Developing a Driving Training Game on Windows Mobile Phone Using C# and XNA

Roger Ouch and Blake Rouse
Computer Engineering and Computer Science Department
University of Louisville
Louisville, Kentucky 40292
E-mail: r0ouch01@louisville.edu, sngskunk@gmail.com

***Abstract*— Since the release of the Windows phone [1] in 2010, the mobile phone industry has the opportunity to develop new and diverse applications on this platform. The iPhone [2] is one of the market leaders of smartphones; there are over 350,000 applications available for download. Although the Windows Phone is released much later than the iPhone, there are lots of opportunities to develop new and interesting game applications. Educational games [3] are a category of games that have been specifically designed to teach people about a certain subject, expand concepts, reinforce development, understand a historical event or culture, or assist them in learning a skill as they play. in this paper, we introduce the development of an education game using the C# language and XNA on a Windows Phone platform that teaches people the basic of driving on roads.**

Keywords*— *video game, educational game, Windows Phone, XNA.

I. INTRODUCTION

In 2009, the total number of reported traffic deaths in the United States was 42,642 [4]. The mortality of young drivers between 16 and 24 years old is much higher than the rest of the population [5]. One way to enhance road safety and lower the number of deaths on the road is to better educate people and improve their driving skills.

Sometimes, learning from books may be boring and inefficient for some young people. So we propose an entertaining approach to deal with road safety problems. Young people are fond of new technology and spend a lot of time with their cell phones. So the idea is to teach people the basic driving rules and let them play at the same time.

In this paper, we present the development of an educational game that intends to teach the rules of road driving on the Windows Phone platform, using C# and XNA languages.

II. EDUCATIONNAL GAME

Games are a form of organized play. Games have stated goals and rules of play to guide players to the goals. The goal can be fanciful or purposeful. According to Salen and Zimmerman [6], a game that is well-designed yields "meaningful play," a condition very like learning. They define meaningful play as "what occurs when the relationships between actions and outcomes in a game are both discernable and integrated into the larger context of the game." When game design focuses on learning outcomes, it is possible to preserve playfulness while allowing meaningful learning.

There is a widespread consensus that games motivate players to spend time on mastering the skills a game imparts. "A number of distinct design elements, such as narrative context, rules, goals, rewards, multisensory cues, and interactivity, seems necessary to stimulate desired learning outcomes." [6]

III. GAME DESIGN

In our educational game, we focus on several aspects of how to teach the user the basics of driving. For examples, we focus on speeding, on the road signs, driving without crossing the center line, etc. Furthermore, a score system showing if the player has passed or failed the driving test has been implemented.

Many other features have also been designed to make this game enjoyable and entertaining to play. Similar to other cell phone applications available in the market, we have designed several other features such as a game option menu, different maps, music options, sound options, and help options.

Figure 1 shows the state machine of our game application.

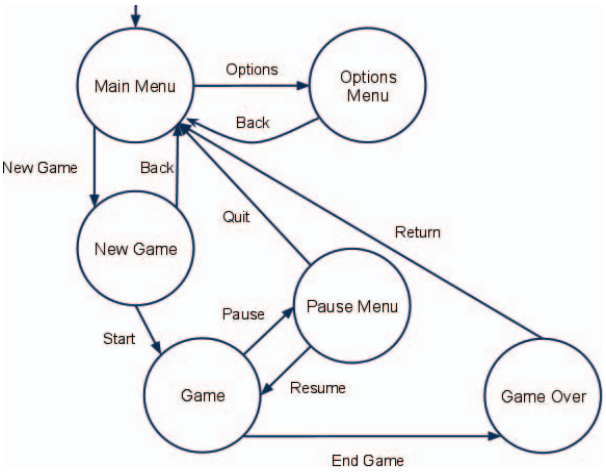

Figure 1. The state machine of the game

IV. INTERFACE

The interface design of the game uses the XNA library [7]. In the development of the game interface, we emphasize the functionality rather than the visual impact of the interface. We make sure that all functionality of this game is working correctly although the graphic design may seem simplistic.

There are two types of interface designs needed for our application. The first one is the design of the menus of the game. Figure 2 contains the main menu of the game.

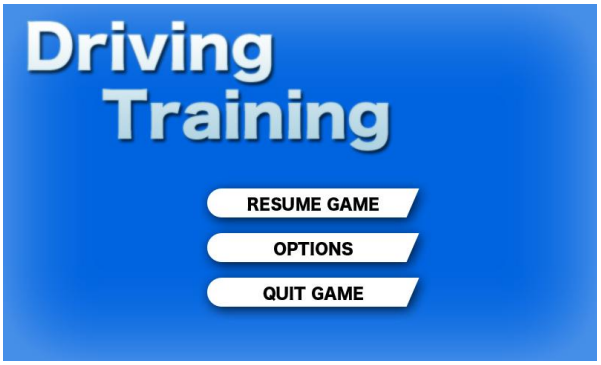

Figure 2. Menu interface sample

The second interface is the game play interface. We specify the elements that should be presented on the game play interface: a main screen, a gas button, a break button, an odometer, a speedometer, and a message box for displaying information to the player. Figure 3 contains a typical screen shot of game play. The graphical designs of other objects are also necessary in the game, such as cars, roads, road signs, etc.

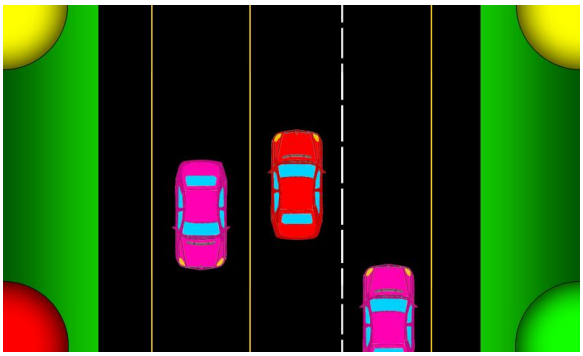

Figure 3. Game Play interface sample

V. CORE GAME ENGINE

To create the core engine of the game, we use predefined classes from the XNA framework library. For this application, we decide to use 2D texture functions. Thus in the game application every object inherits the predefined class GameObject from the XNA framework.

For our game development, we use object oriented design and the C# language for modeling. Our design can be split into two groups of classes. The first group is the menu game objects. All the menu and button classes inherit the GameObject class. Figure 4 contains the class diagram for the application menu pages.

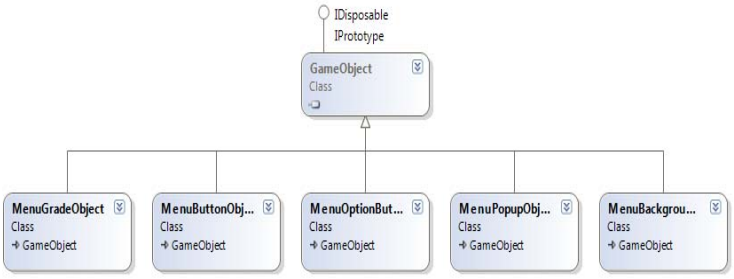

Figure 4. Game menu classes

The second group of classes is game play classes. Each object created and used during the game play is an object inheriting also from the GameObject class. We create, for instances, a Car class, a Road class, a GameButton class, a GazButton, etc. Figure 5 shows the class diagram for the game play.

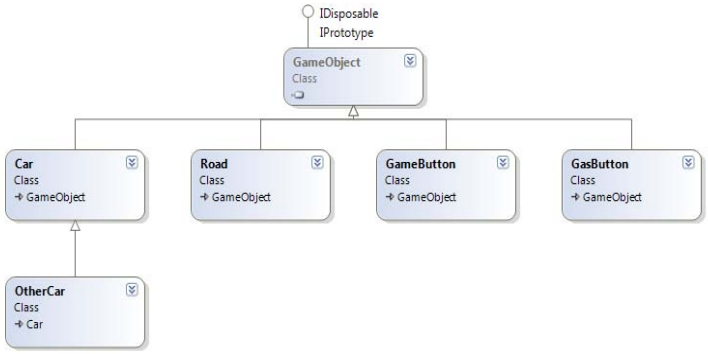

Figure 5. Game play classes

Several other classes have been designed for the core engine. A PhysicsEngine class is implemented to control the object motion and detect collisions. The PhysicsEngine is one of the most important classes in the game because it is used for controlling object, such as car movement, detecting collision between two cars, computing speed, and checking if a car has crossed a center line.

VI. SCORE SYSTEM

A ScoreSystem class has also been implemented to evaluate the performance of the player. The total number of points that a player has is 100. If at the end of the game the player has more than 75 points, the road driving test is considered passed. If the score is less than 75 the player has failed the test. The score system of the game is based on real driving rules that people must master for their driving examinations. To summarize, a system of penalty is given for bad driving practices.

The followings are some examples of game play penalties. A player loses points if he forgets to use the turn signal before changing lane, or if he crosses a center line in a two-way road. The driver also loses points when he exceeds the speed limit. The amount of points lost in the game is a function of the speed over the limit. Thus the player will lose more points if he exceeds 30 miles per hour than if he exceeds only 10 miles per hour over the speed limit. Finally, if the player gets into a car accident, the test is completely failed. Figure 6 shows a screen shot of the end of game.

Figure 6. Test failed screen

VII. GAME SCENARIOS

To create a game that teaches people the basic driving skills, we need to recreate and simulate the real driving conditions. The list of scenarios that can be created is long; we have, for examples, driving in a highway, driving in the city, learning the different road signs, driving at night, knowing the appropriate driving speed, passing a bicycle, car parking, etc.

For the purpose of the game development, we started with the simplest game scenario, which is driving in a highway. All other scenarios can be added later in future development.

If the driver fails the driving test, he can improve his score by restarting a new game and trying to correct his driving mistakes.

VIII. CONCLUSION

The Windows Phone platform has offered a lot of opportunities for developing new kinds of applications and games. Every year in the United States, the number of deaths on the road is very high especially for young people; the development of an educational game that teaches the basics of road driving may be helpful to enhance road safety.

The XNA platform provides excellent tools to create a game on the Windows Phone. The design and development is facilitated by using predefined functions available in the MSDN library. The graphics design we have accomplished in this game may be simplistic because we have focused on the interface functionality rather than the appearance of the look-and-feel.

In our game, we develop the core game engine using the predefined GameObject class. We develop our own classes for other components needed for the game, such as the physics engine and the score system component.

Only one scenario has been implemented in the current version of the game. However, to make the game more educational, several more scenarios can be added in future development.

IX. FUTURE IMPROVEMENTS

To make this driving game more attractive and enjoyable to players, there are many improvements that can be done in the visual design of the application. Also, different scenarios can be added to put the player in different driving situations.

ACKNOWLEDGEMENTS

We would like to thank Dr. Rammohan Ragade for his advices and support during this project. We would like also to thank Dinh Duy Duc for his participation in this project.

REFERENCE

[1] Windows phone page: http://www.microsoft.com/windowsphone/en-us/default.aspx

[2] Iphone app page: http://www.apple.com/iphone/apps-for-iphone/

[3] Wikipedia: http://en.wikipedia.org

[4] Ranking: http://www.photius.com/rankings/road_traffic_deaths_country_rankings_2009.html

[5] U.S. Census Bureau, Statistical Abstract of the United States: http://www.census.gov/compendia/statab/cats/transportation/motor_vehicle_accidents_and_fatalities.html

[6] Salen, K. and Zimmerman, E... Rules of Play: Game Design Fundamentals. Cambridge, Mass., and London, England: The MIT Press, 2004

[7] XNA Creators Club: http://creators.xna.com/

The MSDN XNA Developer Center: http://msdn.microsoft.com/en-us/xna/default.aspx

BIOGRAPHY

Roger Ouch is a Ph.D. student in Computer Engineering and Computer Science Department at the University of Louisville. He received a master degree in computer science from the IUP, University of Avignon, France, in 2003. His research interests include high-performance computing, GPGPU and CUDA, data mining and bioinformatics.