

CISC 3667: Game Design and Development

In-Class Activity 8: Unity Lab 2 - Player Interaction, Collision, and Game State

Assigned: Thursday, October 29

Submission: Link to game submitted to class [Game Jam](#) via Brightspace on Thursday, October 29, 2026, at 11:59 PM.

Late submissions will be accepted until Thursday, November 5, 2026, at 11:59 PM. A penalty of 1/7 of the earned grade will be deducted for each calendar day late.

Overview

In Unity Lab 1, you created a 2D scene containing a player that moves through keyboard input and a balloon that moves automatically.

In this lab, you will extend your Balloon Popper game by implementing shooting, collision detection, object interactions, scoring, and player feedback. We will begin implementing these mechanics together in class. You will then complete and test your game independently.

Part 1: Review Your Existing Project

Open your BalloonPopper project from Unity Lab 1.

Confirm that:

1. Your background and player are visible in the Game View.
2. The player moves left and right using keyboard input.
3. The balloon moves automatically and reverses direction at the screen boundaries.
4. Your Pin Prefab is saved in the project.
5. Your existing scripts run without errors.

Resolve any problems before continuing.

Part 2: Implement Shooting

Your player must be able to launch pins toward the balloon.

1. Create a C# script named **PinMovement**.
2. Implement movement so that a pin travels upward after being created.
3. Create a C# script named **PlayerShoot** and attach it to the player.
4. Allow the player to shoot by pressing the **Spacebar**.
5. Use `Instantiate()` to create a new instance of the Pin Prefab when the player shoots.
6. Make each pin appear at or slightly above the player's position.
7. Ensure that each pin moves independently after being launched.
8. Remove pins that travel beyond the visible play area so they don't accumulate indefinitely.

The player should be able to move and shoot during gameplay.

Part 3: Implement Collision Detection

Your game must detect when a pin hits the balloon.

1. Add appropriate **Collider2D** components to the pin and balloon.
2. Configure the collision interaction using triggers.
3. Add the necessary **Rigidbody2D** component or components so that Unity can detect the trigger interaction.
4. Create a script that responds when a pin enters the balloon's trigger.
5. Use tags or another appropriate method to distinguish pins from other `GameObjects`.
6. When a pin hits the balloon:
 - The balloon should pop or disappear.
 - Remove the pin that hit the balloon.
 - The interaction should not accidentally destroy the player.

Your collision system should respond only to the intended interaction between the pin and balloon.

Part 4: Add Scoring and Basic Game State

Your game must keep track of the player's score.

1. Create an integer variable that stores the current score.
2. Initialize the score to zero when the game begins.
3. Increase the score when the player pops a balloon successfully.
4. Display the score on the screen using a Unity UI text element.
5. Ensure that the score increases only once for each balloon popped.

After a balloon is popped, the player must be able to continue playing.

You may accomplish this by creating another balloon or resetting the existing balloon to an appropriate position.

The game should not permanently stop after the first successful hit.

Part 5: Add Player Feedback

The game should clearly communicate when the player successfully hits a balloon.

Implement the following:

1. The balloon should visibly disappear or pop when hit.
2. The score updates to reflect the successful hit.
3. Play a sound effect when a balloon pops.

You may also include additional visual feedback, such as a brief animation, particle effect, or color change.

Part 6: Test and Refine Your Game

Play your game and verify that all required mechanics work correctly.

Check that:

- The player moves left and right.
- The player stays within the intended gameplay area.
- The balloon moves automatically.
- The player can shoot pins using the Spacebar.

- Pins travel upward.
- Pins disappear when they leave the screen or hit the balloon.
- Pins correctly detect collisions with balloons.
- The player is not accidentally destroyed by their own pins.
- The balloon disappears or pops when hit.
- The score increases correctly.
- The game continues after a successful hit.
- The game provides clear visual and audio feedback.
- No errors prevent the game from running.

Optional Enhancements

If you finish the required mechanics, consider adding one or more of the following:

- Multiple balloons moving at different speeds.
- A balloon that gradually increases in size.
- Different point values based on balloon size.
- A timer or limited number of shots.
- Additional obstacles or distractors.
- A win condition or game-over condition.
- A simple start or restart button.
- Additional sound effects or animations.

These enhancements are optional and should not replace the required mechanics.

Submission

Due: Thursday, October 29, 2026, at 11:59 PM

You will submit your completed Unity game to the [CISC 3667 F26 Unity Lab Submission](https://itch.io/jam/fall-2026-unity-labs-1-2-submission) on itch.io.

Submission Instructions

1. Complete and test your Balloon Popper game, ensuring that all required mechanics from Unity Labs 1 and 2 are functional.
2. Build your game for **WebGL** using Unity's build settings.
3. Compress the contents of your WebGL build output folder into a **ZIP file**, ensuring that **index.html** is at the root of the ZIP.
4. Upload this file to itch.io as an **HTML game** and enable the option to **play it in the browser**.
5. Give your game an appropriate title and include a short description explaining its objective and controls.
6. Submit your game to the designated [CISC 3667 F26 Unity Lab Submission](https://itch.io/jam/fall-2026-unity-labs-1-2-submission).
7. Submit the link to your itch.io game page on **Brightspace**.

Important Notes

- Your game must be playable directly in a web browser.
- Test your published game on itch.io before submitting.
- Make sure the game is accessible to the instructor.
- You must submit your game to the itch.io Game Jam **and** submit its link on Brightspace to receive credit.
- Only one completed game is required for Unity Labs 1 and 2. There is no separate submission for October 22.