ICS2 – Programming Assignment 3

Topics: Everything… Due by 11:59PM on the date specified

Marking:

We will be looking for a number of things for this assignment. **Remember that just because your program works does not mean you will get full marks; this is actually only a portion of the final mark for a question.**

- **Style**
  - Documentation (Header, variables, internal comments)
  - Structure (Indentation, white space, code location)
  - Decisions (Variable Naming, documentation descriptions, data types)
- **Design**
  - Although, not directly evaluated you may not begin work until your design is approved
- **Correctness** (Your program produces the expected results)

Academic Integrity:

Plagiarism and other acts of academic dishonesty will be dealt with harshly in accordance with the school's rules. Talking about the questions is allowed, this is called collaboration. **Copying/Using/Giving code from/to friends or other sources is considered cheating.** To ensure you do not cross this line; when talking about a question with someone do so away from a computer with a paper and pen in hand and write out IDEAS, not code. Then when you are finished talking return to your computer and implement the ***ideas*** you have come up with. This is similar to doing research for a report, then applying your findings in your own way.

Follow the instructions below to get started…

- Download the most recent version of the Game Project and unzip it.
- Rename the folder PASS3 (You may have to delete any currently existing folders named PASS3 before renaming)
- When you are done Phase 2, Select the **PASS3** folder, **tap Alt** then **click File→Send To→Compressed(zipped) folder**, and submit the resulting .zip file on the website.

Time Management:

This program will focus your knowledge in an assignment in which you will need to use everything you have learned this semester. There is no time to spare, finish your design then begin programming. This assignment will challenge you. Are you up for the challenge?

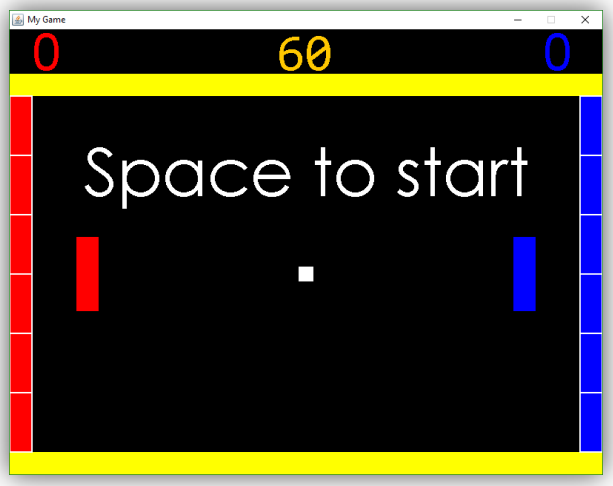
Design Phase:

You are to create a flowchart using draw.io or Word that represents a subprogram that checks for a winner. In this subprogram it must check scores and time. Following the win conditions set out in the program description your subprogram should return a 0 if no winner exists, a 1 if player 1 won and a 2 for player 2.

Pong-Breaker

Task:

Your job for this assignment is to make a game that is a 2-player, cross between Pong and Brick Breaker. It is like Pong in that each player controls a paddle (Rectangle) that can freely move up and down. Player 1 will use the W and S keys, while Player 2 will use the Up and Down arrow keys. There will be a ball that starts in the centre of the play area. At the beginning of each round the ball sits in the centre of the play area until it is time to begin. It will then begin moving one of the four diagonal directions chosen randomly.



The game is made up of two areas, a HUD area to display time and scores as well as a play area where the game is to be played. The paddles cannot move beyond the play area borders. The ball will always move at 45 degree angles and will bounce off of ALL surfaces it contacts. This implies flipping its direction multiplier for either the x or y on collision. These are standard Pong rules.

The brick breaker element occurs behind each player's paddle. At their edge of the screen will be a collection (think array) of bricks that are defending the scoring area (the edge of the screen). If a player has anyone of their bricks hit, that brick is destroyed (moved off screen or faded out) and the other team scores one point. Since there is now a gap in the bricks, if the opponent manages to get the ball through the gap and off the screen they score 2 points.

Every time the ball leaves the screen it is repositioned back to the centre of the screen waiting for the next round to begin. A game has a 60 second time limit that should be displayed. There are two ways to win:

- Be the first to score 7 points
- Have more points than your opponent when time runs out.
- The game is a tie if both time run out and the score is tied.

When a game is complete, a winner should be announced and the players should be given the option to play again. If they choose to play again all data needs to be put back to its starting positions and values. There are a few limitations to consider:

- The paddles cannot move more than 6 pixels per update
- The ball cannot move more than 15 pixels per update
- The ball should start slow and increase by 1 pixel per update every time it collides with anything (resets each round)

Graphical Expectations:

The demo is an example of a level 3 assignment. It shows all needed data and makes use of graphical elements. All elements are rectangles, making it easy to draw.

Level 4

A level 4 will require you to apply your skills. Below is a list of "extras" you could add to your game.

- Use images instead of rectangles
- Add in an Person vs. AI mode
- Sound and music
- Animation of some kind
- Buffs/Debuffs

Programming Assignment Rubric:

The following rubric acts as a checklist. Your mark will be determined by how effectively you completed each item.

NAME:

Achievement Category	Level	Achievement Category	Level
Knowledge & Understanding 1. Structure a. Indentation b. White Space c. Code Location d. Variable Blocking e. Logic Blocking 2. Subprograms and arrays effectively done a. Definitions b. Calling/Accessing c. Procedure vs. Function		**Application** 1. Software Correctness a. Game is fully playable as expected b. All input works as expected c. All limitations are correctly implemented d. Collision detection works as expected e. Game is scored properly f. UI is aesthetically pleasing g. End of round logic works as expected h. End of game logic works as expected	
Communication 1. Documentation a. Headers b. Variables c. Blocks d. Subprograms		**Thinking & Inquiry** 1. Decisions a. Naming b. Documentation Descriptions c. Data Types 2. Appropriate use of subprograms and arrays	

Level 1 (50 – 59%)	Level 2 (60 – 69%)	Level 3 (70 – 79%)	Level 4 (80 – 100%)
- shows little understanding of … - rarely adheres to …	- shows some understanding of … - mostly adheres to …	- shows understanding of … - adheres to …	- shows considerable understanding of … - completely adheres to … - goes above and beyond in showing …