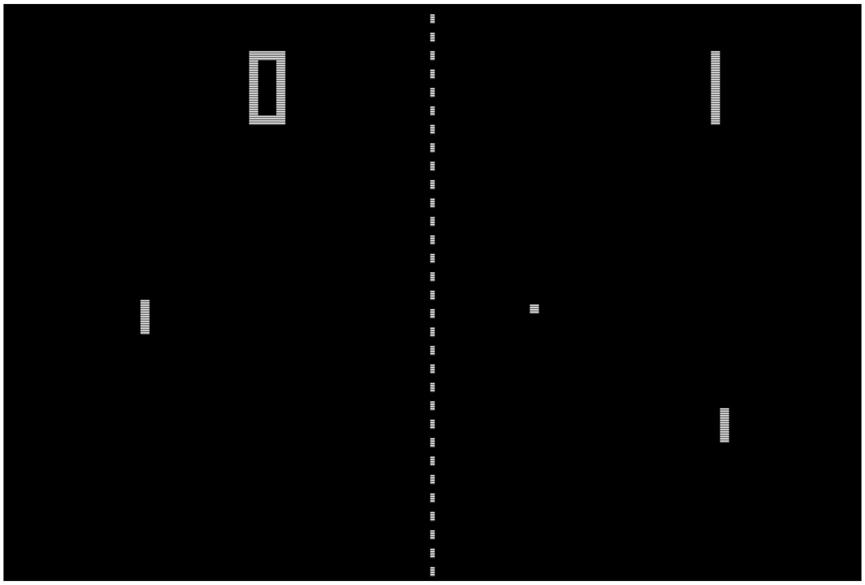
FULL SAIL UNIVERSITY

Pong Project

Weight: 10%

Overview

Due to the popularity of this game, the video game industry was born. Created by Atari, it made its first appearance in 1972 at Andy Capp's Tavern in Sunnyvale, California. A few years later in 1975, a home version of the game was released much to the delight of gamers worldwide.



Assignment

Using the Win32 API, build a Pong inspired game. This means your game should feature two objects that interact with another object. In order to get the best possible grade, try to accomplish as many goals as you can before the due date. This project is due at the end of lecture 8.

Fundamental Objectives

Multi-Threading (+/-10 Pts)

- Implement a game for Windows that utilizes multiple threads.
- The program's WinMain thread must be responsible for Windows message processing.
- The program's GameMain thread must perform game related tasks.

Display Game Components (+/-10 Pts)

- Represent each of the following game components with a graphic:
  - The Ball
  - The Left Paddle
  - The Right Paddle
- The graphic can be a filled shape, a colorful bitmap, or both.

Bitmaps: Background (+/-10 Pts)

- Display a bitmap image that will represent the background.

Simple Physics & Collisions (+/-10 Pts)

- Employ a simple physics system that enables the Ball & Paddles to move across the field of play.
- Implement simple collision detections and responses for the following interactions:
  - Ball to Wall – The Ball must be programmed to move the opposite direction.
  - Paddle to Wall – The Paddle must be stopped from going any further.
  - Ball to Paddle – The Ball must be programmed to move the opposite direction.
- At no point should the Ball or Paddles be allowed to leave the field of play.
- It may be necessary to reposition the Ball or Paddles if they get stuck.

Scoreboard (+/-10 Pts)

- Tally up and render the scores for the Left and Right Paddles.
- It is up to the game designer to determine what constitutes a scoring event and how points will be award or deducted.

Graphical Objectives

Color Selection: Game Components (+2 Pts)

- Allow the user to change the color for the following game components:
  - The Ball
  - The Left Paddle
  - The Right Paddle
  - The Scoreboard
  - The Background

Color Selection: Random (+2 Pts)

- Allow the user to set all the colors used in the game to random values via a menu item.

Bitmaps: Left Paddle (+3 Pts)

- Display a bitmap image that will represent the Left Paddle.

Bitmaps: Right Paddle (+3 Pts)

- Display a bitmap image that will represent the Right Paddle.

Bitmaps: Ball (+3 Pts)

- Display a bitmap image that will represent the Ball.

Bitmaps: Scoreboard Tally (+3 Pts)

- For each point a Paddle has scored, display a bitmap image.
- The bitmap image can also represent lives/health which will be reduced when a Paddle is scored upon.

Bitmap Fonts (+3 Pts)

- Display 3 strings and/or numeric values using a Bitmap Font.
- A Bitmap Font is an image file that contains letters and/or numbers.
- The programmer must make a series of calls to DrawBitmap, each time rendering 1 of the characters in a string or 1 of the digits in a numeric value.

Percentage Bar (+3 Pts)

- Display a filled rectangle that will represent a percentage value used by the game.
- Modify the length of the filled rectangle based upon the percentage value.
- This percentage value can represent anything, some examples being: the Paddle’s health, the amount of energy needed to enable a special ability, or the amount of time until the next boss battle begins.

Gradient Brush (+3 Pts)

- Utilize a Direct2D linear or radial gradient brush to display a game component or user interface element.
- Look up "How to Create a Linear Gradient Brush (Windows)" and/or "How to Create a Radial Gradient Brush (Windows)" in the MSDN for more details.

Sprite Animation (+3 Pts)

- Render a game component using a sprite animation.

Multiple Sprite Animations (+2 Pts)

- Render a second game component using a different sprite animation.

In-Client Title Screen (+3 Pts)

- Begin the game by displaying the title of the game within the window's client area
- Clicking on the screen or pressing a key should start the game.
- A MessageBox Title Screen will award partial credit.

In-Game Instructions (+2 Pts)

- Display the instructions for the game.
- This screen should inform the user what are the controls for the game and the objective of the game.

Custom Icon & Cursor (+2 Pts)

- Create a custom icon and load it up with the LoadIcon function.
- Create a custom cursor and load it up with the LoadCursor function.

Particle Effects (+4 Pts)

- Implement a particle emitter that emits at least 5 particles.
- This effect can be attached to the Ball or the Paddles, be displayed as a collision response, or represent a weather effect in the background.
- The particles can be rendered with bitmaps, rectangles, ellipses, lines, or characters.

Multiple Particle Effects (+2 Pts)

- Implement a second particle emitter that emits different particles.

Shadows (+3 Pts)

- Render a shadow effect for a game component.

3D Graphics (+4 Pts)

- Render a game component using a 3D rendering algorithm or render the field of play so that the Ball or Paddles can move along the Z axis.

Parallax Background **(+5 Pts)**

- Implement a background that consists of 3 or more layers.
- The layers closest to the screen must scroll at a fast pace.
- The layers in the middle must scroll at a medium pace.
- The layers farthest from the screen must scroll at a slow pace.

Input Objectives

Keyboard Controlled Paddles **(+2 Pts)**

- Allow the user to control the Left and Right Paddles via keyboard input.

Mouse Controlled Paddle **(+3 Pts)**

- Allow the user to control the Left or Right Paddles via mouse input.

Joystick Controlled Paddle **(+4 Pts)**

- Allow the user to control the Left or Right Paddles via joystick input.

Multi Mouse/Joystick Controlled Paddles **(+5 Pts)**

- Allow the user to control both the Left and Right Paddles with two mice or two joysticks.
- Using Raw Input it is possible to decipher between touchpad input and external mouse input.

File Menu Objectives

File New **(+2 Pts)**

- Allow the user to restart the game.

Static Saving & Loading **(+2 Pts)**

- Allow the user to save a Pong game to a file defined by the programmer (ex. Game.dat)
- Allow the user to load a Pong game from a file defined by the programmer.
- Please save and load the scores and the game mode (one player or two players).
- File IO can be accomplished by using the fstream classes.

Dynamic Saving & Loading **(+4 Pts)**

- Allow the user to save a Pong game to a file selected by the user.
- Allow the user to load a Pong game from a file selected by the user.
- See Common Item Dialog (Windows) in the MSDN for further details.
- Please save and load the scores and the game mode (one player or two players).
- File IO can be accomplished by using the fstream classes.

Last Chance Saving **(+3 Pts)**

- Allow the user one last chance to save any time the game is about to be cleared out.
- A message box should ask the user, "Would you like to save?" and provide the options: Yes, No, Cancel.
- The following commands clear out the game: File New, Open, Exit, and the Close Button.
- Do not perform a last chance save during the Save or Save As commands.

Title-bar Management **(+2 Pts)**

- Display the Program’s name along with the file name in the title-bar.
- If the game has never been saved or opened, display “Untitled” in place of the file name.
- To get credit for this goal, saving and loading must be functional.
- Use the SetWindowText function to change the title-bar string after window creation.

Sound Objectives

Game Sound Effects **(+2 Pts)**

- Play a sound effect when a game event occurs.
- A game event can be the Ball colliding with a wall, the Paddle scoring a point, or when a winner is declared.
- A sound effect can be played with the Beep function, the MessageBeep function, or the PlaySound function.

Multiple Sound Effects **(+2 Pts)**

- Play a second game sound effect using a different sound function or sound file.
- Playing background music will award credit for this goal.

Mute/Unmute **(+3 Pts)**

- Provide a menu option and keystroke that will mute and un-mute the game sound effects.

Volume **(+3 Pts)**

- Provide a menu option and keystroke that will increase the volume of the game sound effects.
- Provide a menu option and keystroke that will decrease the volume of the game sound effects.

In-Client Volume Control **(+4 Pts)**

- Display within the client area of the game window a control that will allow the user to adjust the volume.
- The control can have 2 buttons, 1 for increasing the volume and 1 for decreasing the volume.
- Or the control can have a slider box that will allow the user to pick a volume percentage.

Music Playlist **(+3 Pts)**

- Create and utilize a playlist file that will determine the order of background music played throughout the game.

Sound API **(+4 Pts)**

- Utilize a Sound API to play sound effects.
- The following are Sound APIs that will give you credit for this goal:
  FMod, DirectX (XAudio2 or DirectSound), OpenAL, SDL, Wwise

Game Play Objectives

Winner Check **(+2 Pts)**

- Once a certain scoring condition is reached, declare a winner.

Multiple Balls **(+3 Pts)**

- Implement two or more game components that have the behavior of a Ball.

Multiple Paddles **(+4 Pts)**

- Implement three or more game components that have the behavior of a Paddle.
- To be considered a Paddle, the game component must be controllable via input and the game component must accumulate a score.

Power-Ups **(+3 Pts)**

- Implement a game component that either helps or hurts the Paddles.

Obstacles **(+3 Pts)**

- Implement a game component that will affect the Ball's velocity or the Paddle's velocity.

Special Ability **(+3 Pts)**

- Implement a special ability that can be performed by a Paddle.
- Provide a key that will invoke the Special Ability.
- Please document the Special Ability within the instructions (if implemented).

Bullet **(+3 Pts)**

- Implement a game component that a Paddle can spawn.
- A Bullet can behave like a projectile or like a mine.

Multiple Bullets **(+2 Pts)**

- Allow the Paddle to spawn 2 or more Bullets at the same time.

Bullet Upgrades **(+3 Pts)**

- Implement a more powerful version of the Bullet.
- Being more powerful means it can move faster, have a bigger collision box, split into multiple Bullets, or do greater damage.

Taunting **(+2 Pts)**

- Implement a taunting action that can be performed by a Paddle.
- While taunting, input and collision for the Paddle must be disabled.

Jumping **(+3 Pts)**

- Implement a jumping action that can be performed by a Paddle.
- Upon reaching the apex of the jump, the Paddle must be pulled back to level ground.

Platforms **(+4 Pts)**

- Implement a platform that the Paddle can jump onto or fall off of.

Tracking Artificial Intelligence **(+3 Pts)**

- Allow the user to enable an AI player that will track the Ball.
- If the Ball is moving up, move the Paddle up and if the Ball is moving down, move the Paddle down.

Efficient Artificial Intelligence **(+4 Pts)**

- Allow the user to enable an AI player that will calculate where the Ball is headed.
- Once calculated, move the Paddle to that exact location.
- While this AI mode will be impossible to defeat, it is a great programming exercise.

Boss Artificial Intelligence **(+4 Pts)**

- Allow the user to enable an AI player that will use a special ability.

Design Objectives

In-Game Contact Info **(+2 Pts)**

- Allow the user to view your name, email address, or Url so that he or she can compliment you, criticize you, or offer you employment.

Technical Objectives

String Table Utilization **(+2 Pts)**

- Put all strings used throughout the game within the string table.
- Minimum 3 Strings.

Full-screen Toggle **(+4 Pts)**

- Allow the user to toggle between Full-screen mode and Windowed mode.
- In Full-screen mode, the title-bar and menu must be removed and the client must cover the entire screen.
- In Windowed mode, the title-bar and menu must be attached and the client will appear within a window frame.
- Do not allow the Ball or Paddles to leave the field of play as a result of a full-screen toggle.

Achievements **(+4 Pts)**

- Implement 3 or more achievements that the user can accomplish while playing the game.
- The following is a list of recommended achievements:
    - Defender – Hit the ball 5 times in a row.
    - Victory – Win the game.
    - On Fire – Win 3 games in a row.
    - Domination – Win by shutout (opponent does not score)
    - Like a Boss – Win a game against a Boss Mode AI opponent.
- Display which achievements have been accomplished by either checking menu items or within an in-game screen.

Demo Mode **(+3 Pts)**

- Implement a mode where all Paddles are controlled by AI.

Pause Mode **(+3 Pts)**

- Implement a mode that disables Paddle input and prevents simulation.

Cheat Code (Key Combination) **(+2 Pts)**

- Implement a key press combination that invokes a cheat code.

Cheat Code (Press & Hold Key) **(+3 Pts)**

- Implement a cheat code that is enabled by pressing and holding a key for 3 or more seconds.

Cheat Code (Sequential Keys) **(+3 Pts)**

- Implement a cheat code that is enabled by pressing a sequence of 3 or more keys.
- The sequence must be inputted in the correct order for the cheat code to activate.

Warnings **(+2 Pts)**

- Successfully compile your game with the "Warning Level" compiler option set to "Level4 (/W4)" and the "Treat Warnings As Errors" compiler option set to "Yes (/WX)".
- To set these options, go to:
  Project Properties → C/C++ → General.

Advanced Physics **(+4 Pts)**

- Implement a physics system that factors in additional attributes like acceleration, gravity, mass, or torque.

Screen Shot **(+4 Pts)**

- Allow the user to press a key that will store the current image on the screen as an image file.

Command Line Parameters **(+4 Pts)**

- Allow the user to specify command line parameters.
- To set the command line parameters while developing, go to:
  Project Properties → Debugging → Command Arguments.
- The command line parameters can control features like AI, full-screen mode, screen resolution, or starting scores.

In-Client Start Menu **(+4 Pts)**

- Upon startup of your project, render a menu within the game client that has 3 or more options.
- The following is a list of recommended menu options:
  - New Game – Starts a new game.
  - Load Game – Loads a previously saved game.
  - AI: On/Off – Toggles the AI opponent on or off.
  - Instructions – Displays the game's instructions.
  - Credits – Displays the game's credits.
  - Exit – Quits the game.
- Allow menu options to be selected via mouse or keyboard input.

In-Client Character Selection **(+4 Pts)**

- Upon startup of your Pong project, provide an in-game character screen that allows the user to select the graphic that will represent their Paddle throughout the game.

In-Client Talent Tree **(+5 Pts)**

- Allow the user to pick and choose special abilities that can be used throughout the game.

Instant Replay **(+5 Pts)**

- Allow the user to watch an instant replay of their point.

In-Client Winner Prompt **(+5 Pts)**

- Rather than displaying the winner message with the MessageBox function, display the winner message within the window's client area.
- The prompt must include 2 options that can be selected, one option that restarts the game and another that ends the game.
- Selecting an option can be implemented with mouse and/or keyboard input.

Grading Criteria

- The 5 fundamental goals are worth 10 points each if implemented or -10 points each if not implemented.
- Complete secondary goals to accumulate more points.
- The maximum project grade is 100.
- Every point earned over 100 will be converted into Course Director Points.
- The student with the most Course Director Points at the end of the course will be awarded the Course Director Award.

Level of Difficulty	Point Value
Fundamental	+10 if implemented; -10 if not implemented
Easy	+1
Medium	+2
Hard	+3
Epic	+4
Legendary	+6

Submission

Pong is due at the end of lecture 8. To submit, perform the following:

- Fill out the excel rubric by entering your name and checking the goals you have attempted. Copy this file into the directory containing your Visual Studio project file.

- Find your Pong project directory and delete any Debug, Release, and Ipch folders. Also delete any *.ncb, *.sdf, and *.aps files.

- Zip the folder(s) containing your project files and excel rubric.

- Rename the zip file to: LastName**.**FirstName**.**Pong**.**zip
  (Substitute your last and first names in place of LastName & FirstName.)

- Copy the zip file into the following network directory: \\studentvfiler\WGP\WGP ####\Turn-In (Where #### equals the year and month).

- If you are unable to connect to the wireless network, copy the zip file onto a pen drive and have a teacher copy it into the Turn-In Folder.

- Once submitted, verify with a teacher that project has been successfully turned-in.