

Culmination Presentation

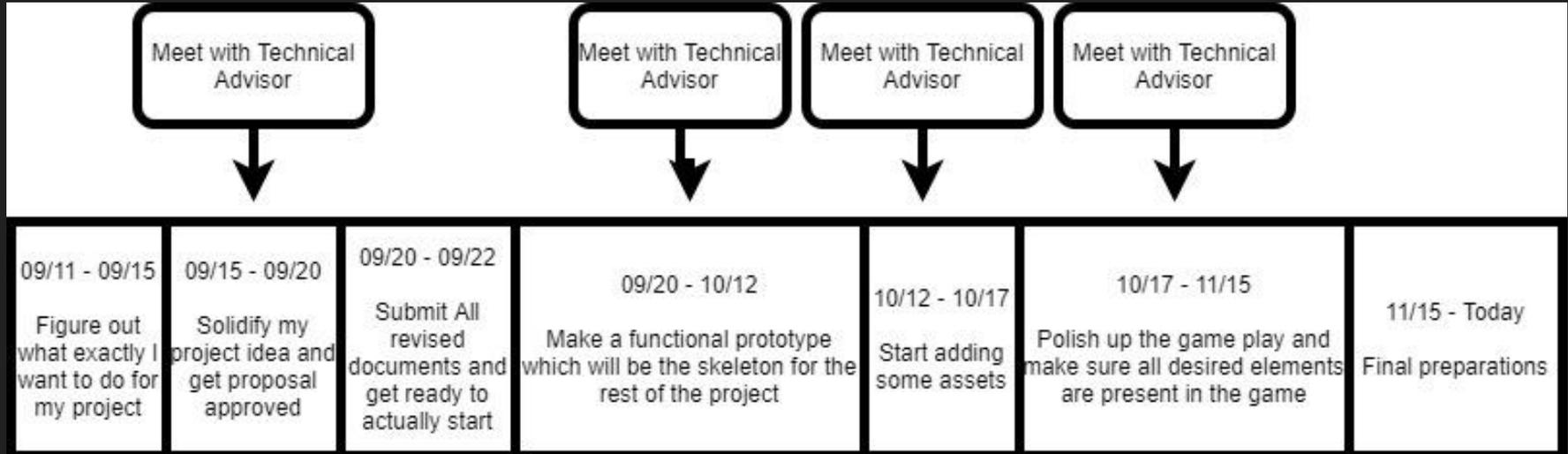
Featuring: “Asteroid Antagony”

By: Giancarlo Macias

Culmination Presentation

Featuring: “Asteroid Antagony”

Schedule/Workflow



What is “Asteroid Antagony”?

Asteroid Antagony is a video game that I have been constructing on Unity since the start of the semester as my culmination project.

What is “Asteroid Antagony”?

In this 2D game, you play as a little spaceship flying through space(ofcourse) while there are many kinds of Asteroids and other projectiles also flying through space(primarily AT you). The goal is to survive as long as possible by dodging anything that comes at you.

What is “Asteroid Antagony”?

Although I started working on this game this semester, the concept for this game has existed for many years. I actually tried to make this game a long time ago way before I knew anything about coding or Unity. This first iteration was called “Meteor Mayhem!” and was made using Android app inventor for use on a mobile android device.

“Meteor Mayhem!” looked like this:



“Meteor Mayhem!” looked like this:

Search

Intro_Screen

Canvas

CheckBox

Clock

Image

Label

ListPicker

PasswordTextBox

Slider

TextBox

TinyDB

Media

Animation

Social

Sensors

Screen Arrangement

LEGO® MINDSTORMS®

Other stuff

Not ready for prime time

AlignVertical
Center

BackgroundColor
White

BackgroundImage
None...

CloseScreenAnimation
Default

Icon
None...

OpenScreenAnimation
Default

ScreenOrientation
Unspecified

Scrollable

Title
Intro_Screen

VersionCode
1

VersionName
1.0

Play_button

Player1

Rename Delete

Media

Astriodforwork.png

Bluecometforwork.png

Cometforwork.png

Non-visible components

Player1

3:09 PM

9:38 PM
12/14/2013

“Meteor Mayhem!” looked like this:

The screenshot shows the MIT App Inventor web interface. At the top, there is a search bar and navigation icons. The main workspace is divided into three sections:

- Component Palette (Left):** Lists various UI components such as CheckBox, Clock, Image, Label, ListPicker, PasswordTextBox, Slider, TextBox, and TinyDB. It also has sections for Media, Animation, Social, Sensors, Screen Arrangement, and other categories.
- Component List (Right):** Shows the components currently on the canvas, including Ship, Asteroid2-5, planet, HorizontalArrangement1, Label2, Label1, Clock1-5, Player1, dieingsound, and labelclock. There are 'Rename' and 'Delete' buttons at the bottom of this list.
- Non-visible components (Bottom):** Shows components that are not currently visible on the canvas, including Clock1-5, Player1, dieingsound, gobackclock, planet, clock, and TextToSpeech1.

The central canvas, titled 'Play_Screen', displays a space-themed game scene. It features a dark starry background, a large planet (resembling Mars) in the upper right, a small ship at the bottom center, and several asteroids at the top. A 'Score: 1' label is visible at the bottom left of the canvas.

On the far right, there is a 'Properties' panel for the selected component, showing settings for 'AlignVertical' (Top), 'BackgroundColor' (White), 'BackgroundImage' (None...), 'CloseScreenAnimation' (Default), 'OpenScreenAnimation' (Default), 'ScreenOrientation' (Unspecified), 'Scrollable' (unchecked), and 'Title' (Play_Screen).

“Meteor Mayhem!” looked like this:

The screenshot displays the App Inventor for Android Blocks Editor interface for a project named "Meteor_Mayhem_Final - Play_Screen". The interface includes a top menu bar with "Saved", "Undo", and "Redo" options, and a toolbar with "New emulator", "Connect to Device...", and "Zoom" (set to 100%).

The left sidebar shows the "Built-In" blocks palette with categories: Definition, Text, Lists, Math, Logic, Control, and Colors.

The main workspace contains several event-driven blocks:

- Play_Screen.Initialize:** Sets planet.Visible to false, Asteroid.Visible to false, Asteroid2.Visible to false, Asteroid3.Visible to false, Asteroid4.Visible to false, Asteroid5.Visible to false, and calls Player1.Start.
- Canvas1.Dragged:** A loop that updates startX, startY, prevX, prevY, currentX, currentY, and draggedSprite, then calls Ship.MoveTo with x=currentX1 and y=350.
- Ship.CollidedWith (Asteroid):** Triggers when the ship collides with an asteroid. It calls TextToSpeech1.Speak (message: "its over dude"), sets labelclock.Timer.Enabled to false, sets gobackclock.Timer.Enabled to true, sets Ship.Picture to explosion.jpg, sets Ship.Height and Ship.Width to 55, sets Ship.Enabled to false, plays dieingsound.Play, and calls Player1.Stop.
- planet.EdgeReached (edge5):** Triggers when the planet reaches the edge. It sets planet.Visible to false and calls open another screen (screenName: "Screen1").
- Asteroid.Timer (Clock1.Timer):** Sets Asteroid.Visible to true, sets Asteroid.X to a random integer from 0 to Canvas1.Width, sets Asteroid.Y to 0, calls Asteroid.PointTowards (target: Ship), and sets Asteroid.Speed to 15.
- Asteroid2.Timer (Clock2.Timer):** Sets Asteroid2.Visible to true, sets Asteroid2.X to a random integer from 0 to Canvas1.Width, sets Asteroid2.Y to 0, calls Asteroid2.PointTowards (target: Ship), and sets Asteroid2.Speed to 15.
- planet_clock.Timer:** Sets planet.X to a random integer from 0 to Canvas1.Width, sets planet.Y to 0, sets planet.Visible to true, calls planet.PointTowards (target: Ship), and sets planet.Speed to 10.
- Label1.Timer:** Reports the boolean false, and a block shows "Label 1 Text" + "number 1".
- gobackclock.Timer:** Calls open another screen (screenName: "Screen1").
- Clock3.Timer:** A timer block at the bottom.

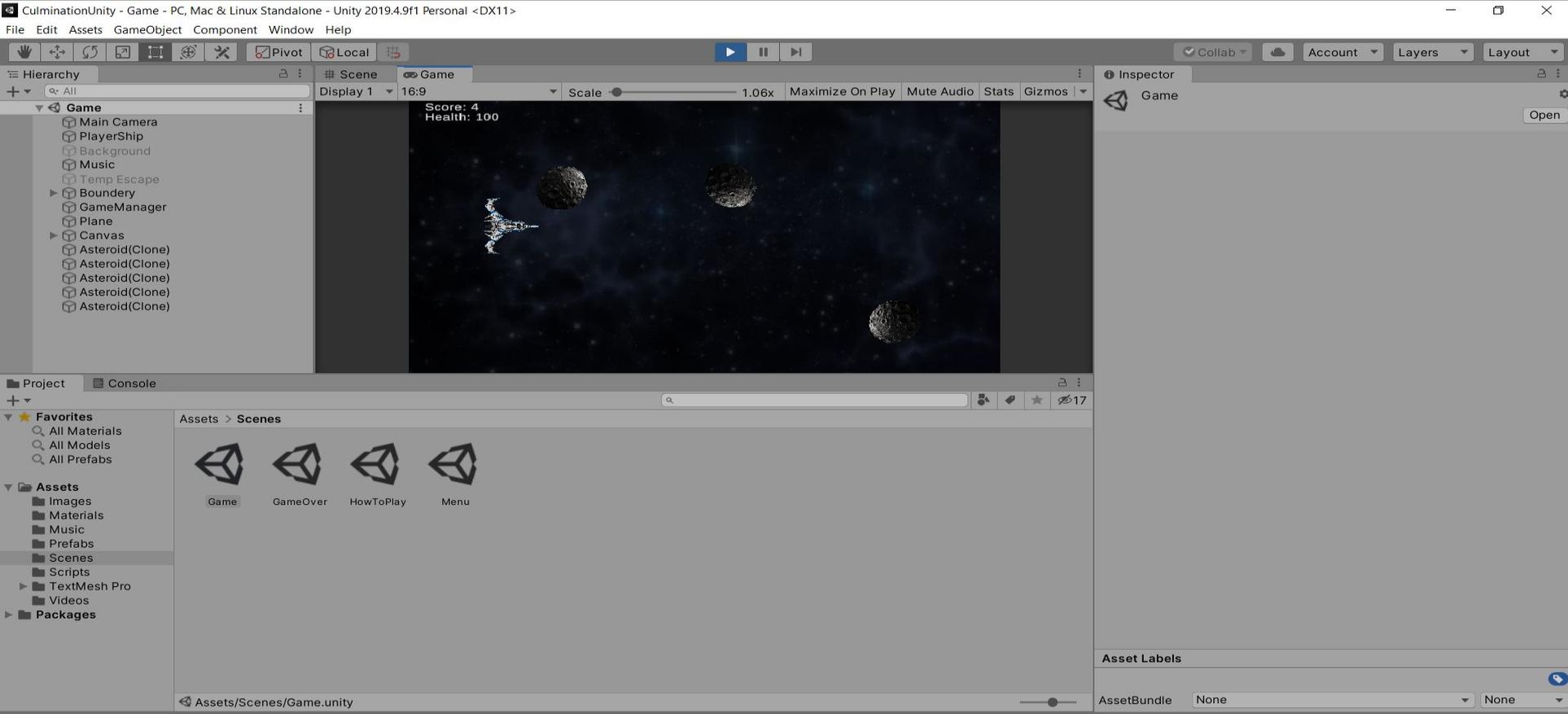
The bottom status bar shows "Built: October 17 2013 Version: v134". The Windows taskbar at the bottom right shows the time as 9:45 PM on 12/14/2013.



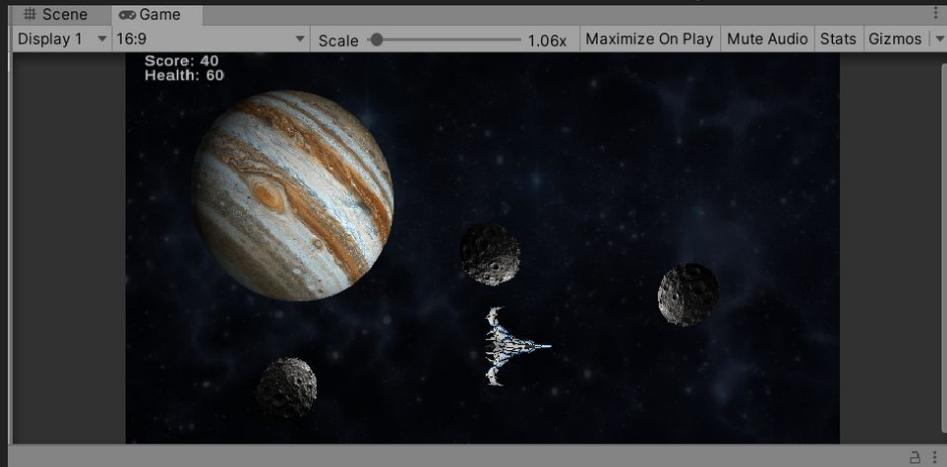
Out with the old
In with the New



This is what workin on my game looks like these days...



This is what workin on my game looks like these days...



This is what workin on my game looks like these days...

The image shows a dual-screen view of a Unity development environment. The left window displays a game menu titled "ASTEROID ANTAGONY" with "How To Play" and "START" buttons. The right window shows the C# script for "GameManager", which handles spawning asteroids and planets.

Unity Hierarchy (Left Window):

- Menu
 - Main Camera
 - HowTitle
 - Canvas
 - EventSystem
 - Music
 - Background

Script Editor (Right Window):

```
46 // Update is called once per frame
47 void Update()
48 {
49
50     healthText.text = "Health: "+health.ToString();
51     scoreText.text = "Score: " + score;
52
53
54     if (asteroidTimeTillSpawn <= 0)
55     {
56         Vector3 pos = new Vector3(xSpawn, Random.Range(ySpawnMin, ySpawnMax), 0);
57         Instantiate(asteroidPrefab, pos, Quaternion.identity);
58         asteroidTimeTillSpawn = asteroidSpawnDelay;
59     }
60     else
61     {
62         asteroidTimeTillSpawn -= Time.deltaTime;
63     }
64     ////////////////
65
66     if (planetTimeTillSpawn <= 0)
67     {
68         Vector3 pos = new Vector3(xSpawn, Random.Range(ySpawnMin, ySpawnMax), 0);
69         Instantiate(planetPrefab, pos, Quaternion.identity);
70         planetTimeTillSpawn = planetSpawnDelay;
71     }
72     else
73     {
74         planetTimeTillSpawn -= Time.deltaTime;
75     }
76     ////////////////
77
78     if (cometTimeTillSpawn <= 0)
79     {
80         Vector3 pos = new Vector3(xSpawn, Random.Range(ySpawnMin, ySpawnMax), 0);
81         Instantiate(cometPrefab, pos, transform.rotation * Quaternion.Euler(0, 0,
82         cometTimeTillSpawn = cometSpawnDelay;
```


My Goal:

My goal is to become a Lead Game Designer. I don't want to be in charge of telling people what to do without first being good at it myself. Because of this, I want to be versatile in the sense that I can help with any aspects of creating video games, whether that be, Level Design, UI/UX Design, Coding, or just being creative.

But first things first...

The thing about creating videogames is that it's not easy. People often become infatuated with the idea of being a Game Designer, and then quickly back down once they see that it's not just a walk in the park. I am currently in the stages of realizing how difficult it is to actually make a good game, but instead of choosing to give up, I am deciding to push forward and learn how to overcome any challenges that come up... especially the infamous coding part.

I want this project to serve as a testament to my dedication to Game Design.

My thoughts

I believe that I've come a long way, from knowing absolutely nothing, to now, being able to create my own video game.

Although I did indeed complete my game for the sake of this project, it's not Game Over. This is a good start, but I know I can do better. Even after this semester ends, I will continue to improve on my game and keep adding more to it and optimizing it until I can proudly publish it on a public platform to be played by the world.

Thank you Prof. Hosni

Without my technical advisor, my game wouldn't be even close to where it is today. The help and guidance is very much appreciated.

Asteroid Antagony Video Game

Giancarlo Macias

Emerging Media Game Design, NYCCT, Brooklyn New York 11201

Introduction

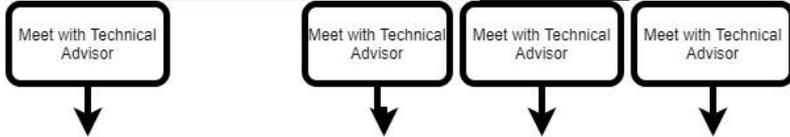
For this culmination project I will be embarking on the journey of creating my own video game from scratch via Unity. My game will be about the player controlling a spaceship trying to dodge asteroids. This game is inspired by an earlier version I made many years ago on Android App Inventor.

Process

This game will be created entirely on the Unity game engine platform. I will be using royalty free assets in my game to save time. My job will be to make the design choices I see best fit and then code them into reality.

Conclusion

My goal was to not only create my very own video game, but to also learn a lot while doing so. I definitely achieved that goal, but i'm also walking away with a future goal of continuing to improve and get better at my craft.



09/11 - 09/15	09/15 - 09/20	09/20 - 09/22	09/20 - 10/12	10/12 - 10/17	10/17 - 11/15	11/15 - Today
Figure out what exactly I want to do for my project	Solidify my project idea and get proposal approved	Submit All revised documents and get ready to actually start	Make a functional prototype which will be the skeleton for the rest of the project	Start adding some assets	Polish up the game play and make sure all desired elements are present in the game	Final preparations





"That's all Folks!"

TM & © 1993 Warner Bros. - All Rights Reserved