[Date]

CI517 – Game Engine Fundamentals: Animation Playback

Module Leader:Almas Baimagambetov

Lynx Button

Student num: 21806200

Table of Contents

[Introduction 2](#_Toc128745957)

[Implementation 2](#_Toc128745958)

[Design 2](#_Toc128745959)

[Critical Review 2](#_Toc128745960)

[Estimated Grade 2](#_Toc128745961)

[Conclusion 2](#_Toc128745962)

[Links 2](#_Toc128745963)

[Github Repository 3](#_Toc128745964)

[Youtube Video 3](#_Toc128745965)

[References 3](#_Toc128745966)

CHANGE NORMAL FONT BACK TO CORBEL(BODY)

# Introduction

Game engines are software development programs often used to assist and manage game creation. They are vital within the games industry and lay down a basic framework for game developers to expand off of and create games easier than if they had begun from scratch.

XCube 2D is a simple game engine that has been developed using SDL 2.0 and C++. Within this project, the XCube engine will be extended to include an animation playback system which uses CSV files to display and play different animations on screen.

# Design

* Frame rate – how many frames displayed per second (24 & 12 are typically most common for 2d animation)
* Playback time – how many frames are within the entire animation (this divided by the frame rate will be how long the animation is in seconds)
* Keyframes would be useful to space out when the image needs to be changed
* CSV file that includes: frame rate, playback time, image locations, playback frames based upon previous images in an array
  + 24, 128, images/buttcheek.png, images/crack.png, 0, 0 ,0, 1, 1, 1 … etc
  + framerate, playBackRate, image location, max sprite in image, width, height, keyframes based on max sprites !
* Create animation class that loads csv upon creation all above information stored easily within class – it would also need sprite it is supposed to be attached to and possibly the location (x, y) of that sprite.
* Easter egg can be name and module animated at the top of the screen
* Research temp thoughts: does sdl have anything to help you animate or thing about keyframes n such?
* CREATE ANIMATION CSV FILES FOR SAVING read and write to them w fstream – a link explaining more <https://java2blog.com/read-csv-file-in-cpp/>
* Use graphics engine draw texture

Design stuff!!

# Implementation

* Start by discussing the class

## Animation Class

Each animation that is created by the user will need to have a specific CSV file which states all the necessary details of the animation in question. A CSV file is used within the subsystem to allow the user to manipulate their animations without changing the subsystem itself. It also allows for animation detail data to be easily changed when debugging an issue. The CSV file for each animation must contain the following data:

1. Animation name.
2. Frame rate.
3. Playback time.
4. Image location.
5. Number of sprites within each row & column.
6. Json file location, if needed.
7. Sprite width & height.
8. Scale width & height.
9. Which way to flip the animation, if needed.
10. Whether the animation is to be looped.
11. All keyframes.

Text

Description automatically generatedWithin the Animation class two functions are used to read and assign values from the given CSV file. The function readAnimCSV uses the C++ library fstream to loop through all the file’s contents and return them as an array. The function will report back an error to the console if the file cannot be opened or there is more than one row within the file.

The next function, assignCSVContent, uses the content array from the last function to initialise all variables within this instance of the Animation class. The main role of this function is to check the file content for any errors which may make the animation unable to function properly. These error checks include:

Figure \_: readAnimCSV function.

* Ensuring all integers can be converted from strings and that they are bigger than 0.
  + This is completed for the frame rate, playback time, keyframes, sprite amount, sprite width and height.

Text

Description automatically generated

Figure \_: Convertion error checks for assignCSVContent.

* Ensuring locations are valid by checking if the file extension is supported.
  + This is completed for the animation sprite sheet location and when checking for a JSON file.

A screenshot of a computer

Description automatically generated with medium confidence

Figure \_: File extension error checks for assignCSVContent.

* Ensuring enough keyframes have been added to the file by comparing the amount of them to the playback time.

Text

Description automatically generated

Figure \_: Amount error checks for assignCSVContent.

If any of these errors occur, the animations ‘run’ variable is set as false and the animation cannot be used until these issues have been fixed. Each error also outputs to the console so, debugging can be efficient and easy for the developer.

* Write up preload funct

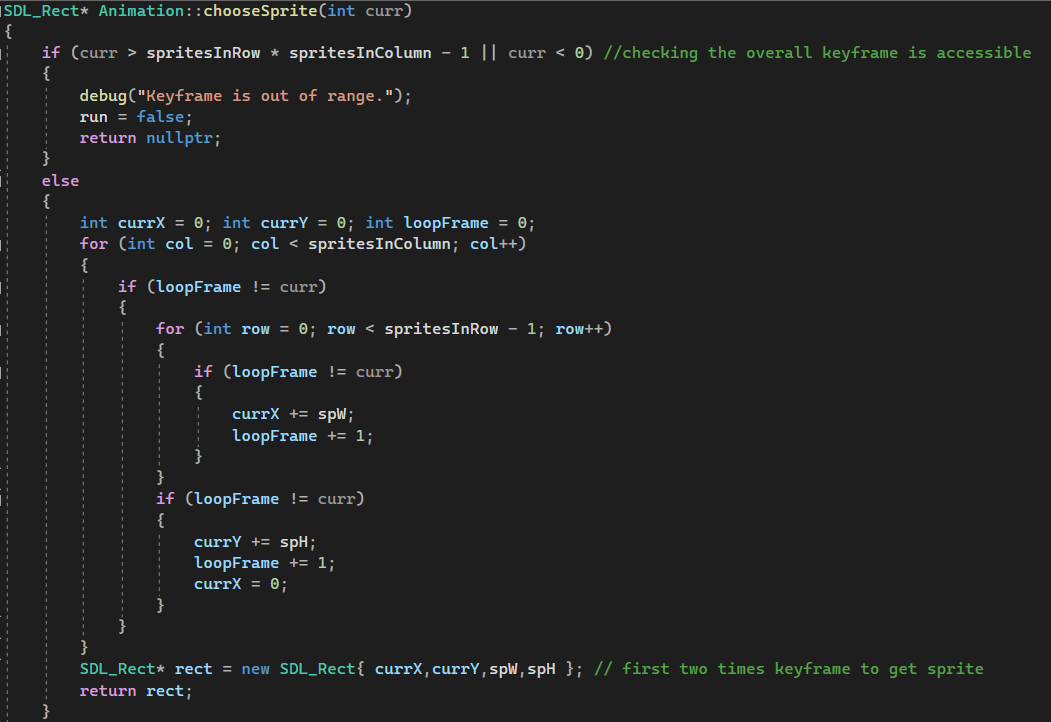


Figure \_: ChooseSprite function.

The function chooseSprite is used when the sprite sheet doesn’t have a corresponding json file to locate the sprites within the image. The function starts by taking the current frame and making sure that it is not outside the range of the sprite sheets sprite amount. It uses the number of rows and columns known within the sprite sheet to loop through the top-left corner of each sprite till the wanted frame is reached. Once the frame is reached, it uses the coordinates of the point found and the raw sprite size to return an SDL rect which can be used to render the frame.

* Write up changeSprite (no json)

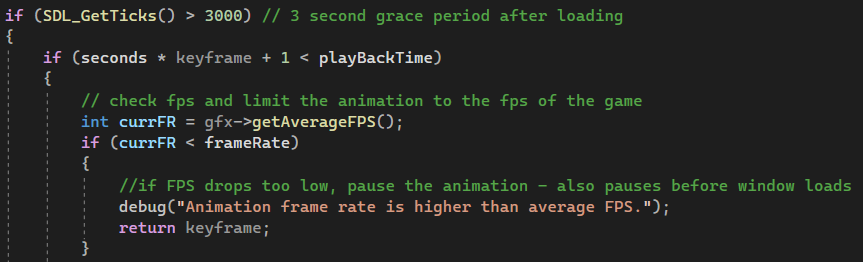


Figure \_: beginning of Change Sprite function.

To begin the function changeSprite retrieves the amount of time which SDL has been initialised and ensure it has been running for at least three seconds. This is to prevent the animation from running before the framerate has stabilised, leading to the animation being faster in the first few seconds it is rendered.

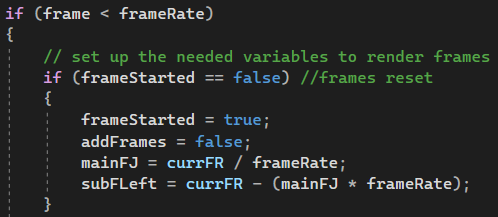
The function then ensures that the current frame rate of the screen is bigger than the frame rate of the animation. If the frame hasn’t been started the function calculates the number of frames that need to be held to achieve the animations framerate.

Figure \_: Change Sprite - Frame reset.

It also calculates if there are any remaining frames which need to be accounted for later. Using these figures, the function uses a nested if statement to switch between the base number of frames to hold a sprite and the added value. It switches between the two values until there are no more frames left within the current framerate that remain outside of the main frame hold.

Text

Description automatically generated Text

Description automatically generated

Figure \_ & \_: Change Sprite - Add Frame System.

Text

Description automatically generatedOnce the max framerate has been reached, the frame is restarted and another second of animation is begun. This allows for longer and bigger animations to take place within the same animation file.

Figure \_: Change Sprite - next second.

Text

Description automatically generatedOne of the last things the function checks for is if the animation has been completed. If all frames of the animation have been displayed at the current rate, it then checks whether the animation should be looped. Looped animations at this stage are completely reset and non-looping animations send a message to the console to report the end of the animation.

Figure \_: Change Sprite – Loop animation.

* Write up render funct
  + Grace period of no animation needed for start up

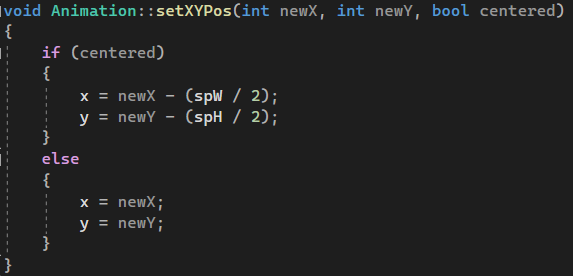


Figure \_: setXYPos function.

The function setXYPos is used within the main game file to change the location which the animation is rendered. The function gives two ways of applying the x and y positions: default and centred. Setting the function to centre divides the sprite width and height by two and takes this from the entered locations. This allows the animation to appear with the entered coordinates in the middle of the sprite. The default option within the function simply renders the animation starting from the entered location, leading the sprite to be rendered below and to the right of the location.

## using animations within the game

* Insert all edits to game.cpp

# Critical Review

* Discuss how plain csvs are not the best security – jsons are probably better?
* Discuss how json helped spritesheets could be used within the system so they can be used as well as square chunked sprite sheets – if you have enough time put this into the engine !!

# Estimated Grade

Engine System:

Game Demo:

Report:

# Conclusion

* Learned about memory management and the difference between pointers references and copies – think about the bug with Almas

# Links

## Github Repository

## Youtube Video

# References

## Articles

ClipStudioOffical (2023) Basics of the Animation Feature. Available at: <https://tips.clip-studio.com/en-us/articles/2047> (Accessed on: 26th February 2023)

Aditya Raj (2022) How to read CSV file in C++? Available at: <https://java2blog.com/read-csv-file-in-cpp/> (Accessed on: 10th March 2023)

Nayonika (2020) CSV file management using C++. Available at: <https://www.geeksforgeeks.org/csv-file-management-using-c/> (Accessed on 10th March 2023)

C Plus Plus (2022) Reference: <string>: stoi. Available at: <https://cplusplus.com/reference/string/stoi/> (Accessed on: 10th March 2023)

Mrtammy2008 (2019) How to handle stoi conversion error? Available at: <https://cplusplus.com/forum/beginner/253696/> (Accessed on: 11th March 2023)

A M (2019) Exception handling with stoi function. Available at: <https://stackoverflow.com/questions/59457656/exception-handling-with-stoi-function> (Accessed on: 11th March 2023)

Kindsonthegenius (2020) C++ String Mnaipulation. Available at: <https://www.kindsonthegenius.com/cplusplus/c-string-manipulation/#t4> (Accessed on: 13th March 2023)

Samual Sam (2020) C++ Program to Find Quotient and Remainder. Available at: <https://www.tutorialspoint.com/cplusplus-program-to-find-quotient-and-remainder> (Accessed on: 13th March 2023)

W3 Schools (2023) C++ Switch. Available at: <https://www.w3schools.com/cpp/cpp_switch.asp> (Accessed on: 13th March 2023)

Chrysanthus Forcha (2022) How do you append to a Vector in C++? Available at: <https://linuxhint.com/append-vector-cpp/> (Accessed on: 13th March 2023)

C Plus Plus (2022) Std::vector::size. Available at: <https://cplusplus.com/reference/vector/vector/size/> (Accessed on: 13th March 2023)

Perforce (2023) Game Engine Overview. Available at: <https://www.perforce.com/resources/vcs/game-engine-overview> (Accessed on: 15th March 2023)

W3Schools (2023) C++ Constructors. Available at: <https://www.w3schools.com/cpp/cpp_constructors.asp> (Accessed 16th March 2023)

Eugen Constantin Dinca (2010) C++ and SDL: How does SDL\_Rect work exactly? Available at: <https://stackoverflow.com/questions/3929920/c-and-sdl-how-does-sdl-rect-work-exactly> (Accessed on: 27th March 2023)

Game Dev Geek (2023) Moving Sprites with SDL. Available at: <http://gamedevgeek.com/tutorials/moving-sprites-with-sdl/> (Accessed on:27th March 2023)

Game Dev Geek (2023) Animating Sprites with SDL. Available at: <http://gamedevgeek.com/tutorials/animating-sprites-with-sdl/> (Accessed on:27th March 2023)

Lazy Foo’ Productions (2019) Loading PNGs with SDL\_image. Available at: <https://lazyfoo.net/tutorials/SDL/06_extension_libraries_and_loading_other_image_formats/index2.php> (Accessed on: 27th March 2023)

Gigi Labs (2015) Loading Images In SDL2 with SDL\_Image. Available at: <https://gigi.nullneuron.net/gigilabs/loading-images-in-sdl2-with-sdl_image/> (Accessed on: 27th March 2023)

The Numb (2017) Lesson 05 – Textures. Available at: <https://thenumb.at/cpp-course/sdl2/05/05.html> (Accessed on: 27th March 2023)

Tbsnsa (2021) How do the source and destination rectangles work in the SDL function SDL\_RenderCopy. Available at: <https://stackoverflow.com/questions/51908174/how-do-the-source-and-destination-rectangles-work-in-the-sdl-function-sdl-render> (Accessed on: 27th March 2023)

SDL Wiki (2023) SDL\_RenderCopyEx. Available at: <https://wiki.libsdl.org/SDL2/SDL_RenderCopyEx> (Accessed on: 27th March 2023)

SSDL Wiki (2023) SDL\_CreateTextureFromSurface. Available at: <https://wiki.libsdl.org/SDL2/SDL_CreateTextureFromSurface> (Accessed on: 27th March 2023)

Johannes Schaub (2022) How to convert a std::string to const char\* or char\*. Available at: <https://stackoverflow.com/questions/347949/how-to-convert-a-stdstring-to-const-char-or-char> (Accessed on: 27th March 2023)

## Assets