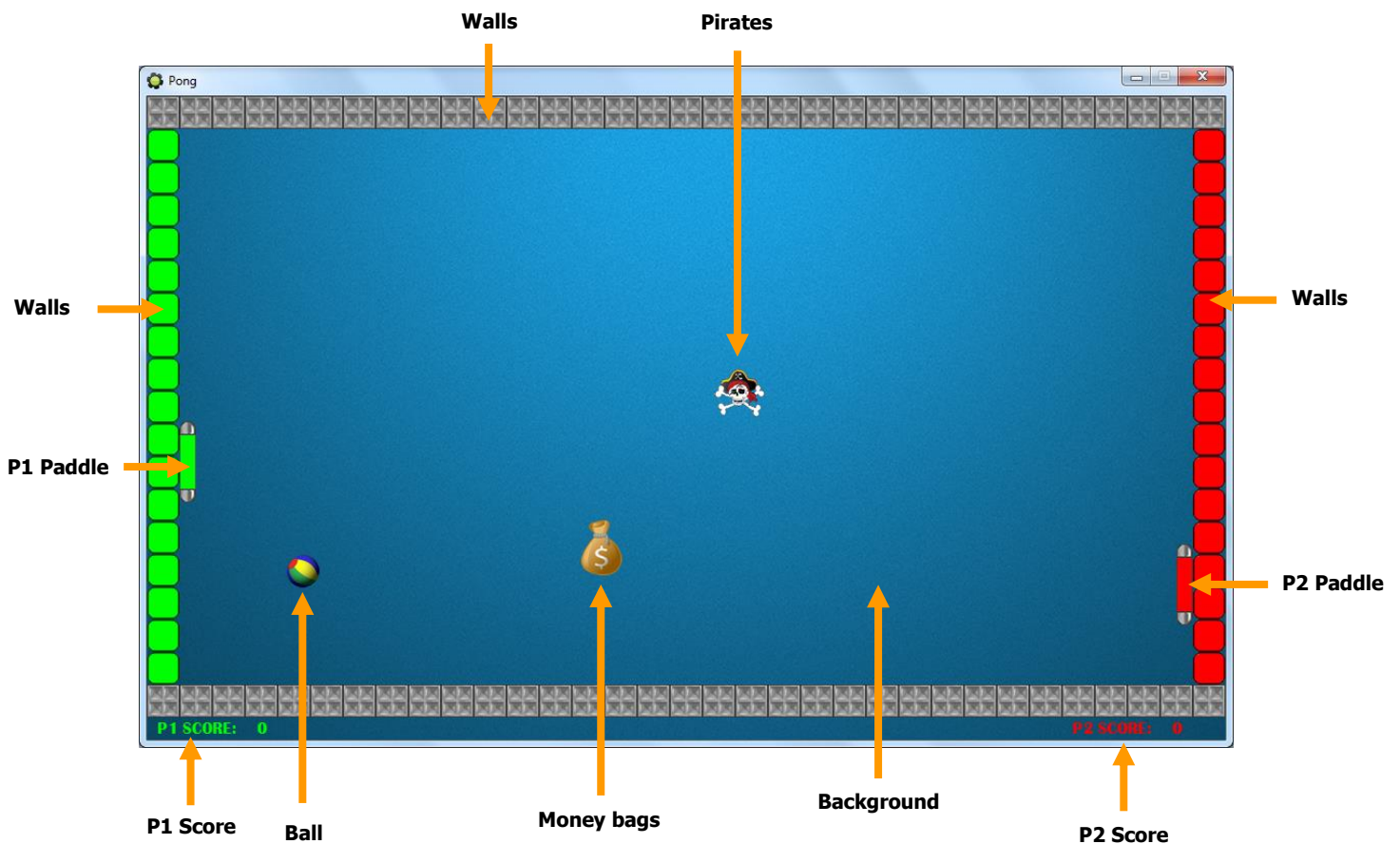


INTRODUCTION TO GEMEMAKER: SUPER PONG

For your first programming assignment, you will be creating a variant of the classic game Pong. In this version of Pong, the objective is to acquire more points than your opponent by shooting the money bags, avoiding the pirates, and ensuring the bouncing ball does not make contact with your home base.

Your user interface should look something like this:



START OF THE GAME

When the game begins, each player's score should be set to 0. You will need to use two different types of scores for the two players, since using the same scoring system for more than one player requires the use of variables (which we will be learning in the next unit). So use the **Score** function to keep track of Player 1's score and use the **Lives** function to keep track of Player 2's score. Be sure to set them both to 0.

Once the game begins, the beach ball must start moving diagonally and must bounce off the walls. Feel free to set the speed to whatever you want the speed to be.

PLAYER CONTROLS

Each player will control their paddles using keys on the keyboard. For example, you may want Player 1 to use keys on the left side of the keyboard and Player 2 to use keys on the right side of the keyboard. I don't really care which keys you decide to use.

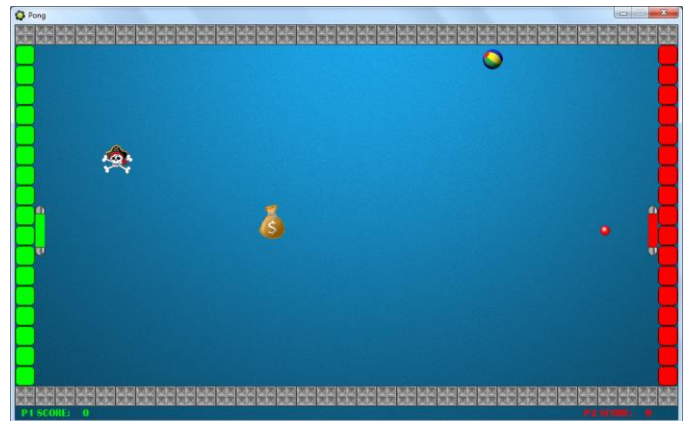
The objective for each player is to avoid ensure the beach ball does not come into contact with the walls behind them. Each time the ball hits a wall behind them, the player loses **200 points**.

MONEY BAGS

While each player makes sure the ball does not come into contact with the walls behind them, the player can gain points by shooting bullets at the money bags that randomly appear and disappear. Each money bag the user hits earns them **50 points**. Again, you decide what key the players will need to press to fire bullets. Player 1 should be firing green bullets and Player 2 should be firing red bullets.



Player 1 fires green bullets



Player 2 fires red bullets

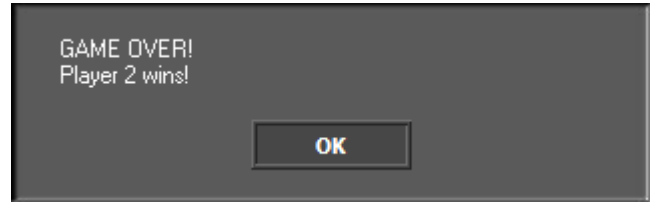
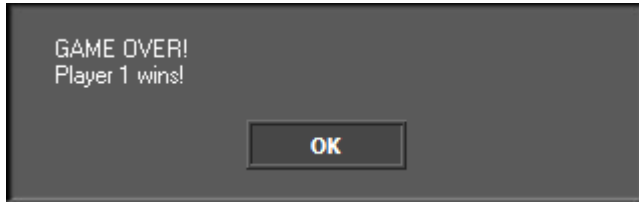
PIRATES

Players have to be careful not to hit the pirates. If a pirate is hit with a bullet, the player loses **100** points. Like the money bags, pirates should randomly appear and disappear on the screen.

ENDING THE GAME

The game ends when either Player 1 or Player 2 has less than 0 points. The player with less than 0 points loses the game and the other player wins the game.

When the game ends, a message should be outputted to the user indicating who wins:



You can create a message by clicking on the **Display Message** icon which you can find in the main2 tab.

SOUND EFFECTS AND BACKGROUND MUSIC

Be sure to include sound effects for the following actions:

- When the ball bounces off the walls
- When a player hits a pirate
- When a player hits a money bag
- When the ball bounces off either of the player's walls and loses the game

You will also need to include background music that plays while the game is running.

Here is a list of websites where you can find free sound files for your game:

<http://www.flashkit.com/soundfx/>
<http://www.grsites.com/archive/sounds/>
<http://www.sfxsource.com/free-sound-effects.cfm/>
<http://www.findsounds.com/types.html>
<http://www.freesound.org/>

GAME DESIGN OUTLINE

In addition to creating the game, you will also need to come up with an outline of your game. Your outline should include a brief description of the game that walks us through the main points and objective of the game.

For your game design outline, you will need to use the following Game Design template:

Game Title

SUMMARY

Provide a brief description of the game, including the objective of the game, the gameplay (i.e. how the game works), characters in the game, the scene(s) in which the action takes place, etc.)

GAME OBJECTS

Describe the objects that you will be using, what these objects will look like, and the purpose these objects will serve in the game.

SOUNDS

Describe the sounds (and sound effects) you will be using and the purpose the sounds will serve in the game.

CONTROLS

Explain what the user will be using to control the objects in the game (e.g. left mouse button, a key on the keyboard, etc.).

GAME FLOW

Describe how the game works, from beginning to end. In other words, describe what happens when the game begins, what the user's options are, what the other objects in the game do, what happens when the user performs certain actions, etc.

LEVELS

If there is more than one level, describe each level. Even if there is only one level, describe how the difficulty of the game increases as the game progresses.

Be sure to save your project, as well as your Game Design outline, in a folder called **Super Pong** in your COMPLETED ASSIGNMENTS folder.

INTRODUCTION TO GEMMAKER: SUPER PONG

CATEGORY	CRITERIA	< LEVEL 1 0 – 49%	LEVEL 1 50 – 59%	LEVEL 2 60 – 69%	LEVEL 3 70 – 79%	LEVEL 4 80 – 100%	MARK
Knowledge and Understanding	Demonstrates an understanding of how to create a game using some of GameMaker’s basic tools and techniques	<ul style="list-style-type: none"> Demonstrates no understanding of how to create a game using GameMaker’s basic tools and techniques <p style="text-align: center;">0-4.9</p>	<ul style="list-style-type: none"> Demonstrates limited understanding of how to create a game using GameMaker’s basic tools and techniques <p style="text-align: center;">5.0-5.9</p>	<ul style="list-style-type: none"> Demonstrates some understanding of how to create a game using GameMaker’s basic tools and techniques <p style="text-align: center;">6.0-6.9</p>	<ul style="list-style-type: none"> Demonstrates considerable understanding of how to create a game using GameMaker’s basic tools and techniques <p style="text-align: center;">7.0-7.9</p>	<ul style="list-style-type: none"> Demonstrates thorough understanding of how to create a game using GameMaker’s basic tools and techniques <p style="text-align: center;">8.0-10</p>	/10
Thinking	<p>The game meets all the outlined requirements</p> <p>Tests and debugs the game to ensure the program meets all the requirements</p>	<ul style="list-style-type: none"> The game meets none of the outlined requirements Tests and debugs the program with no success <p style="text-align: center;">0-4.9</p>	<ul style="list-style-type: none"> The game meets a limited number of the outlined requirements Tests and debugs the program with limited success <p style="text-align: center;">5.0-5.9</p>	<ul style="list-style-type: none"> The game meets some of the outlined requirements Tests and debugs the program with some success <p style="text-align: center;">6.0-6.9</p>	<ul style="list-style-type: none"> The game meets most of the outlined requirements Tests and debugs the program with considerable success <p style="text-align: center;">7.0-7.9</p>	<ul style="list-style-type: none"> The game meets all of the outlined requirements Tests and debugs the program with great success <p style="text-align: center;">8.0-10</p>	/10
Communication	Provides a game design outline that clearly describes the key features and objective of the game	<ul style="list-style-type: none"> Communicates the key features and objectives of the game with no success <p style="text-align: center;">0-4.9</p>	<ul style="list-style-type: none"> Communicates the key features and objectives of the game with limited success <p style="text-align: center;">5.0-5.9</p>	<ul style="list-style-type: none"> Communicates the key features and objectives of the game with some success <p style="text-align: center;">6.0-6.9</p>	<ul style="list-style-type: none"> Communicates the key features and objectives of the game with considerable success <p style="text-align: center;">7.0-7.9</p>	<ul style="list-style-type: none"> Communicates the key features and objectives of the game with great success <p style="text-align: center;">8.0-10</p>	/10
Application	Effectively applies programming knowledge and skills of GameMaker to create a game	<ul style="list-style-type: none"> Applies programming knowledge and skills of GameMaker with no success <p style="text-align: center;">0-4.9</p>	<ul style="list-style-type: none"> Applies programming knowledge and skills of GameMaker with limited success <p style="text-align: center;">5.0-5.9</p>	<ul style="list-style-type: none"> Applies programming knowledge and skills of GameMaker with some success <p style="text-align: center;">6.0-6.9</p>	<ul style="list-style-type: none"> Applies programming knowledge and skills of GameMaker with considerable success <p style="text-align: center;">7.0-7.9</p>	<ul style="list-style-type: none"> Applies programming knowledge and skills of GameMaker with great success <p style="text-align: center;">8.0-10</p>	/10

CURRICULUM EXPECTATIONS THAT ARE COVERED IN THIS ASSIGNMENT:

- B1. Describe fundamental programming concepts and constructs.
- B2. Plan and write simple programs using fundamental programming concepts.
- B3. Apply basic code maintenance techniques when writing programs.