CISC 3667: Game Design and Development

In-Class Activity 9: Godot Lab - Scenes, Menus, and Persistent Data

Assigned: Thursday, November 5

Submission: Submit your playable game to the **CISC 3667 Godot Lab Game Jam** on itch.io and submit your game link and additional requirements on **Tuesday, November 10, 2026, by 11:59 PM.**

Late submissions will be accepted until **Tuesday, November 17, 2026, by 11:59 PM.** A penalty of 1/7 of the earned grade will be deducted for each calendar day late.

Overview

In Unity Labs 1 and 2, you developed *Balloon Popper!* by implementing player movement, object movement, shooting, collision detection, scoring, and player feedback.

In this lab, you will apply those game development concepts in **Godot**, another engine for developing 2D and 3D games.

You will create a 2D game called *Collect the Coins!* in which the player moves around the screen, collects coins, and earns points. You will expand on the gameplay mechanics from the Unity labs by implementing the following:

- **Animated sprites**
- **Timed collectibles**
- **A title screen**
- **Scene transitions**
- **A results screen**
- **A persistent high-score system**

We will work through the development process together in class using the provided step-by-step tutorial. You will then complete, test, and publish your game.

Provided Assets

The following assets are available for your game:

- **Title Screen Background:** Background image for the game's main menu, featuring the *Collect the Coins!* title and bear character.
- **Grass Background:** Background for the gameplay scene.
- **Gold Coin:** Collectible object.
- **Bear Sprite Sheet:** Character sprite sheet for walking animations.

You may use the provided assets or create your own visuals. The required gameplay mechanics remain the same.

Part 1: Create Your Godot Project and Gameplay Scene

Create a new Godot project named **CollectTheCoins**.

Organize your project using the following folders:

- **Scenes**
- **Scripts**
- **Assets**

Create and save a gameplay scene named **Game.tscn**. Set up the gameplay environment with a background, player, collectible coins, and a score display.

Your scene should use appropriate Godot nodes and a clear scene hierarchy.

Part 2: Implement Player Movement and Animation

Create a reusable player scene named **Player.tscn**. Your player must:

- Respond to keyboard input using the arrow keys or WASD.
- Move in all four directions.
- Use an adjustable movement speed.
- Include an appropriate collision shape.
- Play directional walking animations while moving.
- Return to an idle animation when movement stops.

Use the provided bear sprite sheet or your own character artwork.

Part 3: Implement Coin Collection and Timed Gameplay

Create a reusable coin scene named **Coin.tscn**.

Your game must:

- Display multiple collectible coins in the gameplay area.
- Detect when the player touches a coin.
- Remove a coin after it is collected.
- Prevent the same coin from being collected more than once.
- Increase the player's score when they collect a coin.

Add a time-based challenge to your game:

- Coins should spawn automatically every **3 seconds** at randomly selected spawn points.
- Each coin should disappear after **8 seconds** if the player does not collect it.
- The round should last **60 seconds**, giving players a limited amount of time to collect as many coins as possible.
- When the round timer reaches zero, the game should transition to the results screen.

Part 4: Implement Scoring and Game State

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

Implement the following:

- A current score that begins at zero for each new round.
- A score display that updates whenever the player collects a coin.
- A countdown timer that ends the round when time runs out.
- A final score that can be displayed after the round ends.

The game should recognize when the round is complete and prevent additional scoring.

Part 5: Create a Title Screen and Scene Transitions

Create a **MainMenu.tscn** scene that serves as the game's title screen.

Use the provided **Title Screen Background** image as the background for your main menu.

The image already includes the game title, *Collect the Coins!*, and the bear character, so you do not need to create separate title text or add another character image.

Include:

- A Start Game button.
- An Instructions button.
- A Scores button.

Connect the buttons so that players can navigate to the appropriate scenes.

Configure the project to launch the title screen when the game starts.

Part 6: Create a Results Screen and Restart Option

Create a **GameOver.tscn** scene that appears when the round ends.

Include:

- A message indicating that the round is complete.
- The player's final score.
- A Play Again button.
- A Main Menu button.

Make sure that starting a new round resets the current score, timer, and collectibles.

Part 7: Implement Persistent High-Score Data

Your game must save the player's highest score so that it remains available after the game is closed and reopened.

Implement the following:

- Initialize the high score to zero if no saved record exists.
- Compare the final score with the saved high score when a round ends.
- Update and save the high score when the player achieves a new record.
- Load the saved high score when the game starts.
- Display the high score on the Scores screen.

Use Godot's file-access functionality to save and load the data. You may use JSON and a file path such as `user://save_data.json`.

The current score should reset between rounds, but the high score should persist between sessions.

Part 8: Test and Refine Your Game

Play through your game and verify that the required features work correctly.

Check that:

- The title screen appears when the game launches.
- The player moves in all four directions.
- Walking and idle animations work.
- Coins disappear when collected or when their time expires.
- The score increases correctly and only once per collected coin.
- The round timer works and ends the round.
- The results screen displays the final score.
- Players can restart or return to the main menu.
- A new round begins with a reset score and fresh collectibles.
- The high score saves and loads correctly.
- No errors prevent the game from running.

Part 9: Export and Publish Your Game on itch.io

Export your completed game for the web and publish it on itch.io.

1. Configure a web export preset in Godot.
2. Install the required export templates if prompted.
3. Export your game and include all required files.
4. Create an itch.io project page for your game.
5. Upload the web export and configure it to run in the browser.
6. Provide a title and short description explaining the objective and controls.
7. Submit your game to the designated CISC 3667 Godot Lab Game Jam.

Game Jam Link: https://itch.io/jam/cisc-3667-f26-godot-lab-submission

Test the published game in a browser before submitting. Confirm that the gameplay, scene transitions, scoring, and high-score system work as expected.

Optional Enhancements

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

- A second level with more collectibles or a shorter time limit.
- Different types of coins with different point values.
- Moving hazards that the player must avoid.
- A pause menu.
- Sound effects and collection animations.
- A visual indicator showing how many coins remain.
- A new high-score notification.

Optional enhancements should build on the required game rather than replace its core mechanics.

Submission

Due: Tuesday, November 10, 2026, by 11:59 PM

You will submit your completed Godot game to the **CISC 3667 F26 Godot Lab Submission** (https://itch.io/jam/cisc-3667-f26-godot-lab-submission) **on itch.io**.

Submit the following:

1. A **link to your published itch.io game**.
2. A **ZIP file containing your complete Godot project**, including scenes, scripts, and assets.
3. A **short written reflection** answering the following questions:
    - **How are Godot scenes and nodes different from Unity scenes and GameObjects?**
    - **How does your game detect when the player collects a coin?**
    - **How does the time limit affect the player's gameplay experience?**
    - **What is the difference between the current score and the persistent high score?**
    - **Which part of working in Godot was most different from working in Unity?**

Important Notes

- Your complete Godot project must open without errors and be playable.
- Your published game must run directly in a web browser.
- Submit both the published itch.io game link and the complete Godot project ZIP.
- Optional enhancements do not replace the required features.