Design Document

Kate Glazko HW 5

Purpose:

This piece of software is an educational game with the purpose to teach multiplication to kids in an artful and accessible way. This software utilizes polymorphism, classes, and graphical designs to create an engaging and entertaining player experience.

Classes:

Gropaga = The enemy for level one. It is a plane, it has a back-and-forth motion. Upon contact with the gropaga, the player will lose points. Losing enough points may knock the player back a level or two.

Inglip = The main character. This character is operated through key strokes; up, down, right, left . Other characters in the game; such as the shooting gold coin, are also operated through these key strokes. The y appear as though they are barreling towards the main 'Inglip' character.

Kitten= a class whose name was not modified. The classes rain, snow, and sleds are all based on the Kitten class. These classes move in a variety of ways; the snow balls, for example, slightly follow the main Inglip character. Others will move in a circle, up and down, or in a specified pattern.

The Buck and Sman classes are the 'following' classes. Buck is a helicopter that was designed to follow the main Inglip character in a spastic, choppy way resembling a real helicopter. The Snow Man follows Inglip in a smoother way, and does not appear in multiply places as does the helicopter.

The tree and bush classes utilize polymorphism. They are a decoration classes and are printed along with the gold coins during level one.

Main Functions:

GetRect and rect in QT were crucial in my game. In order to be able to determine the position of an object and therefore be able to determine any future collisions, we used Rect to find the position and intersection of any two objects. Such intersections would result in point deductions from a character's score- which could potentially result in an object being 'bumped' down a level.

One of the most importantly utilized functions that nearly every single character used was AutoMove. AutoMove differed for each character significantly; some characters simply traversed a linear path; some traveled in a way in which they bounced against walls.

The shooting characters utilized a type of movement which was a mix between AutoMove and a following function of the Inglip main class.

The following characters utilized a type of movement which was a delayed following of the Inglip Main Class.

The timer function was crucial in the printing the characters, as well as the process of running the whole game.

The CASE function checked the keys pressed and Inglip would follow in the direction of the keys pressed. The objects that were required to travel in the directions of Inglip were also affected by the pressing of the keys.

The Check Collision function is where most of the actual significant events of the game take place. Utilizing a series of if-statements, the Check Collision function will commit a series of picture switches depending on the level achievement that the player reaches. The pictures are a series of numbers and characters, and the backgrounds are a series of worlds with different mathematical multiplications. When the player 'intersects' with, or collides with the correct answer, the background will switch to a new world and the characters will switch to new animations. After three such interactions, the player will level up. You will come across a PAUSE screen; press the 'pause' button to unpause the game and continue onto the next level. The next level implementation is an if-statement within the collision function for level two. New characters will be called, and the levels will increase in difficulty. The helic opter, Buck, will make an entrance in this level, and collisions with Buck will result in equally drastic reprecussions for the player. However, if they are able to achieve the correct answers three times, they will be taken to the third level, which is also within the collision function. The third level is the most difficult and only features one challenge- of multiplication; but involves a 'shooting' present, following snowballs, and a following snowman. If you survive, you have achieved victory!

Global Variables:

Several bools are used in my program to switch between levels without being overridden by each new iteration of the collision function.

Test Cases:

This game is best played by adhering to the proper keystrokes (arrow keys). The game must be quit in order to restart gameplay. Pause is the same button as 'unpause'.

GUI:

The interface is a combination of a QT Window and a QT Board. The QT window utilizes an LCD to keep track of levels and the score. For the game player's pleasure, the window has been made pink. The game play takes place on a 700 x 700 QT Board. The GamePlay is started through MAIN and the window is displayed in window.show().