Everybody Fits In

- Technical Risk Assessment -

Andrew Rimpici and Alexander Hubble

- EGD-220-04 -

TABLE OF CONTENTS

DIFFICULTY SCALING SYSTEM

Document Scaling System

Scaling Overview

- This document uses a 1 through 5 difficulty scaling system where 1 represents very easy, and 5 represents extremely difficult.

Difficulty Representation

- The difficulty will be represented using puzzle pieces shown below:

●○○○○	**Represents 1 out of 5** Very Easy - No trouble to implement
●●○○○	**Represents 2 out of 5** Easy - No trouble to implement but may take more time than usual
●●●○○	**Represents 3 out of 5** Medium - More thought than usual went into planning and implementing
●●●●○	**Represents 4 out of 5** Moderately Difficult - A lot of planning and time went into implementing
●●●●●	**Represents 5 out of 5** Very Hard - Very difficult to plan and to implement. Most likely would take too long to implement before a deadline

THE DELIVERY PLATFORM

Main Delivery Platforms

Web and Tablet -

- *Everybody Fits In* is primarily designed to be developed and played on computers and tablets on a website or downloadable application.
- *Everybody Fits In* is aimed to be a learning tool targeted for schools to use in the classroom and since most modern American classrooms now have Chromebooks or Apple iPads, it would be smart to develop the game for the web.
- The simple drag-and-drop nature of the game makes it easy for the programmers to adapt the application for either tablet or computer based systems, and having it run on the web allows for all device types to run the game with little setup time.

THE DEVELOPMENT ENVIRONMENT

Programming Environment with Unity

Using The Unity Engine -

- The game will be developed in the Unity Game Engine. Unity comes with a lot of built-in features that are right at the hands of the developer's. Having access to these features eliminates the need to construct a game engine tailored to the game from the ground up, thus *saving a great deal of time*.
- A majority of the team is comfortable using the Unity Engine which translates to a more efficient workflow.
- Unity is also extremely useful because it allows for cross-platform development.
    - This comes in handy as the developers can simply export the game to a web, mobile, or desktop version of the game with little overhead—*saving time and money*.

THE DEVELOPMENT ENVIRONMENT
(Continued)

Resource Environment

Adobe Photoshop - ★★☆☆☆

- All of the art assets used for the game will be created in the Adobe Photoshop art program. Photoshop is a very widely known and well supported graphics tool that allows for users to create things of all shapes and sizes. It can be used to draw or edit photos. The program mainly works in a raster format and not vector format which works perfectly for *Everybody Fits In* as the assets will be pixel based and not vector based.

THE DEVELOPMENT ENVIRONMENT
(Continued)

Version Control

Subversion **-** ●●○○○

- The choice of version control for this project is Subversion (SVN). In Pineapple, there is a subversion repository linked where all of the team members can monitor and update the game files. The use of Subversion will help the team make sure all of the game files are up-to-date and where they should be in regard to meeting deadlines.

- In the repository on pineapple there is a root folder called “Unity Tree” that holds all of the different branches and builds for the game. This makes it more convenient to pinpoint certain builds that might have bugs in them in addition to keeping backups and snapshots of different build milestones.

GAME MECHANICS AND SYSTEMS

Game Systems

CutScenes -

- Before each level there will be a cutscene showcasing the current situation and the objective necessary to complete that level. The cutscene consists of a few slides that can be navigated through by pressing the space key. Once all of the slides in the cutscene are shown, the level is started and it is up to the player to satisfy all of the requirements for that level.

Level Management -

- All levels are loaded using a single list attached to the Game Manager Prefab. This list contains the names of the levels in the order they will be played. As long as the level is loaded up in the list, the controller will automatically move to the next level and cutscene.

Win detection - ●●○○○

- The goal of each level is to make the subject happy by either surrounding him/her with classmates (in the case of the first level) or moving him/her to a position in the classroom that they would be happy in; all while fitting in the other classmates to the level as well.
- Once all of the classmates are successfully placed correctly in the level, the players will unify together in the form of a rainbow and the level ends.

Sound system- ●●○○○

- The sound system is a singleton gameobject that controls all sound within the game. Within the object there are two lists of sounds, one for music and one for audio fx's. To play a song, simply type the name of the song into the global variable on the sound system object. As long as this song is within the list, it will be played. It is also important to note that the song can be played / swapped at anytime by calling the setSong function. Any fx from the list can also be played at anytime by calling the playFx function, as long as the passed name is within the list it will be played.

Game Mechanics

Drag and Drop (control scheme) - ★★☆☆☆

- The players can drag the pieces/classmates around the level using either their mouse or finger depending on which platform they are using. Reasons to use a drag and drop control are as follows:
  - It's easy for elementary level children to use.
  - It can be implemented easily on devices such as computer and tablets.
  - It's easy to implement within the game itself.

GAME MECHANICS AND SYSTEMS
(Continued)

Snap to Spot - ●●●●○

- The Snap to Spot mechanic makes it easier for players to verify the piece they just placed is secured in a spot in the level. This mechanic serves two primary functions within the puzzle:
    - To help younger players fit pieces in accurately. Once the held piece is dropped and is within a snap spot, it will snap to that spot in the level.
    - To help check the win state. Upon a piece snapping to a spot, the game checks to see if all pieces are snapped to a correct spot.

Rotation - ●●●○○

- With any polygon shaped pieces, the player is able to rotate by hovering over and either pressing the spacebar, right clicking, or tapping the screen with a second finger if on a mobile device.

GAME MECHANICS AND SYSTEMS
(Continued)

Pre-snapped/Anchor Pieces -

- To help the understanding of the purpose of the game, some of the "Special Needs" pieces will be automatically snapped into the board. Anchor pieces cannot be moved by the player and are merely there to form challenges and help move the player in the correct direction towards the correct piece layout.

THE ART PIPELINE

Creating the Art

- The pieces in *Everybody Fits In* are scaled on a 100x100 scale, with the base scale being a 100x100. What this means is that for any piece the scale can be figured out by the shape of the piece. For instance if we need a rectangle, the dimensions would simply be found by adding 100 in whatever direction the rectangle is facing, so in this example it would be a 100x200 piece. Any non-square objects will always fit within a 100x100 canvas. For any odd pieces (such as the L and Z pieces) the canvas will be an even scale, however any unused space should be transparent.

Using the Repository

Step 1 - Download TortoiseSVN

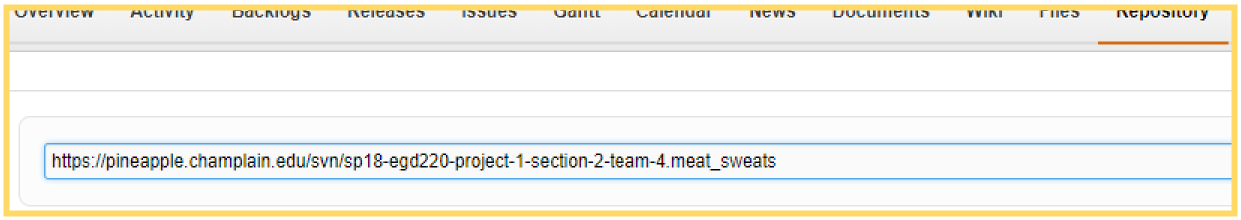
- The first step is to download TortoiseSVN. TortoiseSVN is a client software that allows the user to upload documents and files to a server for other members with access to view and manipulate. TortoiseSVN is free, easy to learn, and easy to use.

THE ART PIPELINE

(Continued)

Step 2 - Navigate to Pineapple

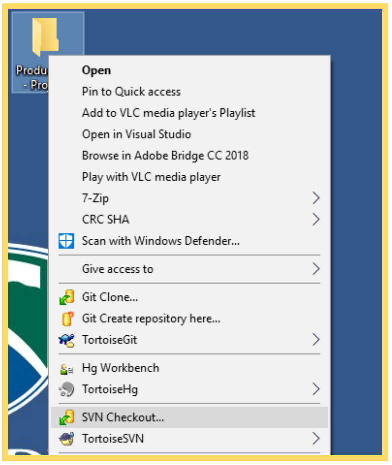
- Once downloaded, the user should log into Pineapple and under the team's "Repository" tab, there will be a URL.



https://pineapple.champlain.edu/svn/sp18-egd220-project-1-section-2-team-4.meat_sweats

Step 3 - Create TortoiseSVN workspace folder

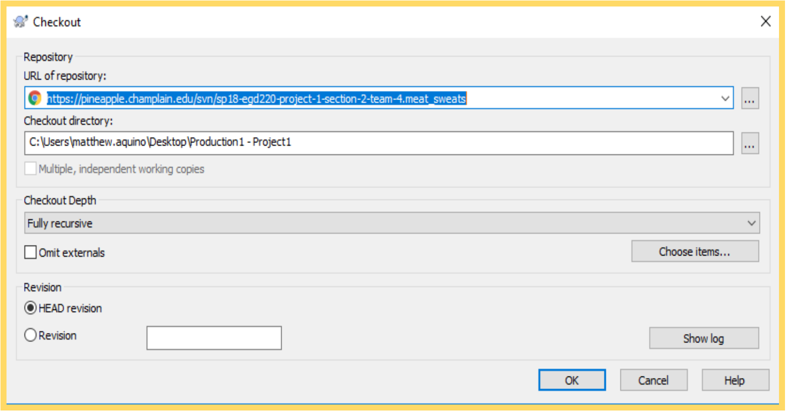
- The user should copy the URL to the computer's clipboard by using the hotkey CTRL + C. Once that is done, the user should create a folder on their desktop or other location of choice. Right-click on the folder and choose the option called "SVN Checkout."


Open
Pin to Quick access
Add to VLC media player's Playlist
Open in Visual Studio
Browse in Adobe Bridge CC 2018
Play with VLC media player
7-Zip
CRC SHA
Scan with Windows Defender...
Give access to
Git Clone...
Git Create repository here...
TortoiseGit
Hg Workbench
TortoiseHg
SVN Checkout...
TortoiseSVN

THE ART PIPELINE
(Continued)

Step 4 - Paste Pineapple URL in TortoiseSVN

- The URL from the previous step should automatically be placed in the window that pops up. The user should click "OK" and log in with their Pineapple credentials when prompted. This will load all folders in the repository into the folder they've created.


Checkout

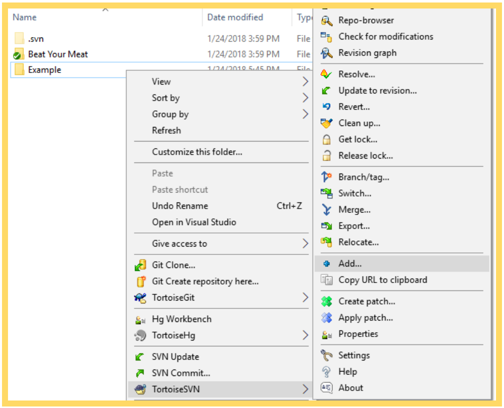
Repository
URL of repository: https://pineapple.champlain.edu/svn/sp18-egd220-project-1-section-2-team-4.meat_sweats
Checkout directory: C:\Users\matthew.aquino\Desktop\Production1 - Project1
☐ Multiple, independent working copies

Checkout Depth
Fully recursive
☐ Omit externals

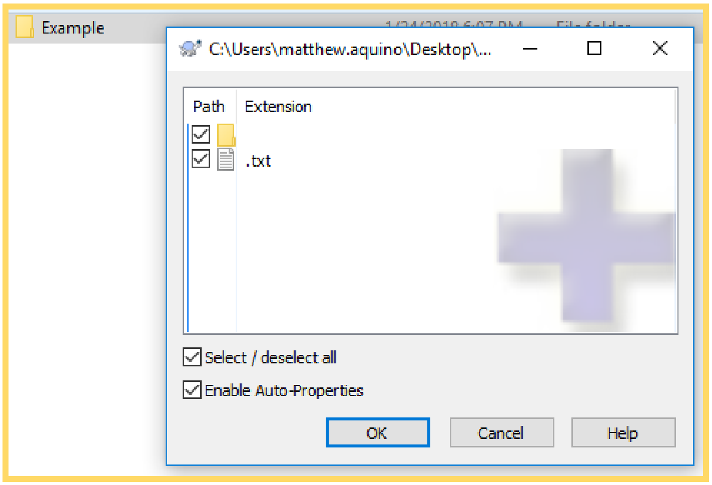
Revision
● HEAD revision
○ Revision

Step 5 - Add Files/Folders to the Repository

- If the user wishes to add a new folder to the repository, all they simply have to do is copy the new folder into the repository folder, right-click on it, highlight the "TortoiseSVN" option, and select "Add."


Name	Date modified	Type
.svn	1/24/2018 3:59 PM	File
Beat Your Meat	1/24/2018 3:59 PM	File
Example	1/24/2018 5:45 PM	File

- Select the folders you wish to add, or choose specific files to add, and click “Ok.”



Step 6 - Commit New Files to the Repository

- Finally, simply commit the folder by right-clicking on it, and selecting “SVN Commit...”. Click “Ok” and the folders should be uploaded to Pineapple. It is very important to always update and commit any new versions of work so that the team is always up-to-date.

Loading the Art in Unity

- The art will be loaded directly in by the designers and / or programmers once the art has been uploaded into the repository. For any odd shapes, the collider will have to be manually set using the polygon collider.

THE DESIGN PIPELINE

Unity Engine

Implementing Puzzles -

- Designers are able to implement new pieces into a level by dragging in a pre-existing gameobject known as a snap-spot or AOE (Area of Effect) object. These objects act as hotspots where if an applicable piece touches it, the piece will snap to that specific spot.
- When a designer places it into the level the piece is automatically registered for that level without the designers needing to worry.

Adding New Pieces -

- To add in new moveable/anchor pieces, designers can drag in pre-existing game objects known as pieces. These objects are automatically registered by the level and are kept track of by the Level script. To transform a piece into an AOE, the designers simply toggle a checkbox in a script in the Unity Editor. Once all of the pieces are correctly snapped to a snap-spot/AOE then the level ends.

Design pipeline
(Continued)

Rotating in Engine -

- Each piece and snap-spot/AOE can be rotated by the designers thanks to a script in the editor. This script allows the designers to pick which way a piece should be rotated in order for that piece to snap to an AOE, and allows the designers to pick which rotation a piece starts out in at the beginning of a level.

Editable Components

Prefabs for Designers	
Pieces	**AOEs**
Editables - Shape Rotation Script - Lets designers pick between North, South, East, West - Piece Script - isAnchor checkbox left empty - Shape Scale Script - Scales the shape	**Editables** - Shape Rotation Script - Lets designers pick between North, South, East, West - Piece Script - isAnchor checkbox checked - Shape Scale Script - Scales the AOE

Milestone Updates

Milestone #1

Deliverables:

Artist Concepts

- Three styles for possible art direction were created.

Visual Design Document (VDD)

- Guide for core mechanics.

Game Rules Document

- Rules pertaining to the game were outlined.

Physical prototype

- Board was created and tested.

Goals for Next Milestone:

- Create a functioning digital prototype that effectively conveys the game's core mechanics

Milestone Updates
(Continued)

Milestone #2

Deliverables:

Digital prototype

- The digital prototype is functioning, however there is no way to make multiple levels.

Art Style

- Solidified the art style for the game

Technical Plan

- The technical plan has been started and is currently being iterated upon.

Goals for Next Milestone:

- Simple cutscenes
- Code in multi-level functionality
- Addition of more levels
- Functionality to unplace puzzle pieces that have already been placed on the board

Milestone Updates
(Continued)

Milestone #3

Deliverables:

Digital Prototype

- The digital prototype has been updated with more polish, art assets, and functionality.

Technical Plan

- The technical plan has been cleaned up and iterated upon.

Goals for Next Milestone:

Final build

- The next Milestone will include our final build of the game, with an optimal amount of polish and assets.

Milestone Updates
(Continued)

Milestone #4

Deliverables:

Digital Prototype, Final build

- The digital prototype should be a functional beta, showing off polished gameplay

Technical Plan

- The technical plan has been iterated upon and cleaned up to represent the current progress of the digital prototype.