Johnson, Alex
Neace, Anthony
Shiner, Hillary

XNA Final Project
Spring, 2013
Group 9

# Hippotastic: Hippo Simulator 1.0

Our team developed a “Virtual Pet” game for the XNA Final Project called *Hippotastic*. Upon creating a new game, the player will be given a hippopotamus as their virtual pet. The player will then spend the rest of the game nurturing and interacting with their pet by participating in mini-games that reward the player with points to support their pet.

**Compile Instructions**

- Must be using at least “XNA 4.0 Refresh” to compile correctly -- XNA does not correctly handle multiple .FBX animations associated with a single model before this release.
- SkinnedModelPipeline DLLs in this package are compiled for use with a 64-bit machine.

**Project Management**

- Repository
  - Our team is using GitHub via Git as a repository for our project.
  - Location: https://github.com/AnthonyNeace/xna-final-project
- Communication
  - Our team is using Trello to communicate goals and progress.
  - Location: https://trello.com/board/xna-final-project/5163312653beb3df0e0095c4

**Game Contributions and Technologies Used**

- Alex developed the user interface and environment for the game.
- Anthony developed the hippopotamus model, textures, and animations, as well as the code to support these in the project.
  - The hippo model was developed in Blender for use with Microsoft’s skinned model pipeline in XNA. A bone apparatus was developed in conjunction with the hippo’s geometry to allow us to quickly prototype and experiment with animations, and then exporting them to XNA.
  - A sample of purely the hippo working with the Skinned Model Demo is available on our GitHub.
- Hillary developed the mini-games to be used during gameplay, the scoring system and the save game feature.
  - The mini-games are 2D games using sprites and mouse event logic to implement buttons. The scoring system is used globally through all the minigames in the project so ensure it is always up to date. The way points are awarded is always to favor the user by awarding more points for winning than deducting for losing. The

save game feature uses a text file that stores the state of the game so that things like the score, the hippos happiness level and the hippos hunger level will resume with the same values as when a player saved the game and exited last. It is done by writing this data to a text file when the user clicks the save button on the homepage and it will also automatically write the data to the file if the user just exits and forgets to save. When the user loads the game the next time to resume playing the text file will automatically be read in and set all the values to the values saved.

**Additional Technologies Used**

- Blender 2.6 is a modeling software that we used to develop, animate, and texture the hippo model. A UV map was exported from blender and quickly modified in Paint.Net to generate textures. There was some difficulty between models because Blender can export .FBX files, but it cannot import them. This presented some challenges if I, for example, forgot to save a .blend and .FBX version of the model.
- Sound effects were added following the model provided by our “Learning XNA 4.0” textbook. We added sounds that played while hovering over or clicking on a button, as well as several background tracks.
- Trello and GitHub -- while not directly related to the code -- were very helpful in managing our time and resources while coordinating on the game. They both helped keep us up to date on our other team member’s efforts and progress.

**Code Structure**

1. Character Customization

    a. Player begins with the default hippopotamus pet.

    b. Player can change the color of their pet model by earning points and purchasing a new color through the accessories option in the home screen

        i. Three different color options

            a. blue, camo, pink

        ii. The pets happiness will increase to 100% when the player purchases a new color for them

2. Pet Interaction Lobby

    a. Description: the game will begin in the “Lobby Game State”, where UI buttons appear allowing the player to interact with their pet. Here the player can access minigames, save their game, purchase food or purchase model colors. The player will return here when loading the game, or completing a minigame.

    b. UI Elements

1. Pet Interaction Buttons: allow the player to perform an action on the pet which should respond with the corresponding animation or sound
    a. “Pet”, “Feed”, "Accessories"
2. MiniGame Accessor: when clicked, the player is taken to the minigame state of the game they chose.
    a. rock paper scissors, card game, matching game

3. MiniGames

a. Description: These are short, trivial games that the player can complete to earn prizes. These prizes can be accumulated over time to purchase new colors for their pet as well as food to nurture them.

b. Player Progression:

i. Enter game, participate depending on gametype

ii. Win:

1. Prize amount awarded depending on gametype
2. Returned to Lobby Game State or play again

iii. Loss:

1. No negative consequence (the casual nature of pet games)
2. No prize awarded
3. Returned to Lobby Game State or play again

**Challenges**

1. .FBX Animation
    a. XNA 4.0 did not support multiple animations on an .FBX file before the "XNA 4.0 Refresh" update for Windows Phone 7.1

2. Rendering bugs in XNA
    a. SpriteBatch resets some variables and we had to manually reset them after every call
    b. BasicEffect didn't render transparent UV textures correctly, used AlphaTestEffect instead.

**References and Content Attribution**:

Code Libraries and Samples:

1. MSDN Skinned Model Sample and Animation Pipeline

   a. http://xbox.create.msdn.com/en-US/education/catalog/sample/skinned_model
   b. Microsoft Permissive Lisence: http://xbox.create.msdn.com/downloads/?id=15
1. How To: Read from a Text File
   a. http://msdn.microsoft.com/en-us/library/db5x7c0d.aspx
2. How To: Write text from a File
   a. http://msdn.microsoft.com/en-us/library/6ka1wd3w.aspx

Sound Assets:

1. loop022.wav
   a. By freesound.org user "bebeto"
   b. Retrieved from freesound.org under the Attribution Noncommercial "Creative Commons" License.
2. button.wav
   a. By freesound.org user “fins”
   b. Retrieved from freesound.org under the "Creative Commons 0" License.
   c. http://www.freesound.org/people/fins/sounds/146718/
3. button.wav (separate soundfx, same filename)
   a. By freesound.org user “fins”
   b. Retrieved from freesound.org under the "Creative Commons 0" License.
   c. http://www.freesound.org/people/fins/sounds/171521/

Image Assets:

1. Iconarchive.com Image Assets
   b. Each of the image assets below has a usage license that can be viewed from the associated URL
   c. http://www.iconarchive.com/show/mammoth-icons-by-fasticon.html
   d. http://www.iconarchive.com/show/fruitsalad-icons-by-fi3ur/apple-icon.html
   e. http://www.iconarchive.com/show/mammoth-icons-by-fasticon/Mammoth-Seated-icon.html
   f. http://www.iconarchive.com/show/ecommerce-business-icons-by-designcontest/alert-icon.html
   g. http://www.iconarchive.com/show/library-icons-by-skuzigraphic/globe-icon.html
   h. http://www.iconarchive.com/show/oxygen-icons-by-oxygen-icons.org/Actions-document-save-icon.html
   i. http://www.iconarchive.com/show/construction-icons-by-mpt1st/paint-bucket-icon.html
   j. http://www.iconarchive.com/show/casino-icons-by-designcontest/Chip-5-icon.ht

ml

k. http://www.iconarchive.com/show/basic-icons-by-pixelmixer/document-icon.html

l. http://www.iconarchive.com/show/snowish-icons-by-saki/Edit-Cut-icon.html

3. http://code.google.com/p/vector-playing-cards/downloads/detail?name=PNG-cards.zip&can=4&q=
4. http://commons.wikimedia.org/wiki/Category:SVG_playing_cards
5. http://wall.alphacoders.com/big.php?i=52026
6. http://wall.alphacoders.com/big.php?i=74956
7. http://all-free-download.com/free-psd/sort-by-newest/page/3/