ECE4893A/CS4803MPG: Multicore and GPU Programming for Video Games

Lecture 10: A Walkthrough of an XNA 2D game

Prof. Hsien-Hsin Sean Lee

School of Electrical and Computer Engineering

Georgia Institute of Technology

Georgia Institute of Technology

2D Games

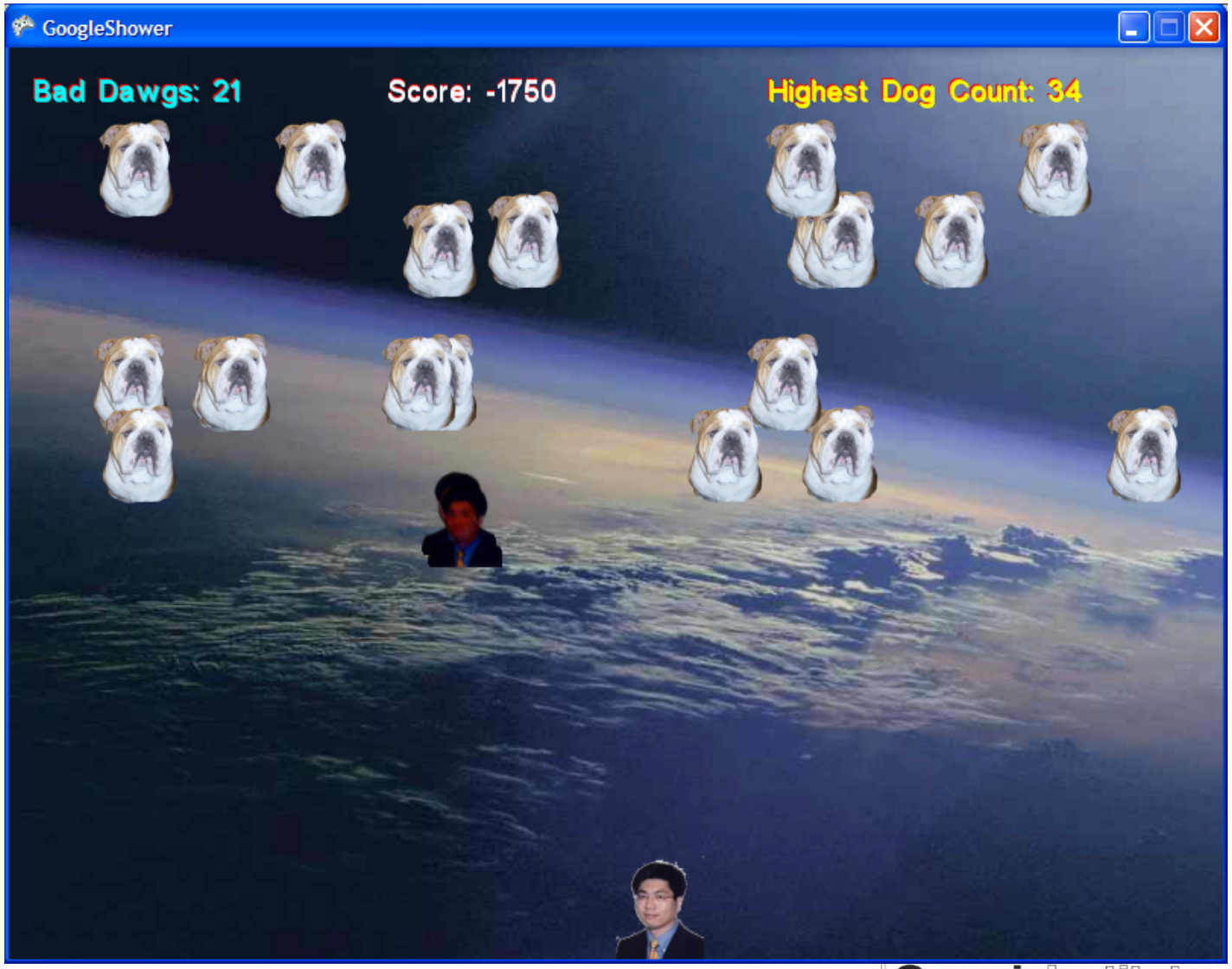
- Using “Sprites”
  - All textures
  - Simple to make or obtain
- Early games before 3D revolution
  - Space Invaders, Lode Runner, Donkey Kong, Pac-Man
- Do not require high performance accelerators
- Simple enough for your grandparents to enjoy

- Easy to do using XNA framework
- No “effect” (.fx) used

My Game: GoogleShower

- Shoot the bad dawgs!
- Consist of four main objects
  - The shooter (ship.cs)
  - The bad dawgs (meteros.cs)
  - The missile (missile.cs)
  - Music (AudioComponent.cs)
- The moving objects are all made of sprites

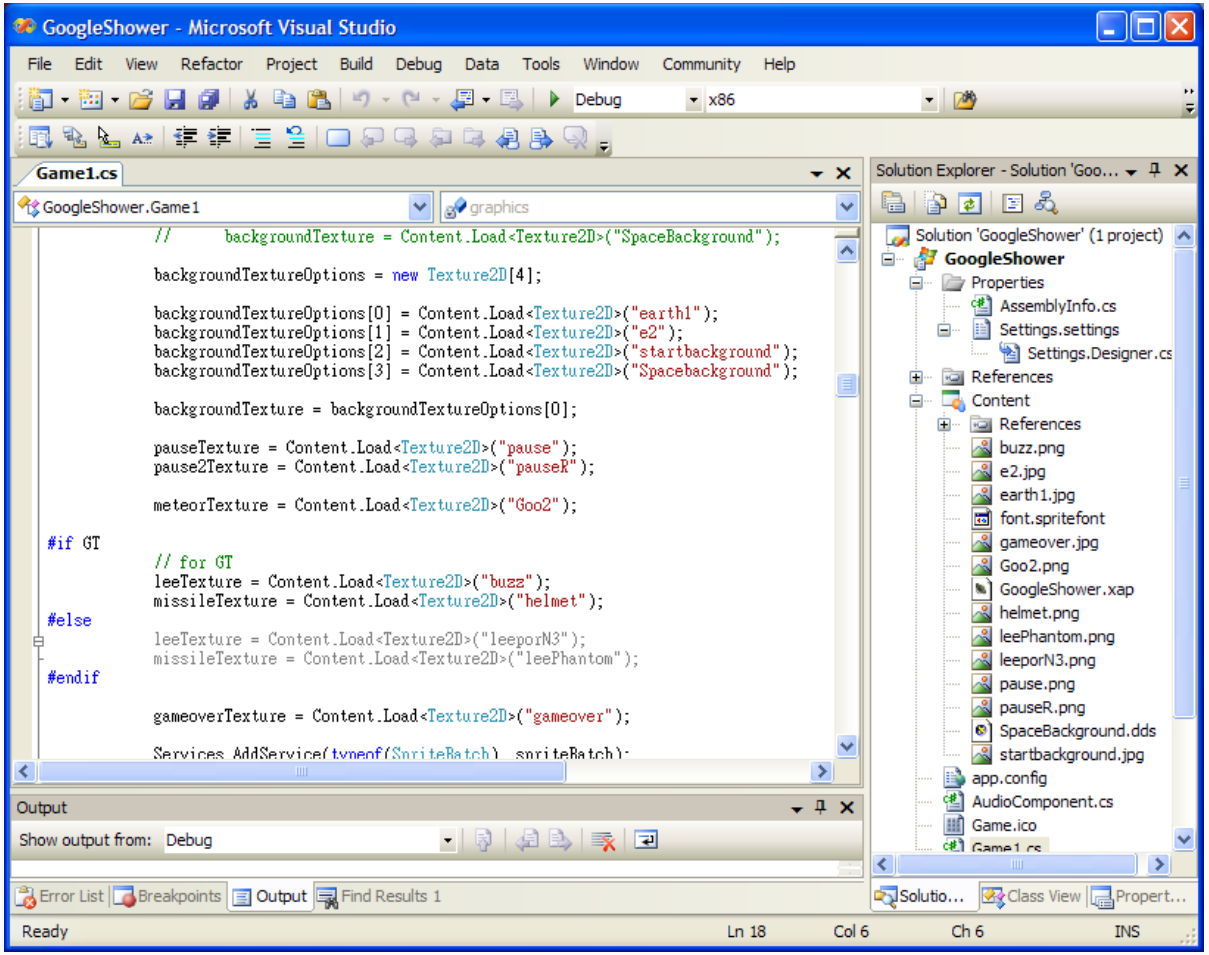
Screen Shot



Demo GoogleShower Game Example

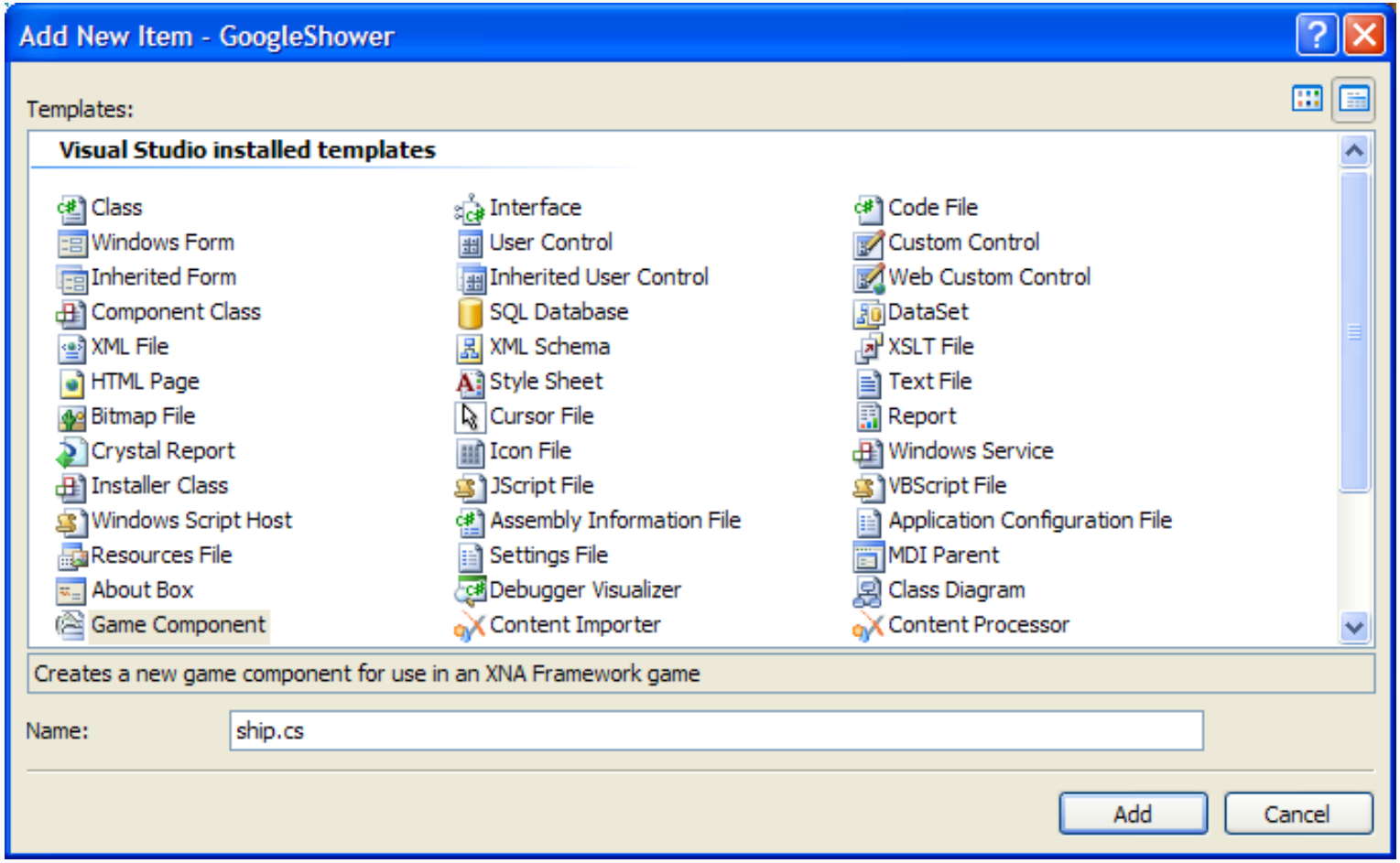
Drawing Sprite Using XNA

- Use “texture”
- Store with XNA’s `texture2D` class
- Include new texture images into the Content Pipeline
- Use “Content.Load” to associate texture variables



Georgia Institute of Technology

Adding Game Component (C# source file)



- Adding new DrawableGameComponent class of objects in the game
- Project → Add Components

Georgia Institute of Technology

Full Screen Mode

GoogleShower.Game1 | Game1()

```
        private int right_margin;
        private int score;
        private AudioComponent audioComponent;
        private SpriteFont gamefont;

        public Game1()
        {

            graphics = new GraphicsDeviceManager(this);
            Content.RootDirectory = "Content";

            // for running at Full Screen mode
            graphics.PreferredBackBufferWidth = 1024;
            graphics.PreferredBackBufferHeight = 768;
            graphics.IsFullScreen = true;
        }

        /// <summary>
        /// Allows the game to perform any initialization it needs to before starting to run.
        /// This is where it can query for any required services and load any non-graphic
        /// related content.  Calling base.Initialize will enumerate through any components
        /// and initialize them as well.
        /// </summary>
        protected override void Initialize()
        {
            // TODO: Add your initialization logic here
```

Game Services

- Game services maintain loose coupling between objects that need to interact with each other
- Register a “global” SpriteBatch for drawing all sprites
- Draw() method will look for an active SpriteBatch in GameServices
- All GameComponents will use this SpriteBatch

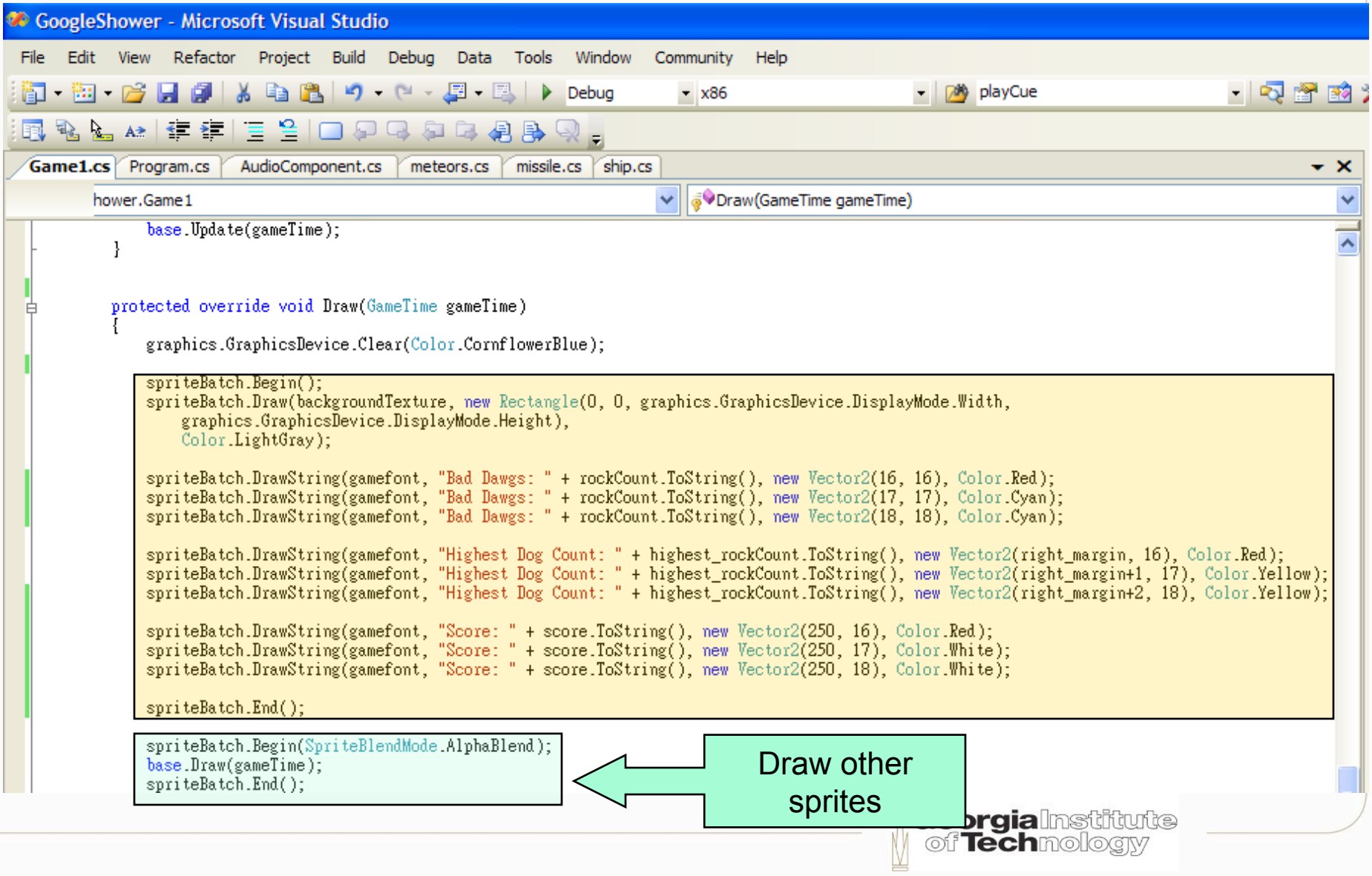
```
// Create a new SpriteBatch, which can be used to draw textures.
spriteBatch = new SpriteBatch(GraphicsDevice);

Services.AddService(typeof(SpriteBatch), spriteBatch);
```

Registering a Game Service
In LoadContent()

```
public override void Draw(GameTime gameTime)
{
    SpriteBatch sBatch =
        (SpriteBatch)Game.Services.GetService(typeof(SpriteBatch));

    sBatch.Draw(texture, position, spriteRectangle, Color.White);

    base.Draw(gameTime);
}
```

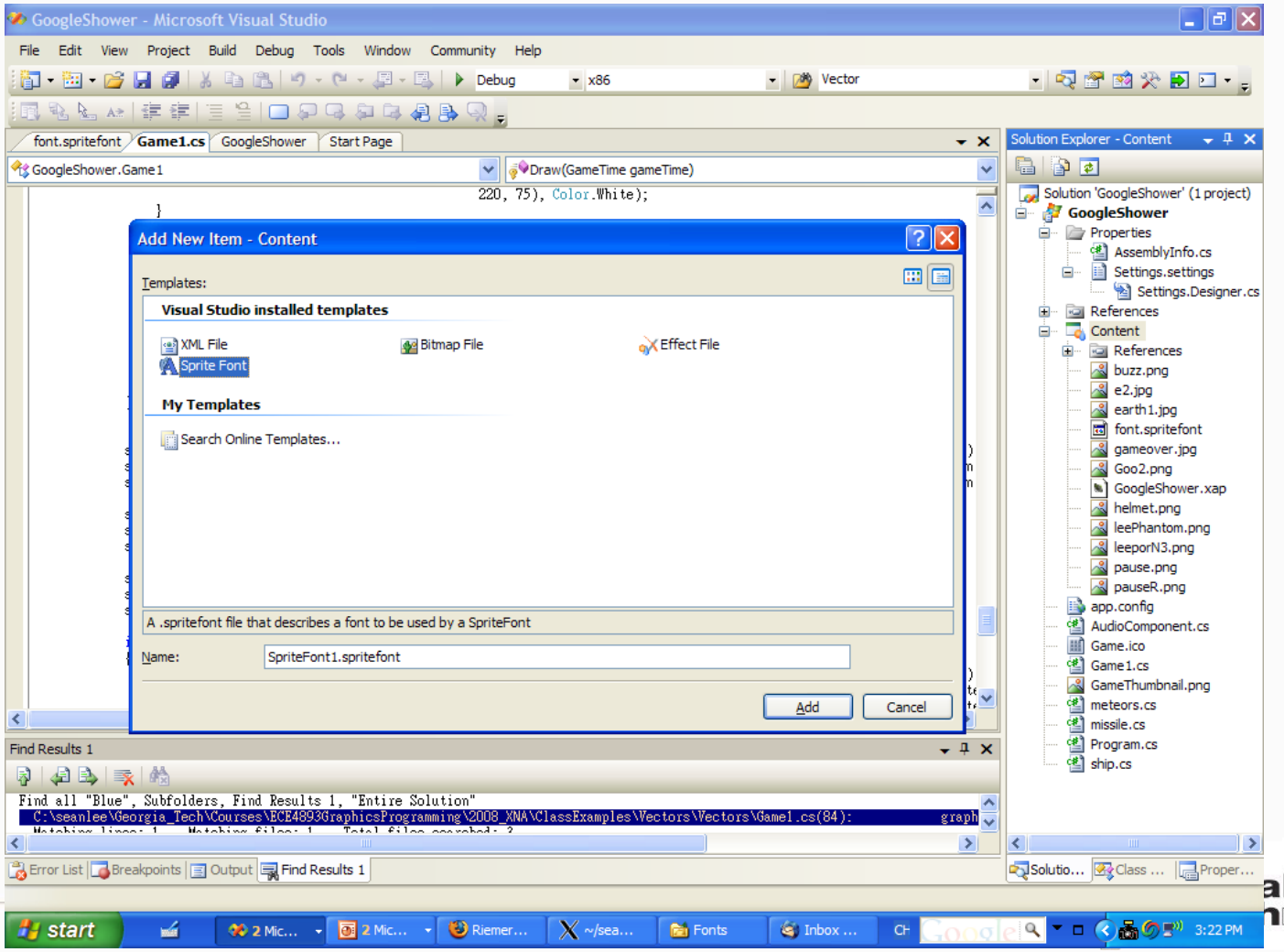
Use GetService to acquire service

Drawing Background in One Pass



DrawString (Scoreboard)

- Draw using SpriteBatch
- Create a font sprite
- Based on available Fonts in the system
- Add font in LoadContent()



Manage Components

- Components: member of GameComponentCollection (i.e., Microsoft.XNA.Framework.Game)
- Use Components.Add() to add a new component to the list

```
protected override void Initialize()
{
    audioComponent = new AudioComponent(this);
    Components.Add(audioComponent);
```

```
private void Start()
{
    if (player == null)
    {
        player = new ship(this, ref leeTexture);
        Components.Add(player);
```

```
private void CheckMissileFired()
{
    // add component is SPACE bar was hit
    if (player.shootMissile())
    {
        audioComponent.PlayCue("shoot");
        Components.Add(new missile(this, ref missileTexture, player.GetPosition()));
        player.resetShoot();
    }
```

- Components.RemoveAt(j) removes j^{th} component

```
if (hasCollision)
{
    audioComponent.PlayCue("explosion");
    // remove collided meteros (Google)
    score++;
    Components.RemoveAt(j);
```

Game Logic

```
private void DoGameLogic()
{
    bool hasCollision = false;
    Rectangle shipRectangle = player.GetBounds();
    foreach (GameComponent gc in Components)
    {
        if (gc is meteors)
        {
            hasCollision = ((meteors)gc).CheckCollision(shipRectangle);
            if (hasCollision)
            {
                audioComponent.PlayCue("missile");
                score -= PENALTY;
                dead++;
                RemoveAllMeteors();
                Start();
                break;
            }
        }
    }

    CheckforNewMeteor();
    CheckMissileFired();
    CheckMissileHit();
    AdvanceLevel();

}
```

- Embedded inside Update() function
- Logistics check

- Check collision
- Check if a missile is fired
- Check if should advance levels

Pause the Game

- Poll the keyboard state
- Simply do not perform any update inside the Update()

```
if (!pause && keyboard.IsKeyDown(Keys.Tab))
{
    pause = true;
}
else if (pause && keyboard.IsKeyDown(Keys.LeftControl))
{
    pause = false;
}

if (pause == false)
{
    DoGameLogic();
    base.Update(gameTime);
}
```

Built-in Test for Collision Detection

- Test bounding boxes of given rectangular sprites
- Return a boolean result

```
public bool CheckCollision(Rectangle rect)
{
    Rectangle spriterect = new Rectangle((int)position.X, (int)position.Y,
                                         MISSILEWIDTH, MISSILEHEIGHT);
    return spriterect.Intersects(rect);
}
```

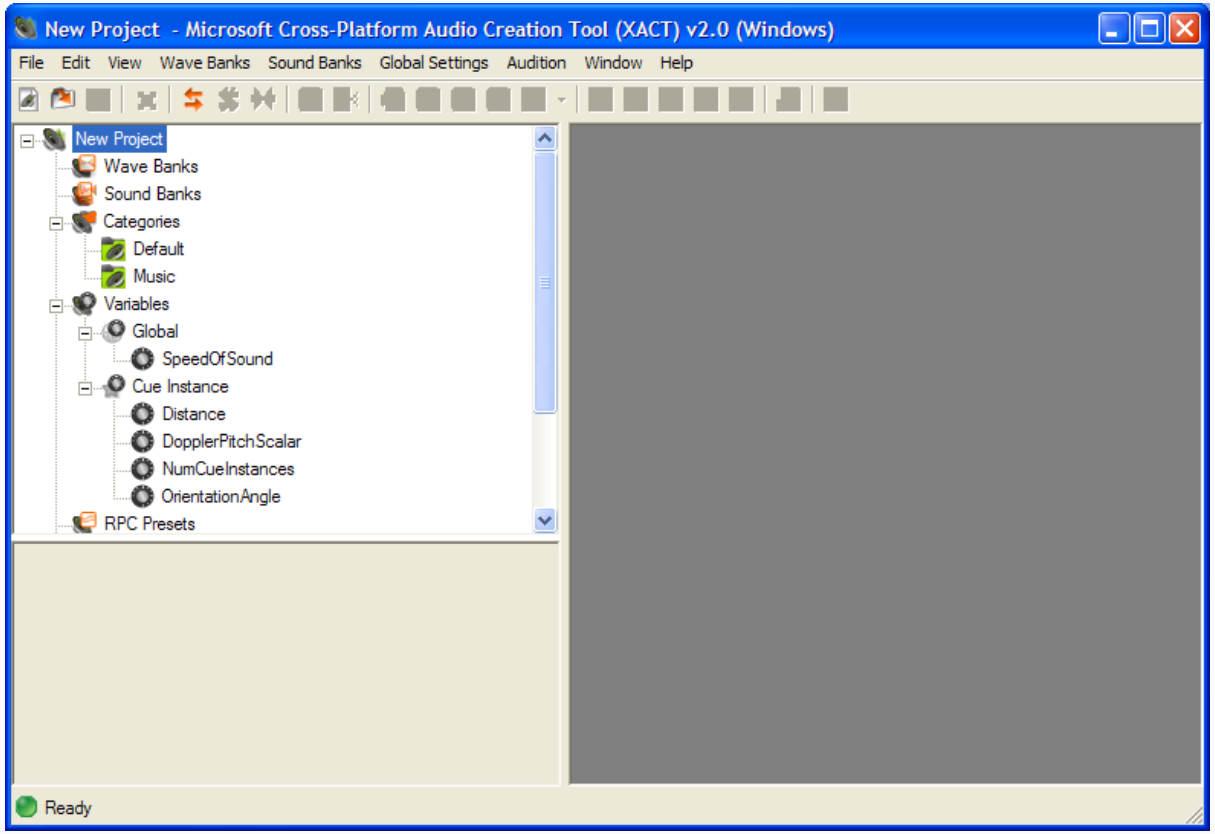
- BoundingSphere also provides similar method

```
public BoundingSphere (
        Vector3 center,
        float radius
)
```

```
public bool Intersects (
        BoundingSphere sphere
)
```

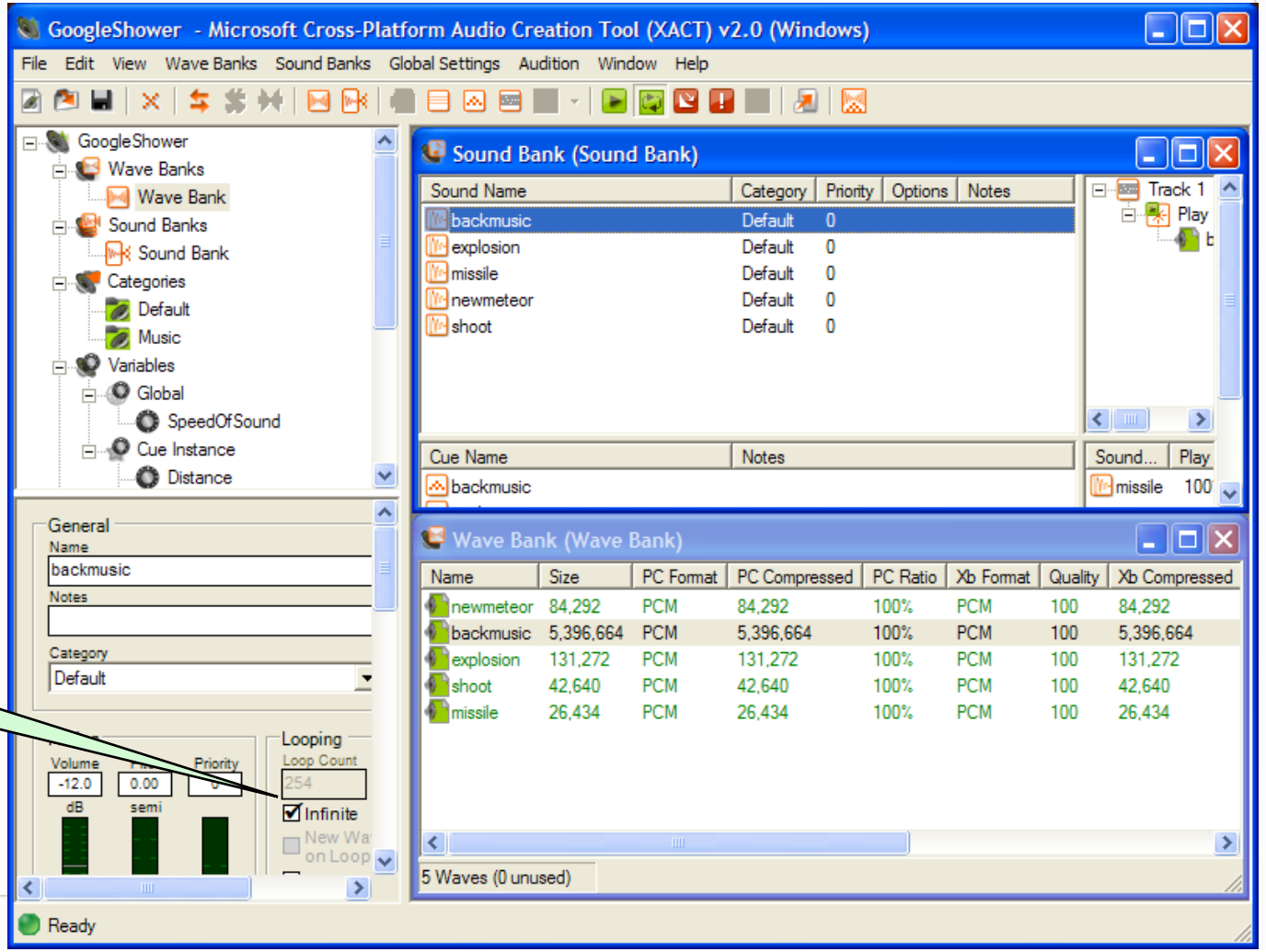
XNA Game Audio Component

- Use “Content pipeline” again
- Use Microsoft Cross-Platform Audio Creation Tool (or XACT) in XNA Game Studio 2.0



XNA Game Audio Component Cont'd

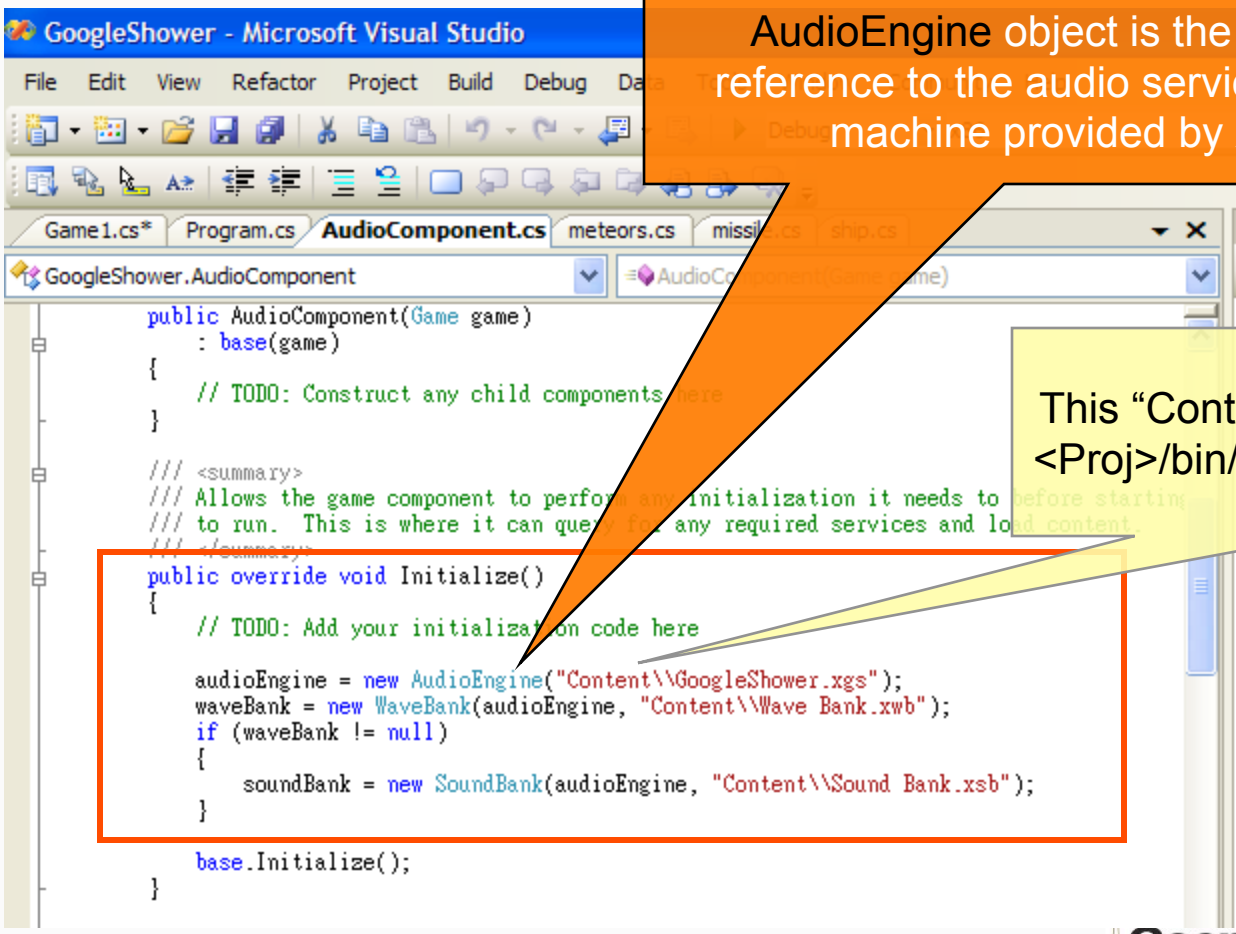
- Create wave banks and sound banks
- Compile them into XAP file used by the content manager



"checked" for looping background music

XNA Game Audio Component Cont'd

- Create a new GameComponent for audio
- Initialize WaveBank and SoundBank in the C# code

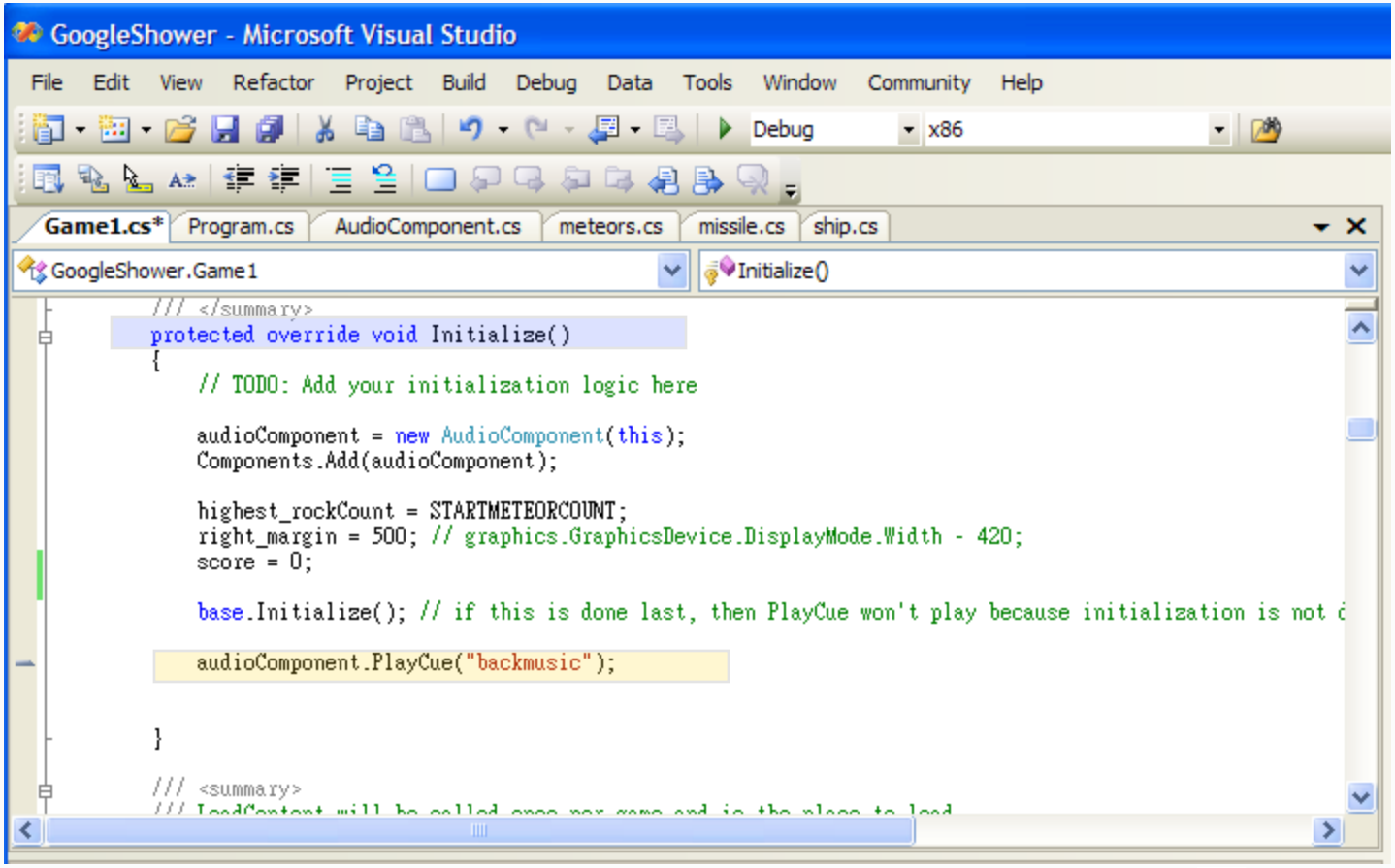


AudioEngine object is the program reference to the audio services in your machine provided by XNA

This "Content" directory is located in <Proj>/bin/x86/<Sln Config>/Content

Background Looping Music

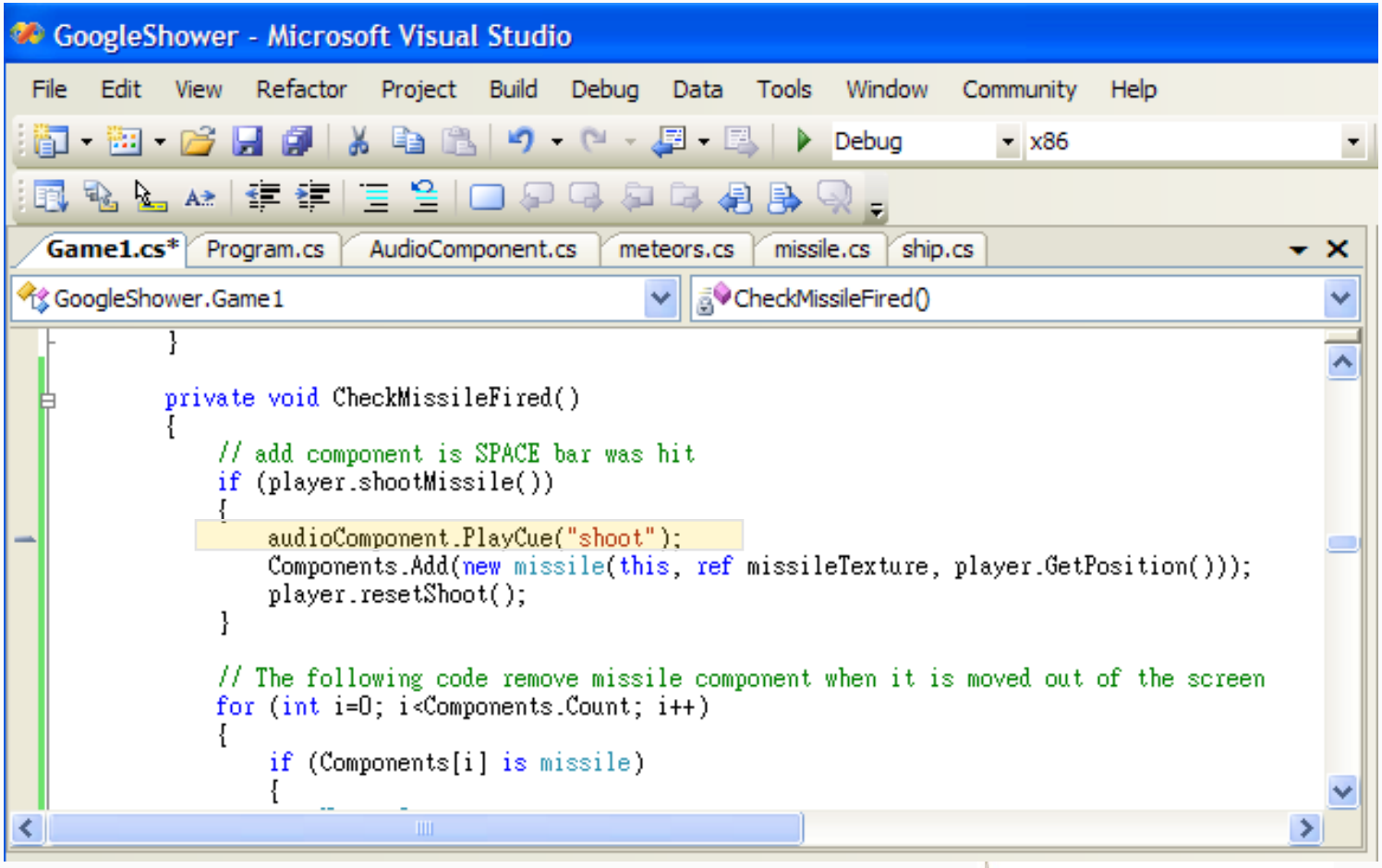
- Add in Initialize code of the Game
- PlayCue is a method of SoundBank



Georgia Institute of Technology

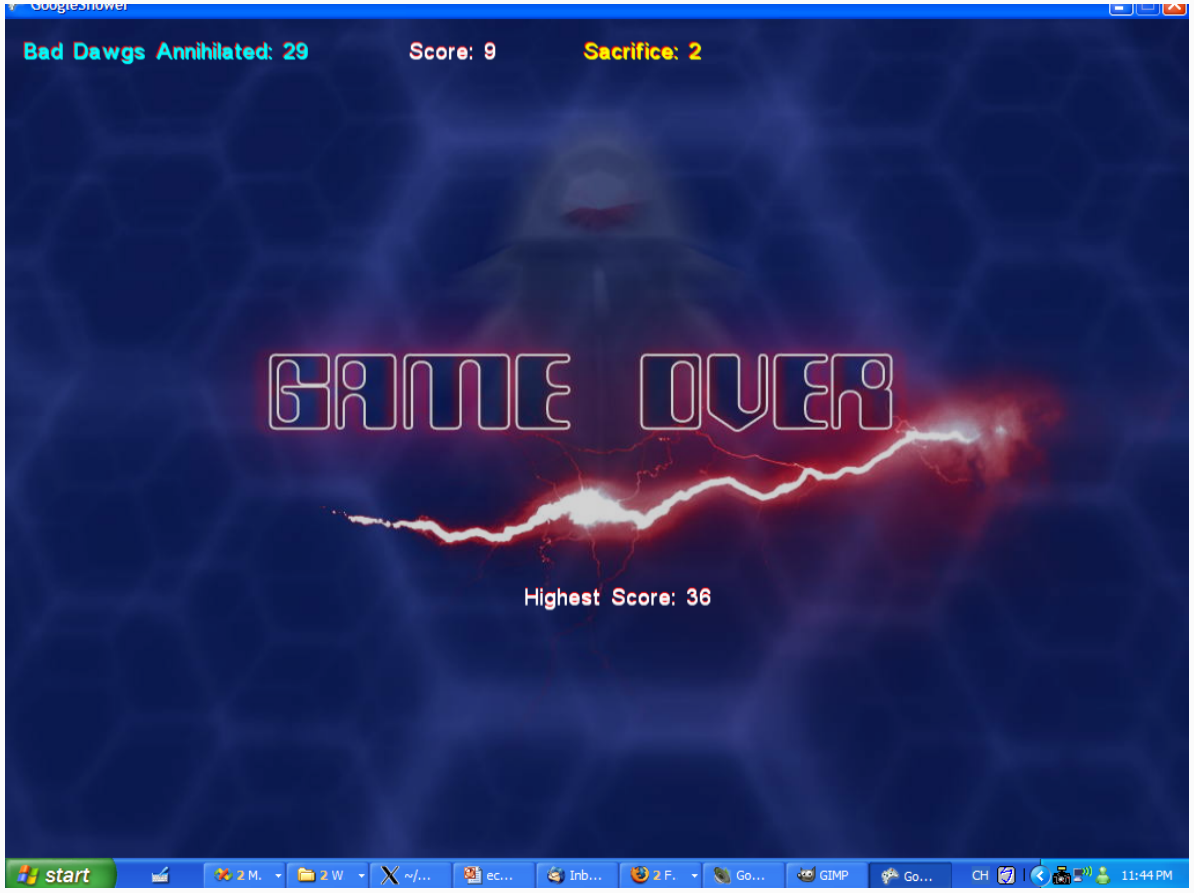
Play Sound On Event

- Shoot.wav in SoundBank was not set to “infinite”, thus will only be played once



Game Over

- Remove all components
- Replace background canvas
- Pause the music



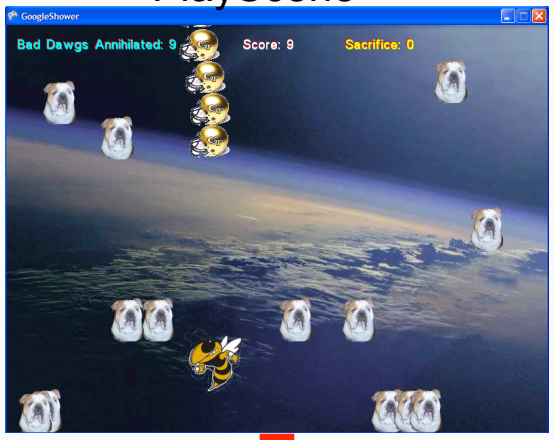

```
private void RemoveGame()
{
    for (int i = 0; i < Components.Count; i++)
    {
            Components.RemoveAt(i);
            i--;
    }
}

private void GameOver()
{
    if (dead >= DEAD_COUNT)
    {
        if (highestscore < score)
            highestscore = score;

        RemoveGame();
        backgroundTexture = gameoverTexture;
        gameover = true;
        player = null;
        backmusic.Pause();

    }
}
```

Georgia Institute of Technology

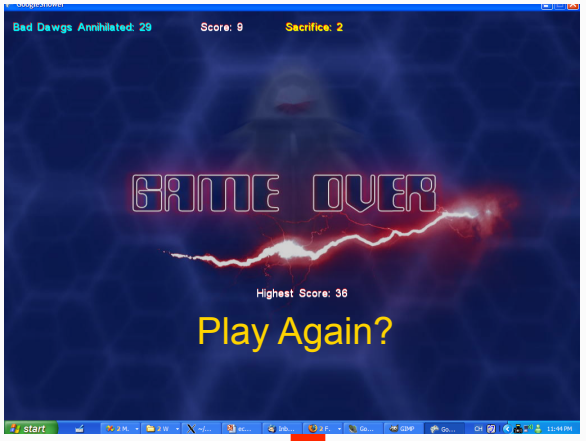
Organized Game Structure

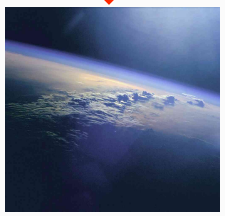
Class GameScene : DrawableGameComponent

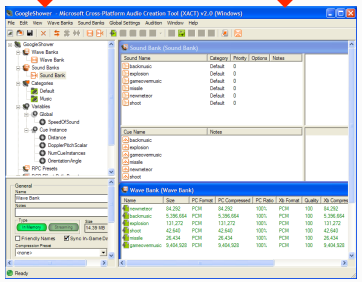
PlayScene



EndScene







SoundBank

Play Again?

Georgia Institute of Technology