Assignment 5: QT Game

Frogger

**Purpose:**

This program will execute a game using the QT. The game will allow the user to control a character to cross the streets filled with enemies to seek the end point.

**Requirements:**

This program is required to allow the user to control a player. The game must have at 12 objects. Objects just include polymorphism and inheritance. Two objects are to chase the players. Players are given a certain amount of lives and have scores for achievements they accomplished.

**How to play:**

The player is controlling a frog. The goal is to reach the other side of the street. The player must avoid the enemies that are moving across the streets. If the frog moves into the enemies, or the enemies hits the frog, 1 life is deducted. There are 3 lives total. There are enemies that chases the frogger while others give frogger powerups, such as lives or faster movement speed. Once reaching endpoint, the game will move on to the next level. Points are gained by reaching the clover. Once 3 lives are gone, game over and program ends.

Controls:

Arrow key UP – move up

Arrow key LEFT – move left

Arrow key DOWN – move down

Arrow key RIGHT – move right

Space bar – start game or replay

Esc - quit

**Classes:**

**SUPER CLASS:**

**VEHICLE: base class**

Kills Frogger if touched.

Vehicle();//Constructor

Vehicle(int xSpeed, int ySpeed);

virtual ~*Vehicle*();//Destructor

void moveBottom(int);

void moveTop(int);

void moveLeft(int);

void moveRight(int);

QRect getRect();

QImage & getImage();

void movePositioning(int, int);

virtual void *autoMove*() = 0; Function to be implemented in subclasses

protected:

int xdir;

int ydir;

QImage image;

QRect rect;

CAR: is a superclass of Vehicle.

Medium size vehicle, fast move speed

public:

Car();//Constructor

Car(int xSpeed,int ySpeed);//Constructor with movement speeds

~*Car*();//Destructor

public:

void *autoMove*();

TRUCK: is a superclass of Vehicle

Bigger size vehicle, slower moves peed.

public:

Truck();//Constructor

Truck(int xSpeed, int ySpeed);

~*Truck*();//Destructor

public:

void *autoMove*();

};

ENEMY: SUPER CLASS

Kills frogger if touched.

public:

Enemy();//Constructor

virtual ~*Enemy*();//Destructor

void moveBottom(int);

void moveTop(int);

void moveLeft(int);

void moveRight(int);

QRect getRect();

QImage & getImage();

void movePositioning(int, int);

virtual void *autoMove*() = 0; function to be implemented in subclasses

int xdir;

int ydir;

protected:

QImage image;

QRect rect;

DOG: is a subclass of Enemy

Moves left to right, flip images!

public:

Dog();//Constructor

Dog(int);

~*Dog*();//Destructor

public:

void *autoMove*();

ANGRYDOG: is a subclass of Enemy

Chases the player. Also flip images depending on where he’s facing.

public:

AngryDog();//Constructor

AngryDog(int xSpeed);

~*AngryDog*();//Destructor

public:

void *autoMove*();

TIGER: is a subclass of Enemy

Moves super fast, but slows down over time. Chases player.

public:

Tiger();//Constructor

~*Tiger*();//Destructor

public:

void *autoMove*();

LION: is a subclass of Enemy

Moves slow, but speeds up over time. Chases player.

public:

Lion();//Constructor

~*Lion*();//Destructor

public:

void *autoMove*();

};

POWERUP: is a super classes!

Gives power when the player picks it up.

public:

PowerUp();//Constructor

virtual ~*PowerUp*();//Destructor

QRect getRect();

QImage & getImage();

void movePositioning(int, int);

protected:

QImage image;

QRect rect;

HEART: is a subclass of PowerUp

Image is a heart.

Does not need functions. Add a live to player when picked up.

Will only use superclass’s function.

public:

Heart();//Constructor

~*Heart*();//Destructor

public:

SPEED: is a subclass of PowerUp

Image is a green potion.

Does not need functions. Increases the player speed when picked up.

Will only use superclass’s function.

public:

Speed();//Constructor

~*Speed*();//Destructor

public:

SLOW: is a subclass of PowerUp

Image is a red potion.

Does not need functions. Decreases the player speed when picked up.

Will only use superclass’s function.

public:

Slow();//Constructor

~*Slow*();//Destructor

public:

FIRE: is a subclass of PowerUp

Image is a bonfire.

Kills the frog once touched.

Does not need functions. Decreases the player speed when picked up.

Will only use superclass’s function.

public:

Fire();//Constructor

~*Fire*();//Destructor

public:

FROGGER: the main character

Image is a frog.

Arrow keys to move!

public:

Frogger();

~Frogger();

public:

void resetState();

void moveLeft(int);

void moveRight(int);

void moveTop(int);

void moveBottom(int);

int getWidth();

int getHeight();

QRect getRect();

QImage & getImage();

void movePositioning(int, int);

private:

QImage image;

QRect rect;

CLOVER:

End point, moves to next level when touched by frogger.

public:

Clover();//Constructor

~Clover();//Destructor

public:

QRect getRect();

QImage & getImage();

void movePositioning(int, int);

private:

QImage image;

QRect rect;

**Global Data/Functions:**

None!

**High-level Architecture**

This program has 3 bases classes, 10 subclasses, and 2 independent classes. The three base classes are: Vehicle, Enemy, and PowerUp. Vehicle has trucks and car. Enemy has AngryDog, Dog, Lion, and Tiger. PowerUp has Speed, Slow, Heart, and Fire. Two independent classes are frogger and clover. Frogger is the main character and wil die once touching a vehicle or an enemy. Frogger will gain special abilities when it touches a powerup: heart will give the frogger live, fire will kill, speed will increase the movement speed, and slow will decrease the movement speed. Frogger will advance to next level once touching clover. All classes interact with each other and are created in the Game file, which is created in the main.

**User Interface**

This program has a unique user interface with the game drawn, the lives and scores on the top. The player is unable to enter the line containing the scores and lives, or leave the game space. If he attempts to leave the game space, he is unable to move and will stay on the same spot. This game will requires the user to press the arrow keys to move, spacebar to replay or start, or esc to quit. The user can quits the game at any time by pressing esc. The user can move by pressing arrow keys. Any other input entered by the user other than arrow keys, esc, or spacebar will not work and will not do anything in the game.

**Test Cases**

To test this game, the user can play the game by executing it. Upon the start of the game, the player can press space bar to start. To move, the players can test the arrow keys. To test that the game works without errors, the user can try to leave the game space to make sure the character is confined to the game space. To check the lives are valid and correct, the user can run into trucks to die, or let the truck run over the character and check the LIVES on the top left corner to see if it has decreased. Upon death, the frog should reappear at the original location at the bottom. Once the round is won, the game should regenerate a new random clover, cars, and trucks at random locations. The scores are increased whenever the user wins, so when the user wins the game, the game is automatically restarted.