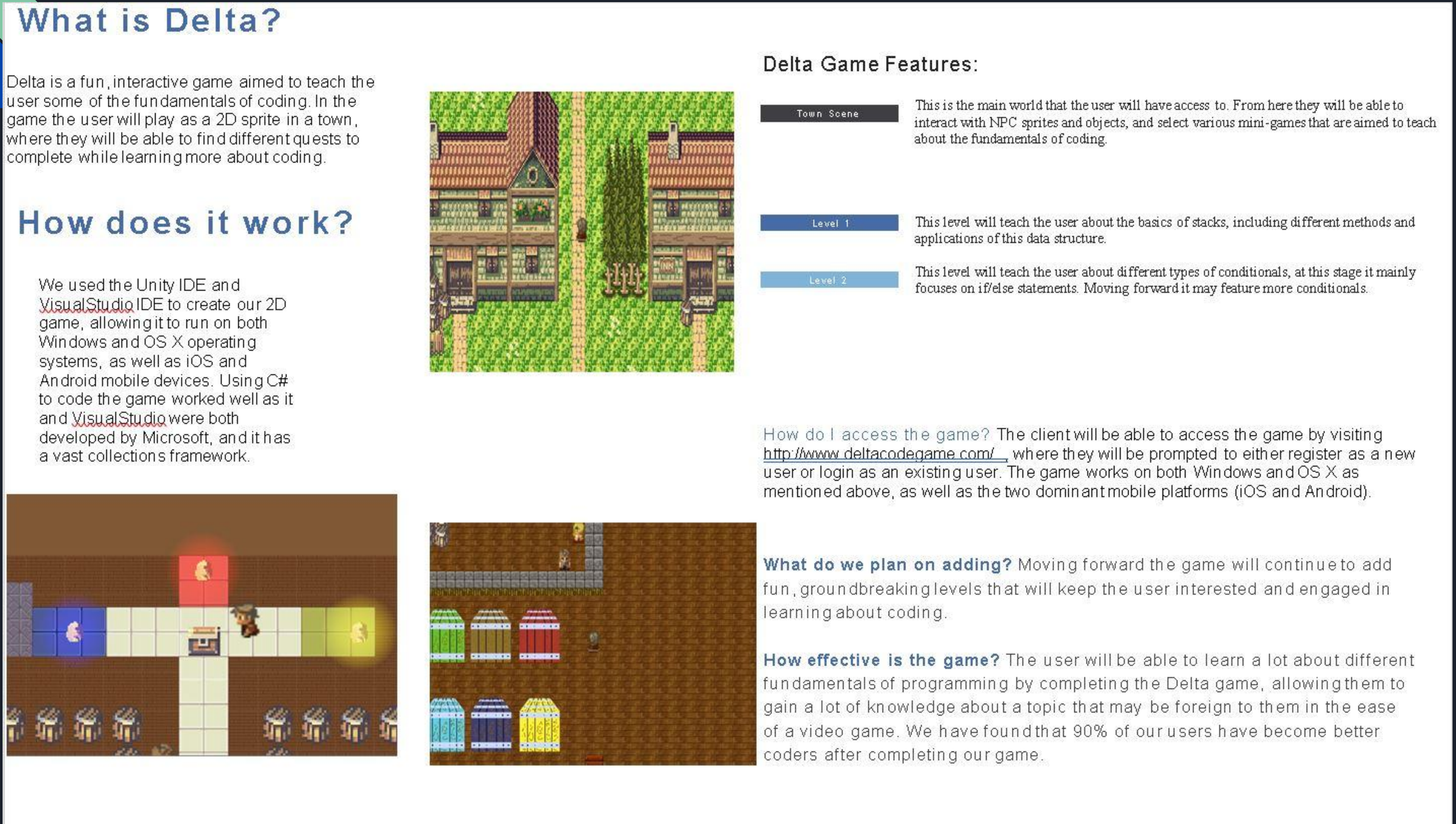
Delta: Introduction to Programming

Developers:
John Crowley
Peter Julian
Chris Kelly
Matt Richardson
Nick Zazula

Agenda

- Brochure
- Purpose
- Implementation IDE & Implementation Language
- Game Structure
- Scripts
- User Guide
- Demo
- Questions/Closing Comments

Brochure



What is Delta?

Delta is a fun, interactive game aimed to teach the user some of the fundamentals of coding. In the game the user will play as a 2D sprite in a town, where they will be able to find different quests to complete while learning more about coding.

How does it work?

We used the Unity IDE and VisualStudio IDE to create our 2D game, allowing it to run on both Windows and OS X operating systems, as well as iOS and Android mobile devices. Using C# to code the game worked well as it and VisualStudio were both developed by Microsoft, and it has a vast collections framework.

Delta Game Features:

Town Scene This is the main world that the user will have access to. From here they will be able to interact with NPC sprites and objects, and select various mini-games that are aimed to teach about the fundamentals of coding.

Level 1 This level will teach the user about the basics of stacks, including different methods and applications of this data structure.

Level 2 This level will teach the user about different types of conditionals, at this stage it mainly focuses on if/else statements. Moving forward it may feature more conditionals.

How do I access the game? The client will be able to access the game by visiting http://www.deltacodegame.com/ , where they will be prompted to either register as a new user or login as an existing user. The game works on both Windows and OS X as mentioned above, as well as the two dominant mobile platforms (iOS and Android).

What do we plan on adding? Moving forward the game will continue to add fun, groundbreaking levels that will keep the user interested and engaged in learning about coding.

How effective is the game? The user will be able to learn a lot about different fundamentals of programming by completing the Delta game, allowing them to gain a lot of knowledge about a topic that may be foreign to them in the ease of a video game. We have found that 90% of our users have become better coders after completing our game.

Purpose

- Teach users the fundamentals of programming across the various programming languages
- In the game, we will attempt to teach the user about control statements, sorting algorithms, data structures, and the importance of commenting. We will tackle this challenge by presenting the user with various challenges to complete, while also explaining how these relate to the world of programming.

Implementation IDE

After being assigned a game to teach users the fundamentals of programming, our team had to decide how we would implement the software. We chose to use the Unity Game Engine. Unity allows the programmers to create a 2D or 3D game, depending on the end goal of the project, across various platforms. This is one of the main reasons why we chose to implement the Unity IDE. For example, with only a few minor changes to our program, our game will be able to run on both OS X, Linux, and Windows computers and on mobile devices with iOS and Android operating systems. Another advantage to using Unity is the fact that it has its own Asset Store. Here we had access to various free and/or cheap components for our games, including sprites and other art. Finally, for our purposes Unity was free to use, which was also a big plus. Each of these facts made our decision easy, as the Unity Game Engine would allow us to design and implement our game with relative ease.

While the Unity IDE was focused more on the graphics and visual aspects of how the game functioned, we would still need an IDE to write the necessary scripts for our project. Based on the fact that we were planning on using Unity, this decision was extremely easy to make. When downloading Unity onto one’s computer, there is also a prompt that asks if he or she would also like to download Visual Studio, as the two work hand in hand with each other. Therefore, we chose to implement our code using the Microsoft Visual Studio IDE.

Implementation Language

Based on our decision to use the Unity IDE and Visual Studio IDE, we decided to implement the C# programming language. We all had experience with Java before this class, so we believed that C# would be relatively easy to learn based on this knowledge. In addition to this, Visual Studio and C# were both developed by Microsoft, so they would be extremely compatible with one another. Another reason we chose to implement C# was due to its vast Collections Framework, which we implement in nearly every script in our project. Finally, of the programming languages that the Unity Game Engine successfully interacts with, C# gave us the most flexibility with how we wanted to create our game.

Game Structure

All of the files that we used to develop the game can be found in the “Assets” folder in the main project file. “Assets” contains all of the folders containing the files we have created to run the game. The four main folders that we used were the “Prefabs,” “Scenes,” “Scripts” and “Sprites” folders. The “Prefabs” folder contains our saved game objects. The “Scenes” folder contains the graphics for each of the game levels we have created so far. Currently, we have a *HomeTown* scene, *Start* scene and scenes for the first three levels .The “Sprites” folder contains images that we used to create the scenes for the game. This include the background and characters in the game. The “Scripts” folder contains all of the files that make the game run. The main folder contains the scripts that control how the character and camera move, the *LevelManager* script, which opens the level connected with a certain building in the Hometown, and two dialog controlling scripts. Finally, in this are two subfolders for levels 1, 2 and 3.

Scripts - General

- AvatarController
- CameraLock
- DialogHolder
- DialogManager
- LevelManager

Scripts - Level 1 (Conditionals)

- ChestInteraction

- EntryDialog1

- ExitToTown

- LightSwitcher

- TownExit

Scripts - Level 2 (Arrays)

- ArrayController

- ArrayManager

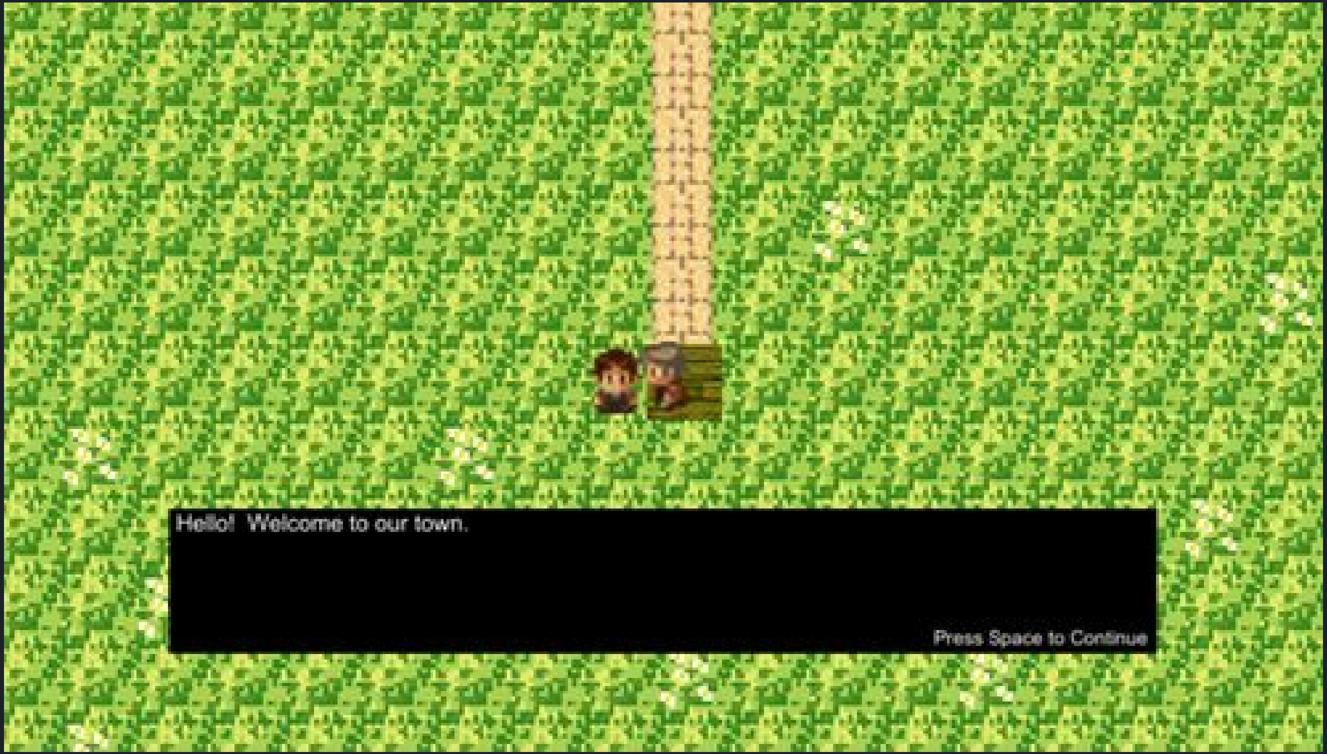
- Exit

Scripts - Level 3 (Stacks)

- BoxController

- BoxManager

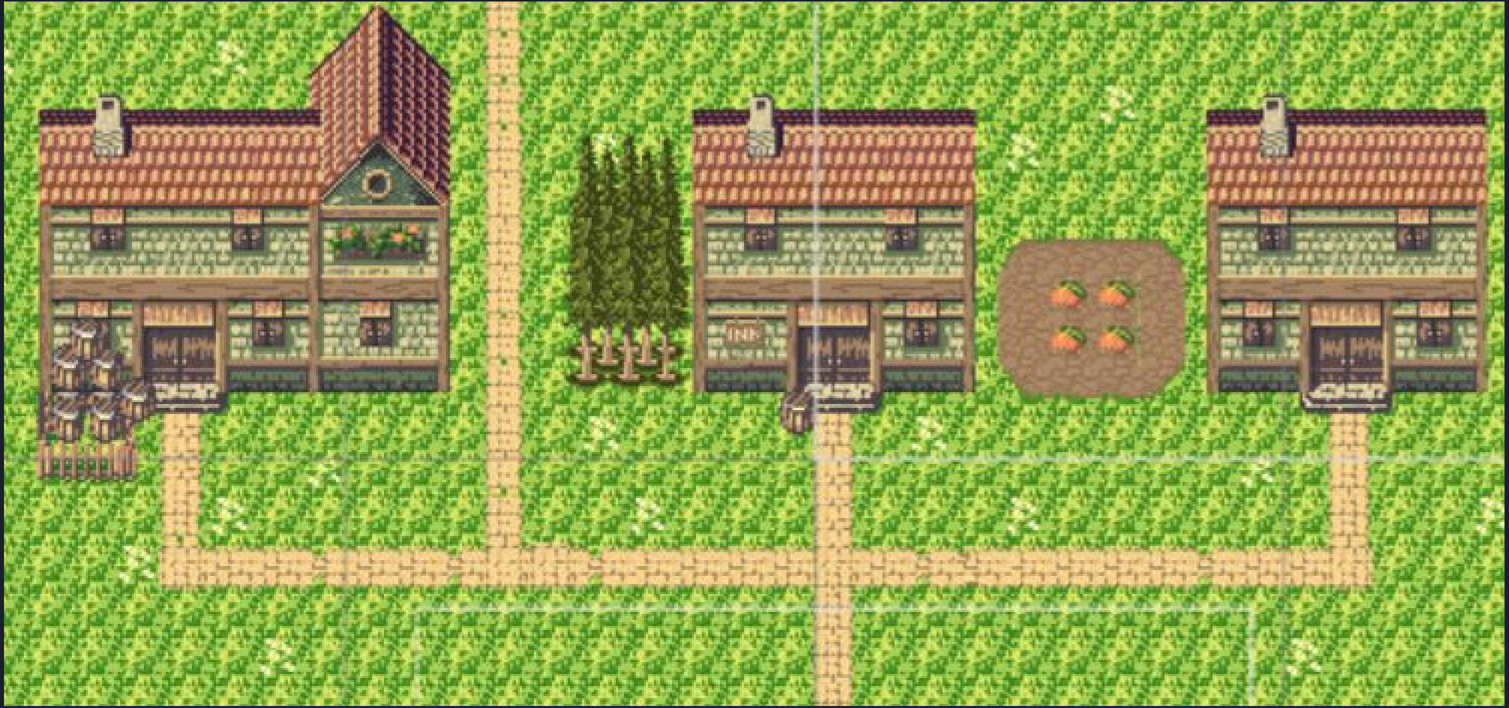
User Guide



Hello! Welcome to our town.

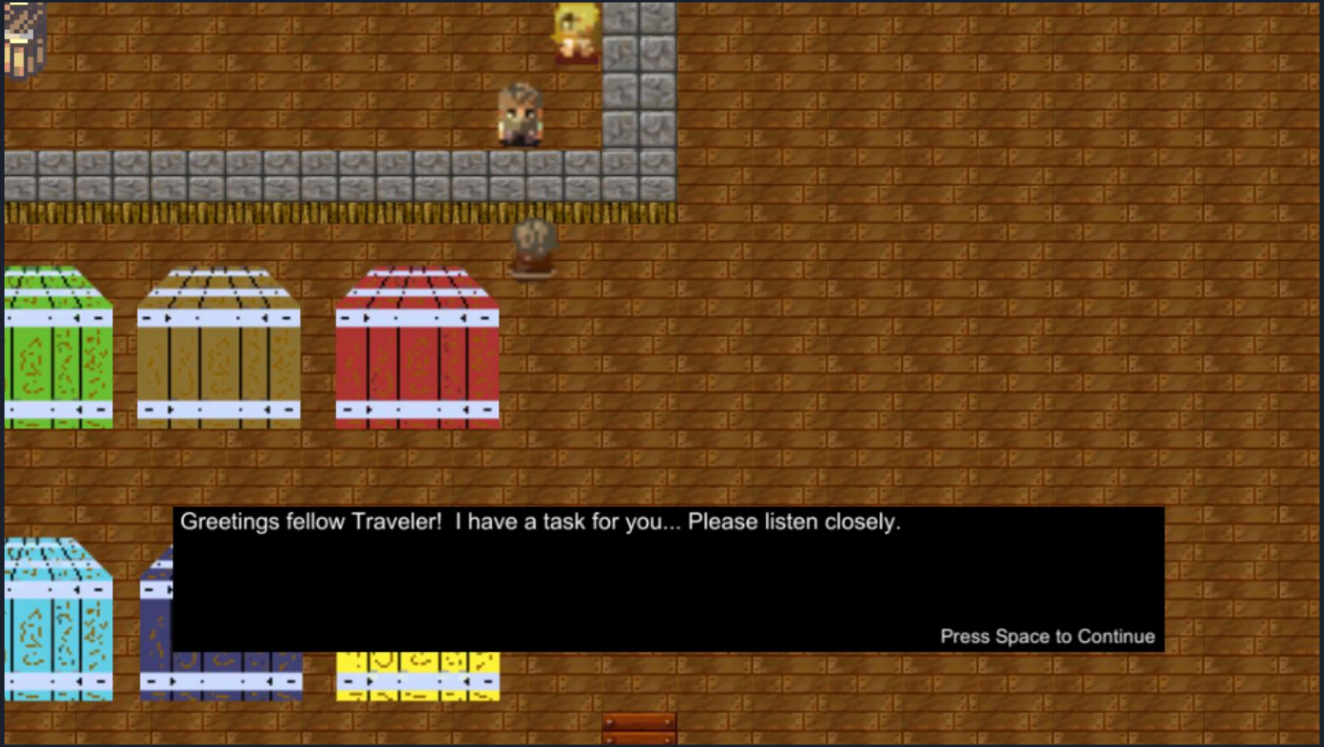
Interact with characters by pressing enter. Move around the town with arrow keys

User Guide



Play one of our levels by walking over to the door of a house

User Guide



Greetings fellow Traveler! I have a task for you... Please listen closely.

Upon entrance into a house, you will find a mini-game and instructions to play

Stand by for Game Demo

Questions?

Thank You!

Happy Holidays!