Dec. 3rd 2103

GameStats.java

jTextField12: “LAP”

jTextField13: Lap of left car.

jTextField14: “LAP”

jTextField15: Lap of right car.

Single.java

jTextField14: “LAP”

jTextField15: Lap of left car.

Start.java (optional)

boolean trigger: signal from XBOX controller to select button.

int selectMode: 1 is when only left three buttons are shown and 2 is when right four buttons are shown.

boolean finish: initialed false. Changed to true when button 4,5,6 pressed.

Nov. 30th 2013

I was in hospital since Wednesday and will still be there for several days. Hope I can go to the class on Monday. Here are the descriptions of three frames I left last time. Start.java is very simple, only receiving game settings and pass to CS, and then status frame comes out.

For Result.java nad Record.java, I think if we connect GameStats.java and Single.java with CS and receive result of the game from there, the only thing need to connected are “RePlay” and “Exit” buttons.

I didn’t do comments on these three frames. 3G signal is so weak in here. I can’t even sync.

Start.java

jButton1: Practise button. Click jButton1, jButton2 will be hidden and jButton 3,4,5,6,7 will be shown.

jButton2: Race button. Click jButton2, jButton1 will be hidden and jButton 3,4,5,6,7 will be shown.

jButton3: Back button. Initialed hidden. Click jButton3, go back to initial condition, jButton1 and jButton2 will be shown, other buttons will be hidden.

jButton4,5,6: Different difficulty levels buttons. When clicked, pass difficulty level to next frame.

jButton7: Exit button. Click jButton7 will exit this program.

Result.java

jTextFields1: “HAZZARD WINNER”

jTextFields2: result of the game passed from GameStats.java. Can be “Player ONE” which is left car, “Player TWO” which is right car, or “TIE”.

jTextFields3: “Player ONE”

jTextFields4: “Player TWO”

jTextFields6: millisecond of Left car result

jTextFields7: minute of Left car result.

jTextFields8: second of Left car result

jTextFields9: millisecond of Right car result.

jTextFields10: minute of Right car result

jTextFields11: second of Right car result.

Three digits millisecond result will show the exactly result of each player.

jButton1: RePlay button. Go back to Start.java.

jButton2: Exit button. Click jButton2 will exit this program.

Record.java

jTextFields1: “HAZZARD”

jTextFields2: “BEST RECORD”

jTextFields5: minute of Left car best record.

jTextFields6: second of Left car best record.

jTextFields7: millisecond of Left car best record.

Three digits millisecond record will show the exactly record of best trial.

jButton1: RePlay button. Go back to Start.java.

jButton2: Exit button. Click jButton2 will exit this program.

Nov. 27th 2013

The order of jTextFields may cause lots of confusion. I am sorry.

I mentioned CS in this file of where I think should send signals and commented $$$send in code.

GamaStats.java

Two cars. Left and Right.

jTextFields1: minute of Main clock.

jTextFields2: millisecond of Main clock.

jTextFields3: second of Main clock.

jTextFields4: speed of left car.

jTextFields5: speed of right car.

jTextFields6: millisecond of Left car sub-clock.

jTextFields7: minute of Left car sub-clock.

jTextFields8: second of Left car sub-clock.

jTextFields9: millisecond of Right car sub-clock.

jTextFields10: minute of Right car sub-clock.

jTextFields11: second of Right car sub-clock.

jLabel5: Left turn of Left car.

jLabel6: Right turn of Left car.

jLabel7: Straight of Left car.

jLabel8: Straight of Right car.

jLabel9: Left turn of Right car.

jLabel10: Right turn of Right car.

jLabel15: Difficulty level of the game.

jButton1: Ready button of Left car.

jButton2: Ready button of Right car.

jButton3: Next button.

boolean variables rFinish and lFinish are initialed false and will be true when car reaches the finish line.

integer arrive counts how many car reaches the finish line, when arrive == 2, next button will be shown.

boolean variables rLeft, rStraight, rRight, lLeft, lStraight, lRight are direction detectors of both cars. They are initialed to both Straight and only can be changed when game starts. Detected in ActionListener.

integer min, sec, ms are main clock. Initialed all 0. Logic wrote at the beginning of the constructor.

String mode receive the difficulty level from Start.java and print out.

boolean rReady, lReady, start are initialed all false. When rReady and lReay are changed to true, Main clock will show lights from red to yellow to green. When green lights are on, start will be changed to true and a start signal should be sent to CS. Game starts.

integer countDown. A variable of how long lights changed after Ready buttons are pushed.

integer lMin, lSec, lMs, rMin, rSec, rMs are initialed to 99 or 999. They will store the result of the race and should be sent to Result.java and CS to record.

double vl, vr. Speed of two cars, initialed to 0, be refreshed in jTextField4 and jTextField5 by ActionListener. Should receive changes from CS in real time.

Single.java

One car. Only Left.

jTextFields1: minute of Main clock.

jTextFields2: millisecond of Main clock.

jTextFields3: second of Main clock.

jTextFields4: speed of left car.

jTextFields5: minute of Left car sub-clock.

jTextFields6: millisecond of Left car sub-clock.

jTextFields7: second of Left car sub-clock.

jLabel5: Left turn of Left car.

jLabel6: Right turn of Left car.

jLabel7: Straight of Left car.

jLabel15: Difficulty level of the game.

jButton1: Ready button of Left car.

jButton2: Try again button.

jButton3: Result button.

boolean variables rFinish is initialed false and will be true when car reaches the finish line.

boolean variables lLeft, lStraight, lRight are direction detectors of left car. They are initialed to Straight and only can be changed when game starts. Detected in ActionListener.

integer min, sec, ms are main clock. Initialed all 0. Logic wrote at the beginning of the constructor.

String mode receive the difficulty level from Start.java and print out.

boolean lReady, start are initialed all false. When lReay is changed to true, Main clock will show lights from red to yellow to green. When green lights are on, start will be changed to true and a start signal should be sent to CS. Game starts.

integer countDown. A variable of how long lights changed after Ready buttons are pushed.

integer minRecord, secRecord, msRecord are initialed to 99 or 999. They are the shortest time of numbers of trials. They should be sent to Result.java and CS to record when Result button is pushed.

double vl. Speed of left car, initialed to 0, be refreshed in jTextField4 by ActionListener. Should receive changes from CS in real time.

Other three frames will be commented later.

Nov. 21st 2013

There are totally four frames.

Start is for player(s) to choose the mode they want to play. They are ‘Practise’ mode which single player mode and ‘Race’ mode which is two players racing mode. Choose different mode will be given different second frame as game stats. There is an ‘Exit’ button there to end the game at any time. When chose any mode, there will be three difficulty levels to choose. ‘Easy’, ‘Medium’ and ‘Legendry’ will also be shown in the coming game stats frame. As three difficulty levels are shown, there will be a ‘Back’ button to level player go back to change his game mode.

GameStats is racing mode game stats frame. It has a main clock in the center. Two sub-clocks to store players’ records. Speed meters and direction monitors are below them. There are two ‘Ready’ button for players to click when they are ready. When two ‘Ready’ button are clicked, Main clock will have a count progress, changing background colour from ‘red’ to ‘yellow’ to ‘green’. When ‘green’ lights are on, game starts. With any player reach the finish line first, his time will be recorded on his own sub-clock and sub-clock back ground will be changed to ‘green’ to congratulate him. The other player’s record will also be recorded when he reaches the end, but his sub-clock background will be changed to ‘red’ to let him know his lose.

Single is practise mode game stats frame. It is very simple like only left car of Racing mode is here in practise mode. Sub-clock is bigger and it can only be changed to green when reach the finish line. The only difference is there is a ‘Try Again’ button to let player try the game again to get greater record.

Result is the frame shown the racing result. Player ONE which is left player and Player TWO is the left. Time will be record in form minute (2 digits), second (2 digits), millisecond (3 digits). WINNER will be shown on top textfield as “Player ONE”, “Player TWO” or “TIE”. “RePlay” button to go directly back to the beginning of the game. “Exit” to end the game.

Record is the fame shown the practise record. Player best practise record will be shown in the middle form minute (2 digits), second (2 digits), millisecond (3 digits) as well. “RePlay” button to go directly back to the beginning of the game. “Exit” to end the game.

Direction are all keep in Straight now. Can be detected by ActionListener().

IN SINGLE.JAVA.

if(sec==2&&ms==574){lFinish=true;} to simulate car reach the end at 00:02:574

IN GAMESTATS.JAVA.

if(sec==2&&ms==574){lFinish=true;} to simulate player one reaches the end at 00:02:574

if(sec==4&&ms==4){rFinish=true;} to simulate player two reaches the end at 00:04:004