Gaming Mechanics in Alice

Sridurga Shilpa GanapathiNed and Karen Villasverde
Department of Computer Science,
New Mexico State University,
Las Cruces NM 88003, USA
Email: ganapathin@nmsu.edu, karenv@nmsu.edu

Abstract

This work describes the design and prototyping of various gaming mechanics using Alice, a 3D programming environment that has 3D graphical objects and drag-drop functionality.

Introduction

The most thriving industry in recent years is the gaming industry. A game is basically a set of actions that require two components, a game and a player and are very interesting because they are interactive, engaging and enjoyable. Games are of different genres and each game involves various stages like coming up with a creative idea, graphical representation of the objects in 3D or 2D, prototyping and finally implementing the idea with all the hardware and software resources available.

The gaming mechanics analyzed are prototyped in Alice, a 3D programming environment developed using Java. Alice is an open source object-oriented programming language with an integrated development environment with drag and drop facility and 3D models. It has intuitive methods and functions and the user can also define his own methods or functions. Below is the Alice programming interface.

Methods

Collision Detection

The most common technical problem to be solved in games is collision detection. Many games, such as shooting and racing games use this mechanic.

In Alice, we can achieve collision detection by creating a user-defined method called "onCollision" and an event that checks whether the objects are colliding or not. This event uses a built-in function "isWithinDistance" that calculates the distance between two objects.

In Fig. 2, the distance between the center of the jet and center of the asteroid is measured using the built-in function "is within distance of" and if they are just half a meter apart from each other this event calls the method onCollision that change the velocity of the jet to zero making the jet invisible and brings in a feeling to the player that the object is destroyed.

Camera Views

Camera views are different camera positions that let the player see the possible views of the game giving a rich gaming experience to the user. These views give a better understanding of the position of the object in most role-playing games and the commonly used views are front view, back view, side view and top view.

Alice has a method called capture views that creates different camera views. A view is captured after changing the camera's position and a built-in method "set camera point of view to" is called to change the camera's position to the view captured.

Conclusions

Gaming mechanics in Alice allow us to identify design and analyze games. The research was based on various games developed by game designers and hobbyist which allowed us to examine and prototype a wide variety of game mechanics.

In spite of these advantages, Alice still has limitations: Alice is not stable; Alice does not support large memory games; Alice acts different on different operating systems and graphics cards, and has limited objects in its 3D library. However, Alice will improve on these limitations on its newer versions.

References


Acknowledgements

This work was supported in part by NSF Grants CNS-0544592 and CNS-0526152, and by a graduate research assistantship from New Mexico State University.