Head Tails Cricket Game in MIPS

HEADINGS:

Table of Contents

1. **Introduction and problem……………………………………………………………………………………………….2**
2. **Technology…………………………………………………………………………………………………..………………...3**
3. **Functionalities………………………………………………………………………………………………………………...4**
4. **Module distribution…………………………………………………………………………………………………………5**
5. **Code ……………………………………………………………………………………………………………………………….6**
6. **Interfaces ..……………………………………………………………………………………………………………………..7**
7. **Conclusion……………………………..……………………………………………………………………..……………..…8**

**1. INTRODUCTION & PROBLEM:**

This game is based upon cricket match. It consists of 2 gameplay which includes single player match and A multiplayer match. There will be a toss between rivals at every start of the match. The match will be play of a single over. The problems that rose while writing the code were to generate random numbers for toss. Another problem was proceeding according to the decision of USER or the CPU to choose batting or bowling and then adding the inputs of only the batter to the score but with research and systematic development, we were successful to build it. Next hurdle was to use functions appropriately to minimize our code as possible.

**2. TECHNOLOGY:**

We have used assembly language in order to create this game. We used conditional statements of assembly, branches, menus for selecting desired game mode, functions to make our code concise and to divide it into modules. Random toss will be originated by computer. While playing with CPU in a single player match random numbers will be generated by the computer, their score will be compared in every match to decide the winner.

In multiplayer the toss is randomized, a player will choose head or tails and random will be generated by the random generator.

**3. FUNCTIONALITIES:**

We have to choose in between heads and tails and Random choice will be generated by the computer for toss, 2 modes can be played in the game. First mode is single player and this game will be played between CPU and user in which computer will generate number randomly and innings will end when the number of user and CPU becomes same.

Second mode is multiplayer match which can be played between two players. It also consists of a wicket. The innings get over as the number of both players becomes same.

The score will be compared at the end of the game, to determine the winner of the game. The player with most runs will eventually win the game.

**4.MODULE DISTRIBUTION:**

Hassan Khan: Single player program mode.

Fawad Ali: Multiplayer program mode.

Ibrahim Ahmed: Toss and Testing.

**5. CODE:**

.data

menup: .asciiz "---Welcome to Hand Cricket Game---\nRules: Each player has six balls and one wicket to score as many runs as he can.\n\nChoose a mode of game?\n[1]Single player\n[2]Multiplayer\n"

toss: .asciiz "---TOSS---\n Player 1 choose a number\n[1]Heads\n[2]Tails\n"

p1win: .asciiz "\nPlayer 1 has won the toss.\nPlayer 1 shall choose to bat or bowl\n[1]Bat\n[2]Bowl\n"

p1lose: .asciiz "\nCPU has won the toss.\n"

mp1lose: .asciiz "\nPlayer 2 has won the toss.\nPlayer 2 shall choose to bat or bowl\n[1]Bat\n[2]Bowl\n\n"

p1bat: .asciiz "\nPlayer 1 is the batter\n"

p1bowl:.asciiz "\nPlayer 1 is the bolwer\n"

start: .asciiz "\nThe game begins!!\n"

wicket: .asciiz "\nWicket!!\n"

score: .asciiz "\nInning score: \n"

nextball: .asciiz "\nEnter Number any number 1-6\n"

userinput: .asciiz "User input\n"

cpuinput: .asciiz "CPU input\n"

changeinn: .asciiz "\n\n---2nd INNING---\nThe batter is now the bolwer\n"

Userwon: .asciiz "\nUser has won the match\n"

CPUwon: .asciiz "\nCPU has won the match\n"

scorecard: .asciiz "\nScorecard:\n"

Userscore: .asciiz "User Score: "

CPUscore: .asciiz " CPU Score: "

NotChasedp: .asciiz "\nThe target could not be chased hence the 1st inning batter won\n"

Chasedp: .asciiz "\nThe target was chased hence the 2nd inning batter won\n"

draw: .asciiz "\nThe match has ended in a Draw\n"

playagain: .asciiz "\nDo you want to play again?\n[1]Yes\n[2]No\n"

user1: .asciiz "The batter's input\n"

user2: .asciiz "The bolwer's input\n"

score1: .asciiz "1st inning score: "

score2: .asciiz " 2nd inning score: "

Firstinningwin: .asciiz "\nThe target could not be chased hence the 1st inning batter won\n"

Secondinningwin: .asciiz "\nThe target was chased hence the 2nd inning batter won\n"

.text

.globl main

main:

b menu

menu:

la $a0,menup #Heads or tails prompt

li $v0,4

syscall

li $v0,5 #Heads or tails input

syscall

beq $v0,1,Singleplayer

beq $v0,2,Multiplayer

b menu

Singleplayer:

b tossm

tossm:

li $t2,0

li $t7,0

li $t3,0

la $a0,toss #Heads or tails prompt

li $v0,4

syscall

li $v0,5 #Heads or tails input

syscall

move $t5,$v0

addi $a1,$zero,2 # random number for toss

addi $v0,$zero,42

syscall

addi $a0,$a0,1

beq $a0,$t5,player1win #for same result with heads or tails

bne $a0,$t5,player1lose #for different result

player1win:

la $a0,p1win #player 1 wins prompt

li $v0,4

syscall

li $v0,5 #player 1 inputs for bat or bowl

syscall

move $t6,$v0

beq $t6,1,scenerio1 #scenerio 1 p1 bats and p2 bowls

beq $t6,2,scenerio2 #scenerio 2 p1 bowls and p2 bats

scenerio1:

la $a0,p1bat #Defines clearly the batter and bowler

li $v0,4

syscall

li $t9,0

b loopbat

scenerio2:

la $a0,p1bowl #Defines clearly the batter and bowler

li $v0,4

syscall

li $t9,1

b loopbowl

player1lose:

la $a0,p1lose #player 2 wins prompt

li $v0,4

syscall

addi $a1,$zero,2 # random number for toss

addi $v0,$zero,42

syscall

addi $a0,$a0,1

move $t6,$a0

beq $t6,1,scenerio2 #scenerio 2 p1 bowls and p2 bats

beq $t6,2,scenerio1 #scenerio 1 p1 bats and p2 bowls

loopbat:

bgt $t3,5,batScore

addi $t3,$t3,1

la $a0,nextball

li $v0,4

syscall

la $a0,userinput

li $v0,4

syscall

li $v0,5

syscall

move $t0,$v0

la $a0,cpuinput

li $v0,4

syscall

addi $a1,$zero,6 # random number for toss

addi $v0,$zero,42

syscall

addi $a0,$a0,1

li $v0,1

syscall

move $t1,$a0

beq $t1,$t0,outloopbat

add $t8,$t8,$t0

b loopbat

outloopbat:

addi $t2,$t2,1 #wickets

la $a0,wicket

li $v0,4

syscall

ble $t2,0,loopbat #if there are wickets left branch back to "loop"

b batScore

outloopbowl:

addi $t2,$t2,1

la $a0,wicket

li $v0,4

syscall

ble $t2,0,loopbowl #if there are wickets left branch back to "loop"

b bowlScore

loopbowl:

bgt $t3,5,bowlScore # for branching on

addi $t3,$t3,1

la $a0,nextball

li $v0,4

syscall

la $a0,userinput

li $v0,4

syscall

li $v0,5

syscall

move $t0,$v0

la $a0,cpuinput

li $v0,4

syscall

addi $a1,$zero,6 # random number for toss

addi $v0,$zero,42

syscall

addi $a0,$a0,1

li $v0,1

syscall

move $t1,$a0

beq $t1,$t0,outloopbowl

add $t4,$t4,$t1

b loopbowl

batScore: #display Score

addi $t7,$t7,1 #innings

la $a0,score

li $v0,4

syscall

la $a0,($t8)

li $v0,1

syscall

b changeinning

bowlScore: #display Score

addi $t7,$t7,2

la $a0,score

li $v0,4

syscall

la $a0,($t4)

li $v0,1

syscall

b changeinning

changeinning:

li $t3,0 #reset balls

bge $t7,3,finalscore

la $a0,changeinn

li $v0,4

syscall

beq $t7,1,loopbowl

beq $t7,2,loopbat

finalscore:

la $a0,scorecard

li $v0,4

syscall

la $a0,Userscore

li $v0,4

syscall

move $a0,$t8

li $v0,1

syscall

la $a0,CPUscore

li $v0,4

syscall

move $a0,$t4

li $v0,1

syscall

blt $t4,$t8,userwon

bgt $t4,$t8,cpuwon

beq $t4,$t8,drawb

userwon:

la $a0,Userwon

li $v0,4

syscall

b exit

cpuwon:

la $a0,CPUwon

li $v0,4

syscall

b exit

drawb: #draw match

la $a0,draw

li $v0,4

syscall

b exit

exit: #termination branch

la $a0,playagain

li $v0,4

syscall

li $v0,5

syscall

beq $v0,1,menu

li $v0,10

syscall

Multiplayer:

b mtossm #branching to tossm

mtossm:

li $t2,0 #variable for wickets

li $t7,0 #variable for two innings

li $t3,0 #variable for over

la $a0,toss #Heads or tails prompt

li $v0,4

syscall

li $v0,5 #Heads or tails input

syscall

move $t5,$v0

addi $a1,$zero,3 # random number for toss

addi $v0,$zero,42

syscall

beq $a0,$t5,mplayer1win #for same result with heads or tails

bne $a0,$t5,mplayer1lose #for different result

mplayer1win:

la $a0,p1win #player 1 wins prompt

li $v0,4

syscall

li $v0,5 #player 1 inputs for bat or bowl

syscall

move $t6,$v0

beq $t6,1,mscenerio1 #scenerio 1 p1 bats and p2 bowls

beq $t6,2,mscenerio2 #scenerio 2 p1 bowls and p2 bats

mplayer1lose:

la $a0,mp1lose #player 2 wins prompt

li $v0,4

syscall

li $v0,5 #player 2 inputs for bat or bowl

syscall

move $t6,$v0

beq $t6,1,mscenerio2 #scenerio 2 p1 bowls and p2 bats

beq $t6,2,mscenerio1 #scenerio 1 p1 bats and p2 bowls

mscenerio1:

la $a0,p1bat #Defines clearly the batter and bowler

li $v0,4

syscall

b mloop #branch to loop

mscenerio2:

la $a0,p1bowl #Defines clearly the batter and bowler

li $v0,4

syscall

b mloop #branch to loop

mloop:

bgt $t3,5,mScore

addi $t3,$t3,1

la $a0,nextball

li $v0,4

syscall

b muser1in #branch to user1 input

muser1in:

la $a0,user1

li $v0,4

syscall

li $v0,5

syscall

move $t0,$v0

b muser2in #branch to user2 input

muser2in:

la $a0,user2

li $v0,4

syscall

li $v0,5

syscall

move $t1,$v0

beq $t0,$t1,mout #check for same input which means wicket. If true branch to "out"

syscall

add $t4,$t4,$t0 #add score

b mloop #branch back to loop if no wicket

mout:

addi $t2,$t2,1 #increment wicket

la $a0,wicket

li $v0,4

syscall

ble $t2,0,mloop #if there are wickets left branch back to "loop"

b mScore

mScore: #display Score

li $t3,0

la $a0,score

li $v0,4

syscall

la $a0,($t4)

li $v0,1

syscall

beq $t7,1,mFinalScore

addi $t7,$t7,1

move $t8,$t4

li $t4,0

la $a0,changeinn

li $v0,4

syscall

b mloop

mFinalScore: #display Score

la $a0,scorecard

li $v0,4

syscall

la $a0,score1

li $v0,4

syscall

move $a0,$t8

li $v0,1

syscall

la $a0,score2

li $v0,4

syscall

move $a0,$t4

li $v0,1

syscall

bgt $t8,$t4,mFirstinningwinb

blt $t8,$t4,mSecondinningwinb

beq $t8,$t4,mdrawb

mFirstinningwinb:

la $a0,Firstinningwin

li $v0,4

syscall

b exit

mSecondinningwinb:

la $a0,Secondinningwin

li $v0,4

syscall

b exit

mdrawb: #draw match

la $a0,draw

li $v0,4

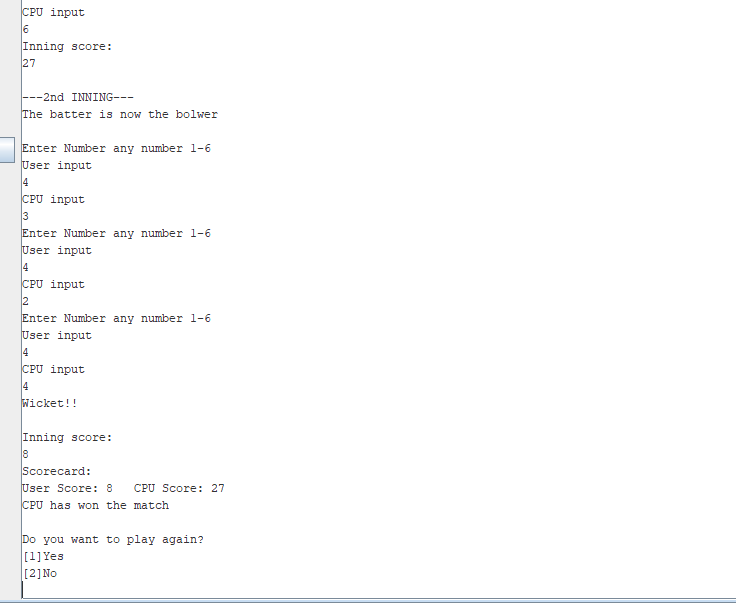
syscall

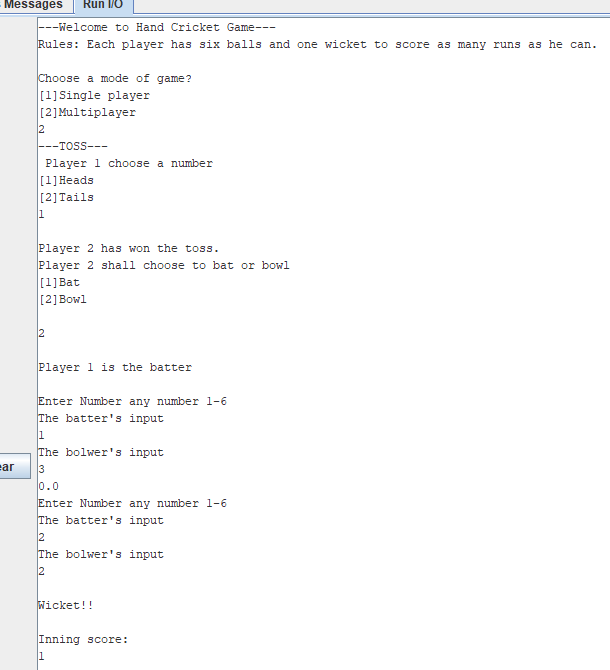
b exit

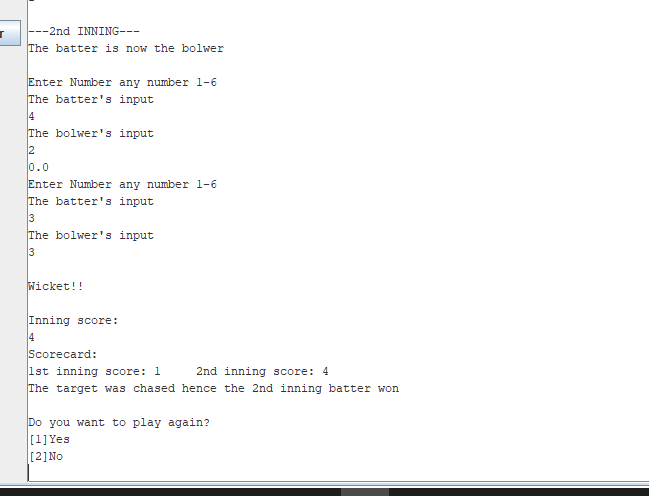
**6. INTERFACES:**

**Single player mode:**

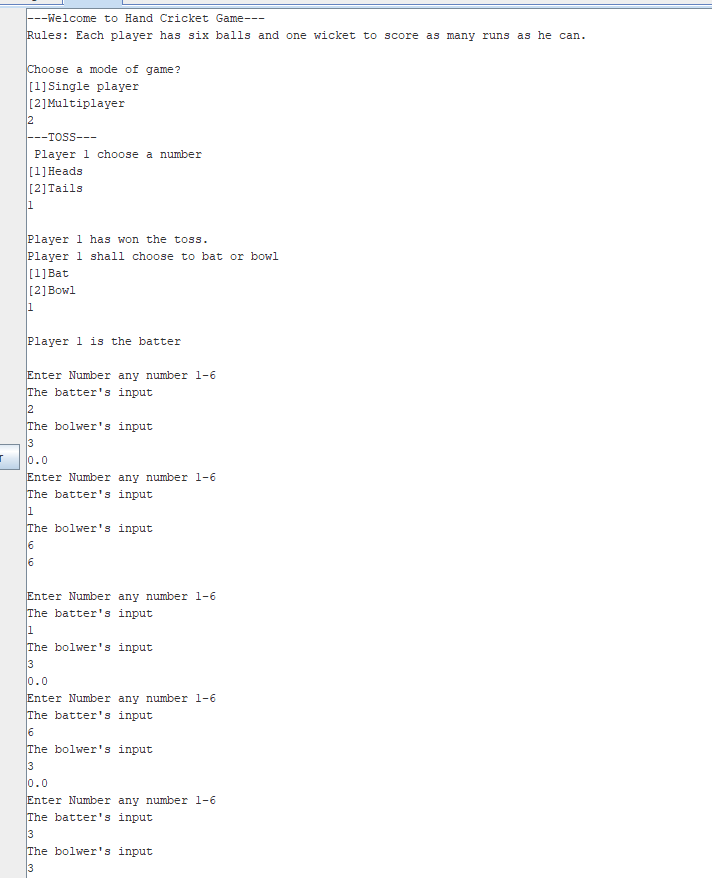


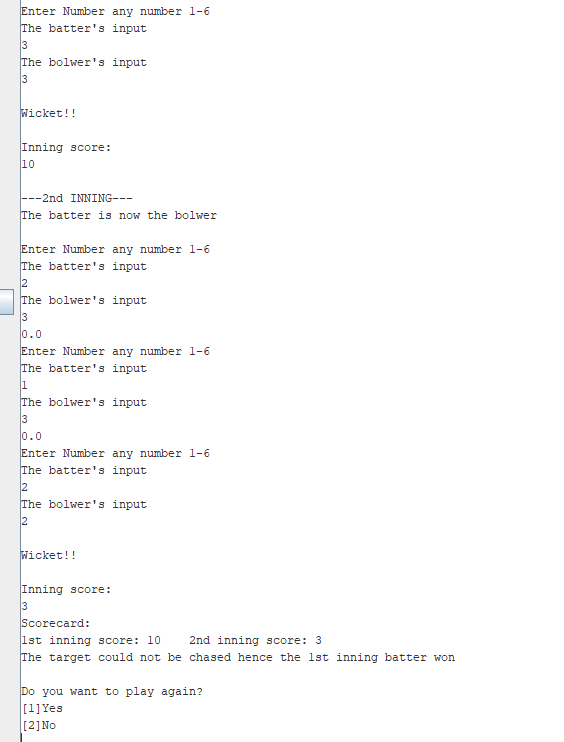


**Multiplayer mode #1:**  




**Multiplayer mode #2:**





**7.CONCLUSION:**

Overall, it was a great experience to learn from thinking ideas till finalising the code for game. It improved our skill and gave us immense knowledge to tackle any situation like ahead. Though it was difficult and problematic way, but the great thing was to gain knowledge which we successfully did and were successful to launch in given time period. We learn to execute the given instruction at right and practical format which were taught during classes. In general, it gives us blink of an idea for how to work in future.