08 Implementation Guidelines

Functional Guidelines

1. Implement a feature on your Image set of functionalities that supports multiple frames (rectangle sources) within a single graphical asset
   1. Modify the existing APIs to account for the new change
   2. Experiment with your application in order to test the new feature is working as expected
2. Implement a flipping feature (mirroring effect)
   1. Support horizontal flipping
   2. Support vertical flipping
   3. Support combination of horizontal + vertical flipping
   4. The new feature should work for all supported graphical objects (images & texts)
   5. Experiment with your application in order to test the new feature is working as expected
3. Support rotation effect
   1. Support changing the center of rotation
   2. The new feature should work for all supported graphical objects (images & texts)
   3. Experiment with your application in order to test the new feature is working as expected
4. Play around with the new effects
   1. Create a set of game objects (not Images)
   2. Try to use everything learned so far to implement basic gameplay
5. **Sprite Sheets and Animations - Lecture**
6. Dealing with graphical resources
   1. Copy the running girl resources into the folder
   2. Convert into one big .png file using the **convert** tool

| convert +append \*.png newFilename.png //horizontal append or convert -append \*.png newFilename.png //vertical append |
| --- |

* 1. Use the **mogrify** tool if you need resizing

| mogrify -resize 256x256 targetImage.jpg //warning - keeps aspect  mogrify -resize 367x1034! targetImage.jpg //discards aspect |
| --- |

* 1. Crop the redundant alpha (from 256 height to 220) using **Gimp** (GNU Image Manipulation Program) or other software of choice

1. Enhance the implementation of the Game
   1. Delete the showText, hideText, helloText
   2. Implement a game/game\_entities/Hero class
   3. Public API - init(), deinit(), draw(), handleEvent()
   4. Pass the rsrcId as a config in the init method. Use the whole resource dimensions (1536x220).
   5. Reduce the sprite dimensions to 256x220.
   6. Ouch! The frameRect that is used inside the resource should be represented
2. Expand the DrawParams struct
   1. Add a ectangle frameRect = Rectangle::ZERO
3. Make the Image class populate the frameRect in its methods
4. Renderer should start using the frameRect
5. Individual frames should be stored for each Image
6. Remove the width and height from ImageConfig

| struct ImageConfig {  std::string location;  std::vector<Rectangle> frames;  }; |
| --- |

1. Enhance ImageContainer functionalities
   1. Add struct Vector frames in the ImageContainer (for Rectangle pointers)

| using Frames = std::vector<Rectangle>;  class ImageContainer {  //the textures we'll be drawing  std::unordered\_map<int32\_t, SDL\_Texture\*> \_textures;  std::unordered\_map<int32\_t, Frames> \_textureFrames;  }; |
| --- |

* 1. Implement getter for individual rsrcId Rectangle frames

| const Frames& getImageFrames(int32\_t rsrcId) const; |
| --- |

1. Enhance the Image implementation
   1. Structure

| class Image {  int32\_t \_currFrame { 0 };  int32\_t \_maxFrames { 0 };  bool \_isDestroyed = false;  }; |
| --- |

* 1. Additional Public API methods

| void setFrame(int32\_t frameIdx);  void setNextFrame();  void setPrevFrame();  int32\_t getFrame() const; |
| --- |

1. Incorporate the frameRect in the Text struct as well
   1. Don’t forget to integrate it in all 3 creation methods - createText, reloadText, setTextColor
2. **Flipping and Rotation - Lecture**
3. Expand DrawParams struct - adding FlipType

| enum class WidgetFlip : uint8\_t {  NONE,  HORIZONTAL,  VERTICAL,  HORIZONTAL\_AND\_VERTICAL  }; |
| --- |

1. Enhance the Renderer draw method
   1. SDL\_RenderCopy() will no longer be used
   2. Use SDL\_RenderCopyEx()
2. Enhance the Game
   1. Associate the KEY\_U/I/O/P with the 4 flip types on the running girl asset
3. Expand DrawParams struct - adding rotation

| double rotationAngle; |
| --- |

* 1. Use those members inside the Renderer - SDL\_RenderCopyEx() method

1. Use KEY\_K/L to achieve rotation of the running girl asset inside the Game with +- 15 degrees
2. Expand DrawParams struct - adding rotationCenter

| Point rotationCenter; |
| --- |

* 1. Use those members inside the Renderer - SDL\_RenderCopyEx() method

1. Use KEY\_H/J to achieve setting of the rotation for the running girl asset in the Game to Point::ZERO or the actual center of the text
2. Enhance the game even further
   1. Implement a rotating wheel class.
   2. Use the same approach as the hero
   3. Set the rotationCenter of the Wheel to the texture center and move it with the keyboard
   4. Move the rotation of the text from the running girl asset to the wheel
3. Bonus: have some fun with the Hero and Wheel structs. Get creative