

Use of Microsoft Visual Studio Express Edition and DirectX to Simulate Movements of a Colony of Penguins that must be tagged for a Science Mission Education

Sandy Harrison (WSSU), Tushar Majithia (Mentor), Spencer Weeks- Jamieson, Jasmin Rivers (ECSU), Terrance Mason (MVSU), Dr. Arvin Agah (Mentor) (KU)

Abstract-

Students begin to take a general interest in becoming scientists or engineers at early stages in their life, such as the middle school level. It is imperative that they are targeted at this point in their lives to be encouraged to pursue their education in science, technology, engineering, and mathematics (STEM) fields. The dilemma is how to get young scholars interested in STEM areas? Our proposed tool is an educational and entertaining video game, as a vast number of young students tend to play video games everyday. Our goal is to combine education and fun into a game in order to teach the players about polar research. The objective of this URE Summer project is to enhance an existing video game to educate young students on the Polar Regions including challenges that scientists face. In developing this game, the team is specifying, designing, implementing, and testing the software for the game. The platforms and packages that the team has utilized include Visual C++ Express Edition and Microsoft Platform SDK (Software Development Kit). Conducting this research and creating this game has taught us to evolve into better programmers and software engineers while providing an entertaining educational resource to students in order to get them interested in STEM disciplines.

Introduction

The primary objective of our research project is to educate young students on the Polar Regions including the many challenges that scientists face. When students take part in this video game, they will engage in educational enjoyment. By tagging the penguins, the game demonstrates startled penguin behavior. From

igloos to the snow, the game itself displays a type of climate that some penguins live in. In developing this game, the team is specifying, designing, implementing, and testing the software. Some of the programs that the team utilized were C++, VC++ Express 2005 Edition compiler, and Microsoft Platform SDK (Software Development Kit). This project consisted of upgrading a previously developed game. While using C++ in the aspect of programming, we modified the previous program's features with realistic graphics. This modification would enrich it with a realistic perspective that would better serve our primary objective. To validate if middle school students would enjoy this game, we used the 2007 CReSIS Middle School Program students as subjects for testing. Penguins are one of the most extraordinary amphibious flightless birds that primarily live in the Southern Hemisphere. The number of penguin species has, and still is, a matter of debate among scientists. The diverseness of the penguin species varies from between 17 living species.

I. Software

Microsoft Visual Studio 2005 Express

Edition: is a complete set of development tools for building ASP.NET Web applications, XML Web services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions.

Microsoft Visual C++ 2005 provides a powerful and flexible development environment for creating Microsoft Windows-based and Microsoft .NET-based applications. It can be used as an integrated development system, or as a set of individual tools.

Microsoft DirectX (Direct eXtension): is a collection of application programming interfaces for handling tasks related to multimedia especially game programming and video, on Microsoft platforms. It is a direct competitor of the OpenGL standard, maintained by the Khronos Group.

Direct3D is widely used in the development of computer games for Microsoft Windows, Microsoft Xbox and Microsoft Xbox360. DirectX is also used among other software production industries, most notably among the engineering sector because of its ability to quickly render high-quality 3D graphics using DirectX-compatible graphics hardware.

II. Microsoft Visual Studio 2005 Express Edition Installation

The following steps were used for downloading the software to test the video game's programming:

- 1) Install Visual C++ Express
- 2) Install the service pack for Visual C++ Express
- 3) Install the Microsoft Platform SDK
- 4) Update the Visual C++ directories in the Projects and Solutions section in the Options dialog box
- 5) Update the core_winexpress.vsprops file
- 6) Generate and build a Win32 application to test your paths
- 7) Click START button in the IDE

III. Accomplishments

A. Penguin Behavior

Penguin behavior was a major factor in producing a realistic game. In order to make the game realistic for middle school students the behaviors of penguins had to be researched; their walking and running speeds as well as appearance. For the reason that penguins are small animals, they take short steps or hops using their tails

to assist themselves on steep climbs. The maximum walking speed for the average penguins is 3.9 kph (2.4 mph). Penguins are however at their fastest when they are lying on their stomachs and pushing themselves with their feet. This is a form of running for penguins, and this occurs when they feel threatened or in danger. With having collected this data on the behaviors of penguins, we then incorporated them into the graphics and programming aspects of the game's reconstruction.



Figure 1



Figures 2/3

B. Graphics

When we first received the game, the penguins and the snow mounds were unrealistic. There are two main components of the game graphics that was in need of change, that is: the penguins and the snow mounds. The snow mounds were the easiest part because there was only one image that needed to be changed, whereas the penguins had twelve different images that needed to be recreated (see Fig 2).

The first image that was changed was the "tombstone" snow mounds. They had to be turned into igloos (see Fig 3) because it seemed easier to recognize than snow mounds. The next thing was to analyze the different images for the penguins, there are two groups of the images: tagged and untagged, and inside those two categories are three types of actions: walking/waddling, panicking, and sliding. Of course there is also a direction for each of those, left or right, because if there were no direction the penguin figures would have another penguin

figure facing left, but sliding right. The major focus was to; first recreate the untagged walking penguin because we could then modify that into each of the other images that we needed. Designing the igloos was the easy part because it took very little research to figure out what one looked like, the only thing hard about it was choosing the colors, which in itself is easy enough. However, the penguins were not so easy because we wanted them to look fun, but also serious.

We started off by searching the web for realistic images of different penguins, that way we would have an idea of what the penguins in our game should look like. Once that was figured out we began to design the untagged left walking penguin. After that all there was to do was simply flip it horizontally to get the untagged right walking penguin. (See Fig 4).



Figure 4



Figure 5

The untagged left panicked penguin is the same as the untagged left walking penguin, except for outstretched arms and (unrealistic) pink blotchy cheeks. With the same process for the untagged right walking penguin, the untagged right panicked penguin is just the left horizontally flipped (see Fig 4).

The next step was to rotate the untagged left walking penguin 90 degrees counterclockwise. After that the heads and feet had to be reconstructed. Once that was done the untagged left sliding penguins were finished, and then once it was flipped horizontally so was the untagged right sliding penguin (see Fig 4).

Finally all there was to do was go back into the original game and put a “tag” on all of the penguins to create all of the tagged versions of their untagged counterparts (see Fig 5).

C. Programming

By examining the code, the team realized that a copious amount of data was collected. Featured below is a sample of the original code before any modifications (Figure 1). The penguins consisted of three different rows. There is the front, middle, and back row. Each of these rows is behaving differently. The penguins could be walking, either panicking, or sliding. The code below is where the team’s programmer decided which row had what type of behavior.

```

{
    if(g_bFromLeft)
    {
        m_kFrontRow.ChooseNextPenguin(PenguinRow::LEFT_SIDE,
            Penguin::WALK);
        m_kMiddleRow.ChooseNextPenguin(PenguinRow::LEFT_SIDE,
            Penguin::WALK);
        m_kBackRow.ChooseNextPenguin(PenguinRow::LEFT_SIDE,
            Penguin::WALK);
        g_bFromLeft = false;
    }
    else
    {
        m_kFrontRow.ChooseNextPenguin(PenguinRow::RIGHT_SIDE,
            Penguin::WALK);
        m_kMiddleRow.ChooseNextPenguin(PenguinRow::RIGHT_SIDE,
            Penguin::WALK);
        m_kBackRow.ChooseNextPenguin(PenguinRow::RIGHT_SIDE,
            Penguin::WALK);
    }
}

```

Figure 1

The game was presented to the team as a preliminary document, in need of upgrading. We engaged into a team meeting and addressed what aspects of the game required modifying. We reached the conclusion that the game’s weaknesses included its graphics and the penguins’ movement and speed. When the preliminary document was first presented to the team, we all observed that the graphics had great potential to become more realistic. The motive of our game setting is imperative because that conveys the information being taught to our game players. At first, the setting contained mounds that resembled tombstones. In addition to the tombstones, the penguins were just moving circular objects on a screen. As it was previously stated, the Arctica team was striving for a realistic setting. Therefore, our graphic designer customized the images. After we modified the penguins, we began to take a closer look at their movement. One team member researched data on penguins, and realized that the original walking speed of the penguins were not penguin-like. In response, the team’s programmer altered the speed, which contributed to the team’s overall objective.

The code below displays the speed, the number of penguins in each row, the penguin’s behavior,

and the number of shots the player has to tag the penguins (Figure 2). The number of penguins can be altered along with the number of shots the player has.

```

{
    pkCurrentPenguin =
    m_kMiddleRow.GetMovingPenguin(i);

    // check if the current penguin is tagged
    if(pkCurrentPenguin->IsTagged())
    {
        // stop the penguin
        pkCurrentPenguin-
        >SetMovement(Penguin::STOP, Penguin::MOVE_LEFT);
    }
}

```

Figure 2

III. Conclusion

The work that was completed on this project served to enhance an existing educational program with biologically inspired graphics and complete basic user testing.

IV. Future Work

The next teams of developers, at the University of Kansas, now have a stepping-stone to build on with which to teach the next generation of scientists.

V. References

- [1]. <http://www.seaworld.org/infobooks/Penguins/behavior.htm>
- [2]. <http://www.asoc.org/general/penguin.htm>
- [3]. Giant penguins once roamed prehistoric Peru (2007). http://www.usatoday.com/tech/science/discoveries/2007-06-25-giant-penguins-peru_n.htm
- [4]. Microsoft Visual C++. (2007). <http://msdn.microsoft.com/vstudio/express/visualc/default.aspx>
- [5]. Microsoft Platform SDK. (2007). <http://www.microsoft.com/downloads/details.aspx?FamilyId=0BAF2B35-C656-4969-ACE8-E4C0C0716ADB>

```

{
const I32 PenguinGameApp::NUM_FRONT_ROW PENGUINS = 10;
const I32 PenguinGameApp::NUM_MIDDLE_ROW PENGUINS = 10;
const I32 PenguinGameApp::NUM_BACK_ROW PENGUINS = 10;

// initialize the penguin speed constants
const F32 PenguinGameApp::FRONT PENGUIN_SLIDE_SPEED =
0.015f;
const F32 PenguinGameApp::FRONT PENGUIN_PANIC_SPEED =
0.01f;
const F32 PenguinGameApp::FRONT PENGUIN_WALK_SPEED =
0.05f;
const F32 PenguinGameApp::MIDDLE PENGUIN_SLIDE_SPEED =
0.09f;
const F32 PenguinGameApp::MIDDLE PENGUIN_PANIC_SPEED =
0.06f;
const F32 PenguinGameApp::MIDDLE PENGUIN_WALK_SPEED =
0.003f;
const F32 PenguinGameApp::BACK PENGUIN_SLIDE_SPEED =
0.08f;
const F32 PenguinGameApp::BACK PENGUIN_PANIC_SPEED =
0.04f;
const F32 PenguinGameApp::BACK PENGUIN_WALK_SPEED =
0.002f;

// initialize the starting number of tags (shots)
const I32 PenguinGameApp::INITIAL_NUM_TAGS = 15;
}

```

Figure 3

Also featured in the code is the penguin's behavior after being tagged. When tagged, the penguin is uniquely marked with a blue spot. Each row of penguins reacts differently when tagged. Some stop, others move left, and some move right (Figure 3). The game ends when all of the penguins are tagged.