepted - normal data
	“12dwada”	Input Rejected - abnormal data

Testing of Gameplay

Feature to Test	*Test Data to Use*	*Expected Outcome*
Coin movement - displays the coin in the correct spot as requested by user, does not jump coins or replaces coins.	'2' '10' '1' '4'	If move is successful then [next players turn] otherwise re-print the input statement and print output why.
Coin removing - coin is removed when requested by user after certain conditions are met.	'1'	'[player] removed a coin.' [coin is removed]
Player names printed - prints when its the correct players turn.	'Ben' 'Travis'	'Ben, what coin would you like to select to be moved?' 'Travis, what coin would you like to select to be moved?'
Winning output statement - prints output of winner using correct player name.	'Ben wins' 'Travis wins'	'Congratulations, Ben is the winner!' 'Congratulations, Travis is the winner!' 'Congratulations, Fred is the winner!'
Number line - display correct amount of numbers that correspond with the boxes.	Grid Size: '10', '15'	'1 2 3 4 5 6 7 8 9 10' '1 2 3 4 5 6 7 8 9 10 11 12 13 14 15'
Boxes - displays boxes around all items, does not display extra boxes or not enough.	'10'	

Invalid input feedback - prints output telling user that invalid data has been entered.	‘Dawda1131’	‘Invalid input, try again.’
Coin automatic move - automatically moves a coin if there is only one available space.	‘11’	[coin moved to the only available spot to the left, which is next to the coin]
Displays the coin selected - arrows point to coin selected after user has chosen a coin to move.	‘1’	‘>O<’
Default settings - offers the use default settings that they can use or change if they would like.	‘Yes’	[creates game using default settings]
Game restart - user can play the game again or quit.	‘Yes’	[game restarts depending on user input]

Evidence of Testing

Testing of User Inputs

Coin selection

Input: “2”	ben, what coin would you like to select to be moved? 10 How many places to the left would you like the coin to be moved? 2	**PASS**
Input: “ 2”	travis, what coin would you like to select to be moved? 8 How many places to the left would you like the coin to be moved? 2	**PASS**
Input: “[blank]”	ben, what coin would you like to select to be moved? Enter a value between 1 and 30	**PASS**
Input: “[spaces]”	ben, what coin would you like to select to be moved? Enter a value between 1 and 30	**PASS**
Input: “[tt1]”	ben, what coin would you like to select to be moved? tt1 Enter a value between 1 and 30	**PASS**

Coin movement onto another coin

Input: “2”	How many places to the left would you like the coin to be moved? 5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Invalid turn, try again Ben, what coin would you like to select to be moved?	**PASS**
Input: “[blank]”	Travis, what coin would you like to select to be moved? 3 How many places to the left would you like the coin to be moved? Enter a value between 1 and 2	**PASS**
Input: “[spaces]”	Travis, what coin would you like to select to be moved? 3 How many places to the left would you like the coin to be moved? Enter a value between 1 and 2	**PASS**
Input: “[tt1]”	How many places to the left would you like the coin to be moved? tt1 Enter a value between 1 and 2	**PASS**

Player names

“Travis”	Player 1, what is your name? Travis Player 2, what is your name?	**PASS**

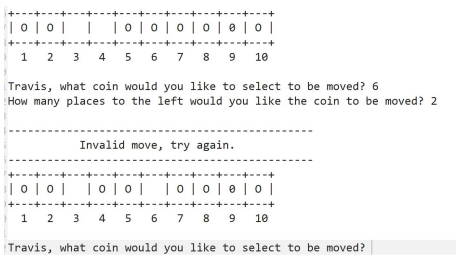
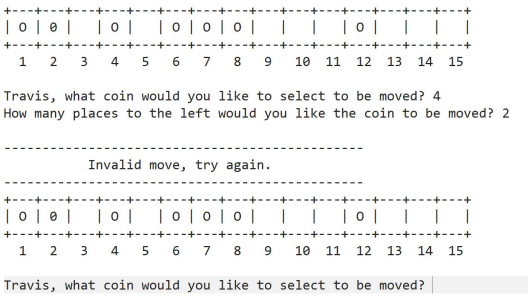
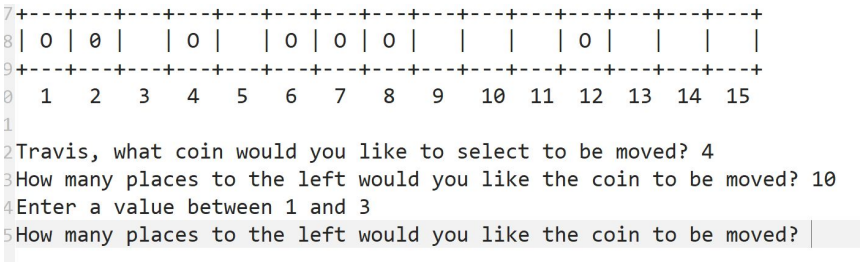
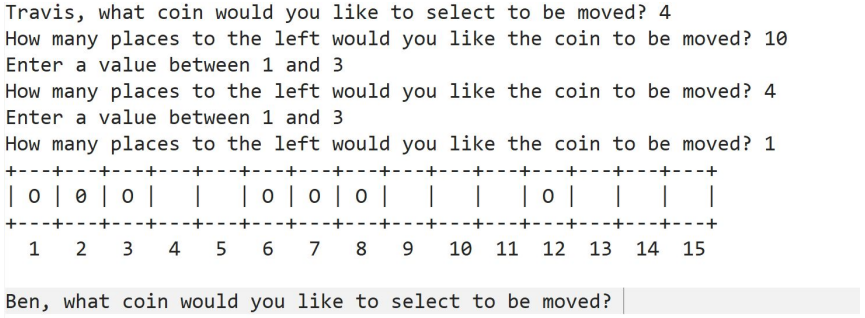
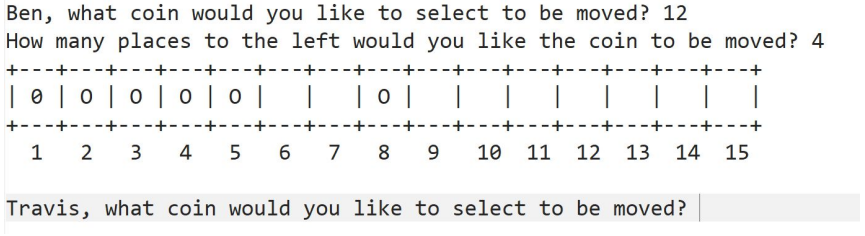
“[blank]”	Player 1, what is your name? Travis Player 2, what is your name? Player 2, what is your name?	**PASS**
“[spaces]”	Player 1, what is your name? Invalid input, please enter some text	**PASS**
“Ben12***& dadwa”	Player 1, what is your name? Ben12***& dadwa Player 2, what is your name?	**PASS**

Answering Yes/No Question

“Yes”	—————————————————————————— Would you like the instructions? (Yes/No) Yes —————————————————————————— How to play!	**PASS**
“No”	—————————————————————————— Would you like the instructions? (Yes/No) No —————————————————————————— Enter your names to continue! —————————————————————————— Player 1, what is your name?	**PASS**
“yEs”	—————————————————————————— Would you like the instructions? (Yes/No) yEs —————————————————————————— How to play!	**PASS**
“123abc”	Would you like the instructions? (Yes/No) 123abc Invalid input, please enter 'Yes' or 'No' Would you like the instructions? (Yes/No)	**PASS**

Testing of Gameplay

Coin movement

'2'		**FAIL**
		**PASS**
'10'		**PASS**
'1'		**PASS**
'4'		

Coin removing

'1' (for gold coin)	Travis, what coin would you like to select to be moved? 1 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+ | O | O | O | O | | | O | | | | | | | | +---+---+---+---+---+---+---+---+---+---+---+---+---+---+ --- Congratulations! Travis is the winner! ---	**PASS**
'1' (for silver coin)	+---+---+---+---+---+---+---+---+---+---+ | O | O | | O | O | O | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 ben, what coin would you like to select to be moved? 1 --- ben removed a silver coin! --- +---+---+---+---+---+---+---+---+---+---+ | | O | | O | O | O | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 travis, what coin would you like to select to be moved?	**PASS**

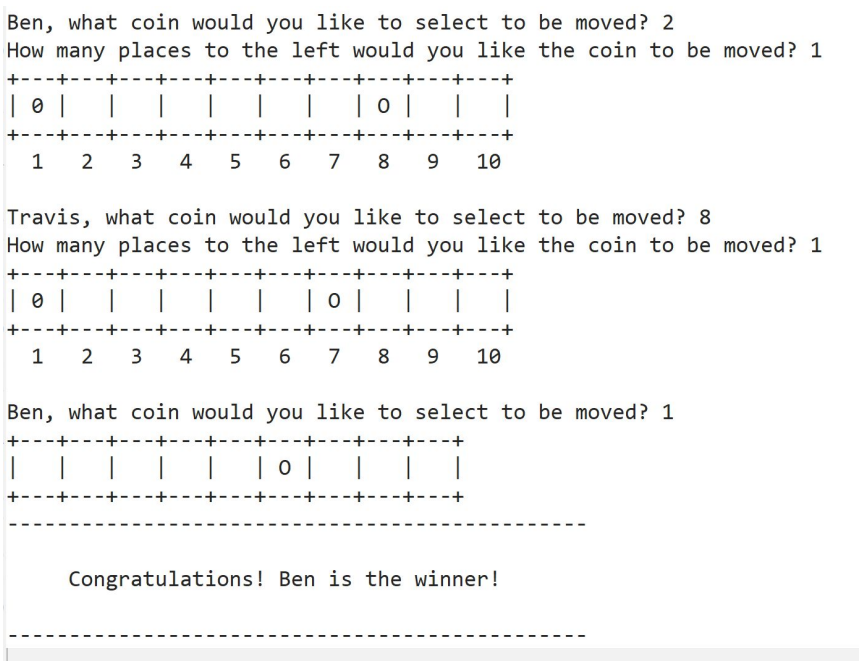
Player names printed

Names 'ben' and 'travis' alternating each turn	Travis, what coin would you like to select to be moved? 2 How many places to the left would you like the coin to be moved? 1 +---+ | O | | | | | | O | | O | | | 0 | | | | | | | O | +---+ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Travis, what coin would you like to select to be moved? 8 How many places to the left would you like the coin to be moved? 1 +---+ | O | | | | | O | | | O | | | 0 | | | | | | | O | +---+ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Travis, what coin would you like to select to be moved? 10 How many places to the left would you like the coin to be moved? 1 +---+ | O | | | | | O | | O | | | | 0 | | | | | | | O | +---+ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	**FAIL**

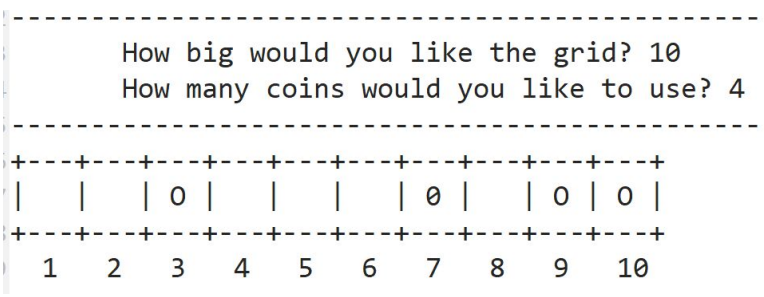
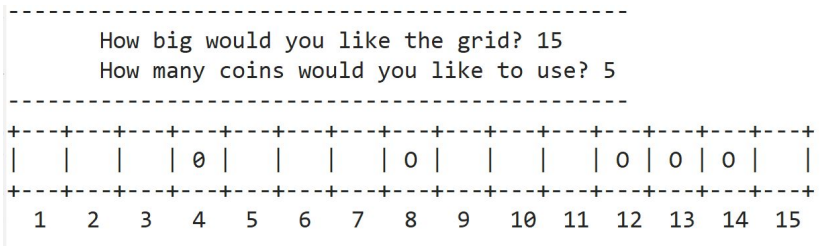
	travis, what coin would you like to select to be moved? 2 How many places to the left would you like the coin to be moved? 1 +---+---+---+---+---+---+---+---+---+---+ | O | | | O | O | O | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 ben, what coin would you like to select to be moved? 4 How many places to the left would you like the coin to be moved? 1 +---+---+---+---+---+---+---+---+---+---+ | O | | O | | O | O | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 travis, what coin would you like to select to be moved? 5 How many places to the left would you like the coin to be moved? 1 +---+---+---+---+---+---+---+---+---+---+ | O | | O | O | | O | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 ben, what coin would you like to select to be moved? 6 How many places to the left would you like the coin to be moved? 1 +---+---+---+---+---+---+---+---+---+---+ | O | | O | O | O | | O | 0 | O | O | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 travis, what coin would you like to select to be moved?	**PASS**

Winning output statement

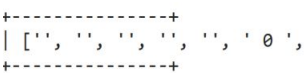
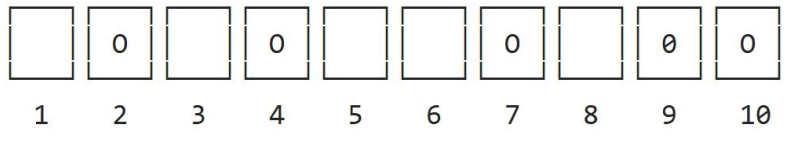
Travis wins	+---+---+---+---+---+---+---+---+---+---+ | | | | | | | | 0 | | O | | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 Ben, what coin would you like to select to be moved? 7 How many places to the left would you like the coin to be moved? 6 +---+---+---+---+---+---+---+---+---+---+ | 0 | | | | | | | | O | | +---+---+---+---+---+---+---+---+---+---+ 1 2 3 4 5 6 7 8 9 10 Travis, what coin would you like to select to be moved? 1 +---+---+---+---+---+---+---+---+---+---+ | | | | | | | | O | | +---+---+---+---+---+---+---+---+---+---+ ---------- Congratulations! Travis is the winner! ----------	**PASS**

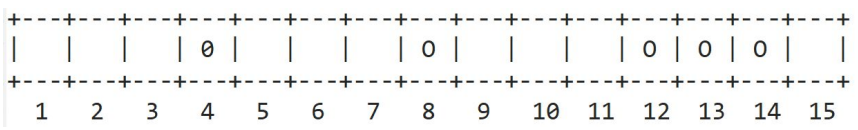
Ben Wins		**PASS**

Number line

Grid Size ‘10’		**PASS**
Grid Size ‘15’		**PASS**

Boxes

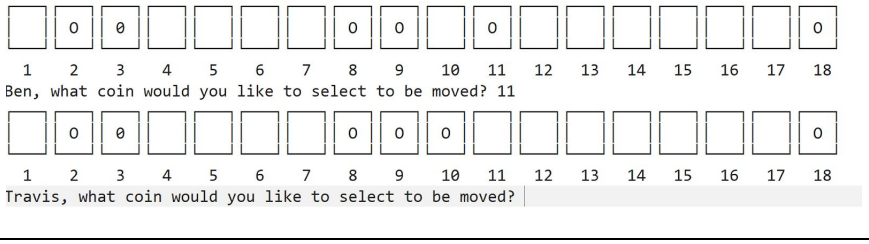
Grid Size ‘10’		**FAIL**
		**PASS**

Grid Size ‘15’		**PASS**

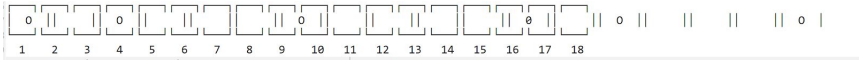
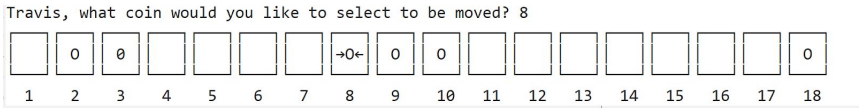
Invalid input feedback

When input is not an integer when required	Travis, what coin would you like to select to be moved? adwadaw Enter a value between 1 and 30	**PASS**

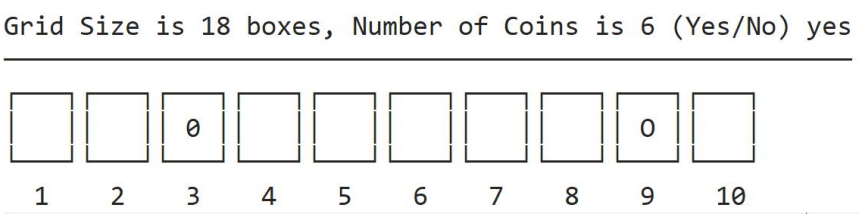
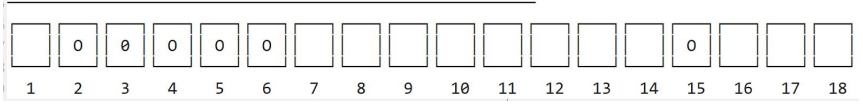
Coin automatic move

‘11’		**PASS**

Displays coin selected

‘8’		**FAIL**
		**PASS**

Default settings

‘Yes’		**FAIL**
		**PASS**

Game restart

'Yes'	Would you like to play again? (Yes/No) yes _____ Welcome to Old Gold! _____ Would you like the instructions? (Yes/No)	**PASS**

Test Log

Date	*Testing Performed, Outcomes, Bugs Found, Etc.*
27/8 -> 1/9	Introduction - inputs for names, welcome text, input for grid and number of coins, creating lists Easy process, very simple as I have done this before and understand what to do. `from Input import *` `from Output import *` `GRID_MIN = 10` `GRID_MAX = 30` `COIN_MIN = 2` `COIN_MAX = 30` `print("-------------------------------------------------\n")` `print("             Welcome to Old Gold\n")` `print("\n-------------------------------------------------")` `print("        Enter your names to continue!")` `print("-------------------------------------------------")` `P1 = inputText("        Player 1, what is your name? ")` `P2 = inputText("        Player 2, what is your name? ")` `print("-------------------------------------------------")` `GRID_SIZE = inputNumber("       How big would you like the grid? ", GRID_MIN, GRID_MAX)` `NUM_COINS = inputNumber("       How many coins would you like to use? ", COIN_MIN, GRID_SIZE)` `print("-------------------------------------------------")`
27/8 -> 29/8	Grid list creation - adding coins, randomising, adding spaces. I figured out how I should go about adding coins to the list and what I should do to have the spaces added in between. The randomising was very easy as I could use the random import.

	`` GRID = [" "] * GRID_SIZE GRID.append("0") COINS = ["O"] * NUM_COINS del COINS[:1] del GRID[:NUM_COINS] GRID.extend(COINS) random.shuffle(GRID,random.random) ```
29/8 -> 30/8	Creating boxes/styling around each item in grid. I found this to be a very difficult process. If I were to create this function outside of the file, it would have to use variables that python can recognize. The most challenging bit for me was creating this in an efficient amount of code. I managed to create a function that does this, but it just has so many lines. A bug I had was that I struggled to only show the items and have one box around each item. My original code was trying to make a box around everything and was showing the 'raw' style of the a list. ``` def BOX(List): #Boxes around items for i in range(len(List)): print("┌──┐", end = "") if(i == len(List)-1): print("", end = "\n") for i in range(len(List)): print("│{}│".format(List[i]), end = "") if(i == len(List)-1): print("", end = "\n") for i in range(len(List)): print("└──┘" , end = "") if(i == len(List)-1): print("", end = "\n") ```
6/9 -> 10/9	Selecting and moving a coin This step was very difficult for me as well. I repeated bug I was having was that the elif statement will begin when the first if statement is on the last iteration of searching the indexes. What would happen is that the coin would jump coins then the elif statement would work. To fix this I made it so that the elif statement begins as soon as the first if statement begins trying to match the conditions with the first index. ``` COIN_MOVE = inputNumber("How many places to the left would you like the coin to be moved? ", 1, COIN_SELECT - 1) PC_COIN_MOVE = (COIN_MOVE) - 1 COIN_NEW_PLACE = (PC_COIN_SELECT) - (PC_COIN_MOVE) - 1 for x in range(COIN_NEW_PLACE, PC_COIN_SELECT): if GRID[x] != " ": INVALID() PLAY(PLAYERS,GRID) break elif x == PC_COIN_SELECT - 1: GRID[COIN_NEW_PLACE] = GRID.pop(PC_COIN_SELECT) # GRID.insert(PC_COIN_SELECT , " ") ```
6/9	Winning statement Creating this was also not to difficult. I applied basic knowledge and created if statements to detect when the gold coin is removed.

	`` if COIN_SELECT == 1 and GRID[0] == GOLD_COIN: GRID.pop(0) BOX(GRID) print("---\n") print(" Congratulations! {} is the winner!".format(PLAYERS)) print("\n---") return ```
7/9	Number line I made a function that prints/assigns a number to each box/item. After 9 numbers on the line, and extra space is added to accommodate the extra digit the numbers 10+ create. ``` def NUMBER_LINE(List): for i in range(1, len(List) + 1): if i < 10: #For the first 9 boxes print(" {} ".format(i), end = "") #they get this else: print(" {} ".format(i), end = "") #10+ boxes get this if(i == len(List) + 1 - 1): print("", end = "\n") ```
7/9	Players I had a little trouble with this. I had a bug where one person continuously goes, it doesn't switch between the first and second person. To solve this in the parameter I put the list of players and grid. This alternated between players each turn and kept it linked to the main game. First name in the list would go then the second. They would alternate names until there is a winner. ``` def PLAY(PLAYERS,GRID): BOX(GRID) NUMBER_LINE() print("\n") COIN_SELECT = inputNumber("{}, what coin would you like to select to be moved? ".format(PLAYERS), 1, GRID_MAX) PC_COIN_SELECT = (COIN_SELECT) - 1 ```
7/9	While loop/functions (tying everything together) I created a while loop that contains if statements and functions for the game to operate. It also is responsible for ending the game when the gold coin is removed. I also made the main parts functions so that it is much easier to understand how the game is operated. This loop also makes sure the first player has it turn then the next. ``` def PLAY(PLAYERS,GRID): ```

```
for item in GRID:
    if GOLD_COIN in GRID:
        PLAY(PLAYERS[0], GRID)
    else:
        break
    if GOLD_COIN in GRID:
        PLAY(PLAYERS[1], GRID)
    else:
        break
```

12/9/18

Instructions

I created instructions that can be shown if ‘Yes’ is typed. If anything else is typed nothing will happen.

```
_____________________________________________

                Welcome to Old Gold!
_____________________________________________
Would you like the instructions? (Yes/No) yes
_____________________________________________________________________________________

                                How to play!
The aim of the game is to be the first person to remove the gold coin '0'.
In order to move a coin there cannot be any silver coins 'O' to the left of it.
Coins can only be moved to the left and cannot jump over eachother. To remove a coin enter '1'
For the coin to be removed, it must be in the first square of the grid. To select
a coin to be moved, type the number that is under the coin that you want to select.
To move the coin, enter how many places you want the coin to move to the left.
_____________________________________________________________________________________
```

12/9

Unicode

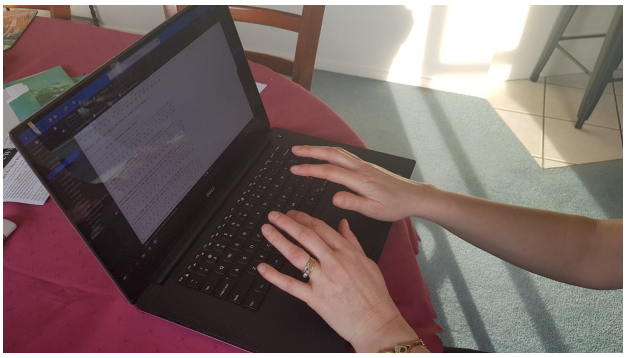
I replaced all the characters with unicode. This gives my board a cleaner look and makes the lines look much better. I also changed some line spacing and words around. I am not going to redo any previous testing as this would not affect any of my code. Although if any bugs arise from this change, I will retest parts of my code.

```
_____________________________________________

                Welcome to Old Gold!
_____________________________________________
Would you like the instructions? (Yes/No) 123abc
Invalid input, please enter 'Yes' or 'No'
Would you like the instructions? (Yes/No) no
_____________________________________________
Enter your names to continue!
_____________________________________________
Player 1, what is your name? Travis
Player 2, what is your name? Ben
_____________________________________________
Would you like to use the default settings?
Grid Size is 18 boxes, Number of Coins is 6 (Yes/No) yes
_____________________________________________

 1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18
Ben, what coin would you like to select to be moved? 8
```

13/9

Code comments

I added code comments for other people to understand my code.

	`` #Project: Old Gold Game #Author: Travis Byrman #School: Waimea College #Date: September 2018 #Module: main.py #Purpose: The game, this is how Old Gold operates ```
13/9	User testing I tested the game with a user, they enjoyed the game, clearly understood how to play the game. They also found the game very easy to operate and play. 
14/9	Coin automatic move The user now only needs to select the coin they want to move if there is a coin in the second slot to the left. The coin they select will automatically move to the only available spot on the left. Was easy to do, just used an if statement. ``` if COIN_SELECT > 1 and GRID[COIN_AUTO_MOVE] != " " or COIN_SELECT == 2 and GRID[0] == " ": GRID[COIN_DETECT] = GRID.pop(PC_COIN_SELECT) GRID.insert(PC_COIN_SELECT , " ") #insert blank into the old coin spot. BOX(GRID) #Displaying items in grid in boxes NUMBER_LINE(GRID) #Displaying number line under boxes return ```
15/9	Displays coin selected When the user selects a coin, the coin shows up with arrows pointing at it to tell the user what coin they have selected. The only problem I had with this was that I had to change the amount of spaces inserted and removed on the grid to allow enough room for the arrows.

	`` if GRID[PC_COIN_SELECT] == SILVER_COIN: GRID.pop(PC_COIN_SELECT) #Remove the silver coin GRID.insert(PC_COIN_SELECT , "→O←") #Insert coin with arrows BOX(GRID) #Print box with arrows and coin NUMBER_LINE(GRID) #Prints number line GRID.pop(PC_COIN_SELECT) #Remove the coin with arrows GRID.insert(PC_COIN_SELECT , " O ") #Reverts back to previous coin, doesnt show user coin without arrows. This is for the PC to identify the coin. ```
15/9	Default settings I added default settings the user can use or the user can customise the settings. Those settings are layout size and number coins. A bug I found was that when the game restarted, it used the settings the user chose from the last game. This was an easy fix though. ``` DEFAULT = inputTextYN("Would you like to use the default settings?\nGrid Size is 18 boxes, Number of Coins is 6 (Yes/No) ") if DEFAULT.lower() == "no": #If input is 'No' game options will be asked If input is 'Yes' default options are used. ```
15/9	Game restart I added a way to restart the game if they want to play again. This was easy to do. If the user chose not to play again a thank you statement is printed. ``` Would you like to play again? (Yes/No) no Thank you daw & daw for playing Old Gold! ```
16/9	Final Testing I tested the game with again with an intended user to see weather they like the changes I have made. The intended user said that the game was great and that they liked the automatic movement of the coins.