

Frogger

Project Document

Copyright © 2015-2016

Muhammad Hammad Siddiqui, Barira Yousuf

Supervisor: Dr. Waqar Saleem

license: <https://opensource.org/licenses/MIT>

Index

Index

1. [Index](#)
2. [Game Design](#)
 - a. [Summary](#)
 - b. [Gameplay](#)
3. [Technical](#)
 - a. [Screens](#)
 - b. [Controls](#)
 - c. [Mechanics](#)
4. [Development](#)
 - a. Game Class
5. [Graphics](#)
6. [Sounds/Music](#)

Game Design

Summary

This game is designed for the final project of a course: Computational Thinking II, at Habib University. The game is built on [SFML](#). Using terminal:

- Download the game from GitHub:
`git clone https://github.com/HammadSiddiqui/frogger-sfml`
- Install SFML using [Homebrew](#): `brew install sfml`
- Use [cmake](#) or an IDE, like [Clion](#), that supports compilation via MakeFiles to compile the program. The make files are included in the project.
- Enjoy the Game!

Gameplay

This is a 2 dimensional game in which a frog needs to cross a road and a river without colliding to an obstacle. The frog needs to reach its friend waiting on the other side.

The obstacles appear in two forms:

- Queue of vehicles running either left or right of the screen
- River Water

Technical

Screens

1. Welcome Screen
2. Main Game Play Screen
3. Gameover/Game-win screen with score.

Controls

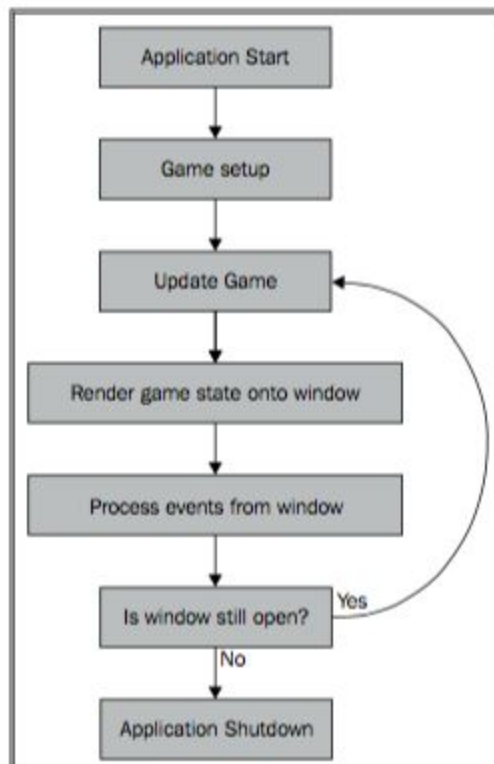
Only 4 Keyboard keys are used in this game:

- Up key to move the frog up
- Down key to move the frog Down
- Right key to move the frog right
- Left key to move the frog right

Mechanics

The cars and log move left and right on a constant y-axis. The frog can move either x-axis or y-axis.

Development



The Game() class

1. Constructor
 - a. initWindow()
 - b. initCars()
 - c. initFrog()
 - d. initWoods()
 - e. initScorecard()
2. run()
 - a. processEvents()
 - b. update()
 - c. render()

All the “init” methods are called inside the constructor to generate all graphics and position them correctly before the game begins.

The real time game happens inside the run function which calls:

- processEvents(): to keep track of all the events happening inside the window.
- update(): the main logic of the game computed based on the type of events received from processEvents.
- render(): this method renders the new values/position of the graphics on the window.

Graphics

All the graphics were provided by: **GARYCXJK@MULTIVERSEWORKS.COM**

The Graphics were all free to use as long as credit is given....Which it is...

Sounds/Music

1. Game Background Music: <https://www.youtube.com/watch?v=MqFhczSljwM>

THE END.