****

Poo Poo Pipe

Testing Report

Playtest date: Dec 14, 2019

GAM200F19KR

FALL 2019

**Name Title Coding Responsibility**

**Kim HaGyeong          Producer                   Architecture**

**Son JuHye     Graphic programmer   Graphics**

**Lee UiJin               Technical Director                   Architecture**

**Hong eunjin Designer                       Design, Sound**

# Mechanism Testing

PooPooPipe is 2d puzzle game that player should connect the scattered pipe puzzles from start pipe to the end pipe through fitting puzzle in the blanks and rotate puzzle. Thus, the main mechanisms are rotation, translation, and pipe connection.

* rotation
  + Pipes can be rotated by inputting the desired angle value with function *'setRotation'*.

***(Errors and problems)***

->It does not move accurately depending on the angle entered. for example, pipes are inputted 60.f angle into setRotate but cannot rotate exactly.

-> The shape was crushed when it rotated.

***(testing)***

- Check the coordinate of rotated shape and value after pass *'setRotate'* function with std:cout.

- Added various angle values and compared how their value changed through 'setRotate' function.

- because it may be an error caused by wrong equation, math equation is changed several times.

***(realize)***

-> The value to be entered should have been in radian value, but we were entered just degree value, not radian. thus the pipes were not rotated well because the lossing of value.

-> In the meantime, we used glm as a math library, but rotation equation in glm is different we desired. so there is a problem with rotation because of it.

***(Solution)***

* we changed glm to custom math library that we wrote and insert radian value into setRotate function.
* It works well.
* Translation
  + Move the mesh according to the mouse cursor value.

***(Errors and problems)***

-> The afterimage of the mesh remains, and did not move straight.

***(testing)***

- Using std :: cout, we checked the coordinates of the moved shapes one by one

- Using std :: cout, we checked if the mesh was clicked correctly

- we used std :: cout to make sure the mouse coordinates are coming in correctly.

***(realize)***

-> Since the value of the origin did not follow the mouse, the afterimage remained.

***(Solution)***

* We made a get function that takes the coordinates of the mouse cursor and kept updating it to the mesh coordinates.
* works normally
* pipe connection
  + When facing each other side with a pipe in the hexagon, we should determine whether the pipes are connected or not.

***(Errors and problems)***

->we didn't know how keep tracking pipe connecting.

- asked to professors algorithms that how can we implement.

- asked at *‘stack overflow’* how can we implement it.

***(realize)(Solution)***

* create bool type array, and each time it is rotated, the values are shifted.

the values of each side are compared to determine the connection.

***(Errors and problems)***

->there is strange pipe that couldn't keep tracking pipe connection along the location.

***(testing)***

- using std::cout, we check the value according to condition.

- we took the contents(codes) in the conditional satement out and it with revising the codes and value again to confirm where the error occurred.

***(realize)***

->the presence or absence of connection was also being determined in a position where the pipe did not receive a position value and should not be connected.

***(Solution)***

* adding more conditional statement so that pipe can be connect or not connect by location too.

# Graphic Testing

Poopoo pipes include textures, cameras, meshes, images, timers, window applications, inputs, and transforms.

* GLFW
* Link GLFW libraries to use GLFW.

***(Errors and problems)***

->GLFW does not connect well and error occurs in SHADER.cpp

- Ask the professors why there is an error

- Ask the Internet how we can connect.

***(realize)***

-> glfw doesn't do xcopy correctly

***(Solution)***

* We connected xcopy with the help of a professor to make the program work.
* TEXTURE coordinate
  + we want to insert a picture in a shape

***(Errors and problems)***

-> Texture coordinates are weird

***(testing)***

- Trying out various coordinates to see where it fits.

***(realize)***

->The order of the coordinates was inserted incorrectly

***(Solution)***

* The order of the coordinates was corrected and the picture floated normally.

# Audio Testing

***(Errors and problems)***

-> Had problem of putting the sound effects to the game, which is doing infinite loop while they should not.

***(testing)***

- Place the sound effects play function which I think is the right place

- Control the loop count by adding another parameter, which will set the channels loop count.

***(realize)***

-> The function I putted the sound effect was in the function which does the collision check, which keeps updates the mouse location. So, of course the infinite loop happens

***(Solution)***

* Implement the sound effect play function which does not keep updates it information.

# Architecture Testing

the architecture programmers created most of the practical levels, so there were n difficulties in programming using the architecture part.

* Component
  + use *'addComponent'* function to each object to add the desired component.

***(Errors and problems)***

->with using addComponent function, the program would burst and terminated by null value.

***(testing)***

- with debugging, we checked where the null value entered.

- we check which component has an error with clearing the components.

***(Errors and problems)***

->the value was not pushback so the null value was entered.

***(testing)(Solution)***

* make all component can be pushbacked and exact values are inserted..
* Puzzle Component
  + add a component that can checks pipe connection.

***(Errors and problems)***

-> the program was bomb and terminated with puzzleComponent.

***(testing)(realize)***

- with debugging, we checked where the null value entered.

***(testing)(realize)***

-> there is a error with the header includes order.

* change the order of header include in order to program work well.

# Prototype Testing

The prototype engine contains only relatively simple features. Floating shapes, inserting photos, and simple translations were all that was possible. Therefore, most of the functionality has been added now and is documented in another testing section.

* Image
  + We want to insert an image into a shape

***(Errors and problems)***

-> we used freeImage library to float an image, but we want to use another library to float images

- We asked our friends who used other libraries to find out how.

- We searched on YouTube or the Internet and found a way.

***(Solution)***

* We can now insert images using Stb.
* Object
  + Make object cpp to make a large amount of objects easy

->I don't know what structure the object should have

- Studying architecture while reading books

- Ask seniors how to make it.

* We created an object cpp and a manager to manage it so that we can manage it.