CPSC 359 – Winter 2021
Project Part 2
Raspberry Pi Video Game
Weight: 20%
Due: April 22 @ 11:59 PM

Objective: The objective of this assignment is to expose you to video programming on the Raspberry Pi.

Project Lead TA: Abdullah Sarhan asarhan@ucalgary.ca

Game Objective: You will implement a version of the famous *Frogger* video game. In this game, A frog tries to cross a road, channel, etc… with moving objects. To cross a road, the frog cannot be hit by moving objects. To cross a channel, the frog can use floating objects to avoid drowning. Hence, there are two types of objects: ones that can kill the frog and others that can save it. There will be a minimum of four challenges for the frog (road, channel, and the rest are left for your creativity). The game is won when the frog successfully makes it to the destination after the fourth challenge without consuming all the allotted time or the maximum number of required moves and lives.

Go to https://froggerclassic.appspot.com/ and play the game online and get some ideas. The destination in our game will be when the frog reaches a castle at the top edge of the last challenge screen. Here are two example challenges from the same Website.

Game Logic

The ***game environment*** is a finite 2D *n* x *m* grid (not necessarily a square). The frog can move in all 4 directions using the JOYPAD on the SNES controller. Display the time left for each challenge, the number of lives, the number of steps taken by the frog, and the score at the bottom of each screen. You will show the new challenge screen when the frog completes the current challenge.

- A ***game map*** is an instance of the game environment (for a value of $n \& m \geq 20$)

- Specifies the score of the player, the number of lives left (starting with a minimum of 4 lives), the remaining number of moves, and remaining time. The score is calculated as the remaining time plus the remaining moves plus the remaining lives times a constant (chose it so that it makes sense).
- There should be at least 4 different stages of the map that the frog explores moving forward.
- The objects in the game are the frog and other objects depending on the challenge. Each screen should have a minimum of 15 objects.
- The objects and obstacles need to move in both directions (left-to-right) and (right-to-left)
- The frog can move in all four directions.
- The objects can be obstacles (e.g., cars on a road) or vehicles to carry the frog (e.g., logs on a water channel).

- A ***game state*** is a representation of the game as it is being played, and contains:
  - An instance of a *game map.*
  - *The frog and the different objects positions* on the game map (Initialized to a starting position).
  - The *score*.
  - Number of *lives* left.
  - *Time* left.
  - Steps/*moves* left.
  - A *win condition* flag.
  - A *lose condition* flag

- The ***game transitions*** into a new state when the player performs an action
  - Each time a frog makes a move, the number of remaining moves is decremented by one.
  - If the frog is hit by a moving object (e.g., on a road), drowns (e.g., in a water body), falls (e.g., in the air), etc …, one life is lost.
  - When the frog reaches the castle, the win condition flag is set.
  - If time is elapsed before reaching the castle, the lives are consumed, the steps are consumed, or the score becomes 0, the lose flag is set.

- ***Value-Packs***:
  - You must implement in each challenge at least one *value-pack* of your choice that adds some feature to the game (such as a pack to giving lives to the frog, speeding up the game, increasing the time or score etc.).
  - You may get extra marks for additional creative *value-packs*.
  - The value packs should appear after about 30 seconds into a challenge

- ***Action***:
  - The game starts when the player presses the START button. Timer starts immediately.
  - The JOYPAD causes the frog to move in the required direction
  - Moving into a *value-pack* tile will result in:
    - The effect of the pack applied.
    - The removal of the *value-pack* marking on the screen.

- ***Game ends*** when:
  - The frog reaches the castle (win).
  - The frog's lives become zero (lose).
  - The timer becomes zero (lose).
  - The number of remaining steps becomes zero (lose).
  - The player decides to quit.
  - The game is over when either the *win* or *lose condition* flags is set in the game state.
    Recall that in case of a win, the score is calculated as the remaining time plus the remaining moves plus the remaining lives times a constant of your choice.

Game Interface

- Main Menu Screen
  - The Main Menu interface is drawn to the screen
  - Game title is drawn somewhere on the screen
  - Creator name(s) drawn somewhere on the screen
  - Menu options labeled "Start Game" and "Quit Game"
  - A visual indicator of which option is currently selected
- The player uses the SNES controller to interact with the menu
  - Select between options using Up and Down on the D-Pad
  - Activate a menu item using the ***A*** button
  - Activating Start Game will transition to the Game Screen
  - Activating Quit Game will clear the screen and exit the game

Game Screen

- The current game state is drawn to the screen
  - Represented as a 2D grid of cells
    - All cells in the current game state are drawn to the screen
    - Each cell is at least 32x32 pixels
    - 2D grid should be (roughly) in the center of the screen
  - Different object types are drawn with a different visual representation (at least, in color and shape)

        - E.g.: frog, car, log, etc…
    - Score, lives left, steps left, and timer are also drawn on the screen
        - A label followed by the decimal value for each field
    - If the “Win Condition” flag is set, display a “Game Won” message
    - If the “Lose Condition” flag is set, display a “Game Lost” message
        - Both messages should be prominent (large, middle of screen and centered)
        - Both messages should contain the score achieved.
- The player uses the SNES controller to interact with the game
    - Pressing the Joypad buttons results in a frog movement in the appropriate direction
    - Pressing the Start button will open a Game Menu
        - With two menu items: Restart Game and Quit
        - Visually display menu option labels and a selector
        - Menu drawn on a filled box with a border in the center of the screen
        - Normal game controls are not processed when Game Menu is open
        - Pressing Start will close the Game Menu
        - Press up and down on joypad to select between menu options
        - Pressing the ***A*** button activates a menu option
        - Activating Restart Game will reset the game to its original state
        - Activating Quit will transition to the Main Menu screen
    - If the win condition or lose condition flags are set
        - Pressing any button will return to the Main Menu screen.

Grading Rubric

1. Game Screen

    a. Main Menu Screen ***(5 marks)***

i.	Draw game title and creator names	1
ii.	Draw menu options and option selector	1
iii.	Select between menu options using up/down on joypad	1
iv.	Press A button with Start Game selected to start game	1
v.	Press A button with Quit Game selected to exit game	1

    b. Draw current game state: ***(~~32~~ 35 marks)***

i.	All objects are drawn according to interface specifications	5
ii.	Frog moves in the available spaces	2
iii.	Player can only make the frog move using the joypad	3
iv.	Frog stays stationary if no action is made by the player	3
v.	When the frog moves up and outside the current screen, the move will show new scenes and when it moves down, the old scene will reappear	5
vi.	Two types of objects used and moving in both directions	5
vii.	Score/Lives are drawn as specified.	4
viii.	Implement value-pack as specified.	4
ix.	Game Won message drawn on win condition	2

x. Game Lost message drawn on lose condition 2

c. Draw game menu: ***(4 marks)***
   i. Filled box with border in center of screen 1
   ii. Draw menu options and option selector 1
   iii. Erase game menu from screen when closed 2

d. Interact with game: ***(15 marks)***
   i. Use joypad to move the frog 2
   ii. The first value pack will appear after nearly 30 seconds 7
   iii. Press Start button to open game menu 1
   iv. Press any button to return to main menu (game over). 1
   v. The value packs are randomly shown on the screen 4

e. Interact with game menu: ***(4 marks)***
   i. Use up / down on jpypad to change menu selection 1
   ii. Press A button on Restart Game; resets the game 1
   iii. Press A button on Quit; Go to Main menu 1
   iv. Press Start button to close game menu 1

2. Well-structured code ***(~~15~~ 17 marks)***
   a. Use of functions to generalize repeated procedures 10
   b. Use of data structures to represent game state, etc. ~~5~~7

3. Well documented code ~~2~~10

4. Proper use of threads where appropriate 10

Total **100**

Programs that do not compile cannot receive more than **15 points**. Programs that compile, but do not implement any of the functionality described above can receive a maximum of **25 points**.

You should make your code work on a screen resolution of 1280x720. Failing to do so may result in the code not working properly and losing marks.

Teams: Continue working with your teammate from part 1 where applicable.

Submission: Submit a .tar.gz file of your entire project directory (including source code, make file, README, etc) to the appropriate dropbox on Desire2Learn. Only one submission per team is required. Peer assessment may be required.

Late Submission Policy

Late submissions are penalized as follows:

- 12.5% are deducted for each late day or portion of a day

Hence no submissions are accepted 8 days after the deadline

Academic Misconduct

This project can be done in pairs. Your final submission must be your team's own original work. While you are encouraged to discuss the assignment with your colleagues outside your team, this must be limited to conceptual and design decisions. Code sharing by any means is prohibited, including looking at someone else's paper or screen. Any re-used code of excess of 5 lines must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks

Marks posted on D2L are subject to change (up or down).