

# Final Project

*Let's make "Antarctic Adventure"!*

EEE3313: Introductory Digital Labs, Spring 2023

**Announcement:** Friday, May 26, 2023

**Due:** 23:59, Wednesday, June 21, 2023

**Late Submission is not allowed**

## 1. Introduction

The goal of this project is to evaluate your programming skills by creating a simple running game called "Antarctic Adventure." You will have the opportunity to review the concepts covered throughout the semester and apply them in this project. Please feel free to use any relevant class materials from previous sessions. For reference and inspiration, you can visit [<https://www.youtube.com/watch?v=F3MuCv-MPOA>]. This video will provide you with a better understanding of the game's concept and mechanics.

## 2. Project Specification

The project is divided into multiple steps, each involving the implementation of a specific feature from Antarctic Adventure. By successfully implementing the functionality as specified in each step, you can earn corresponding points. Your project score will be determined by the total sum of points you earn throughout the completion of these steps.

### ☐ **Step 1. Display game images (10 points)**

Make your project look like a game. You need to display background graphics, a runner (penguin), obstacles, and coins (fishes). There is no strict restriction on details of the images as long as they can be distinguished. You should also display your student ID in the upper right corner.

#### **To-Do:**

- Display background graphic (road and sky)
- Display a runner
- Display obstacles
- Display coins
- Display your student ID

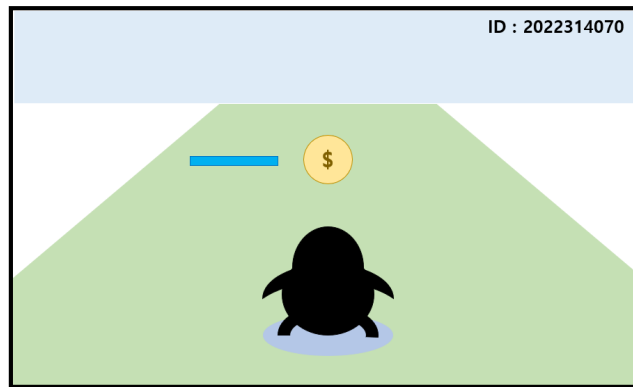


Figure 1: Example for Step 1

### ☐ Step 2. Make background effect (10 points)

To create the illusion of the runner moving forward along the road, incorporate visual effects on the background graphics. Display a series of background images in sequence repeatedly, giving the impression of continuous movement.

#### To-Do:

- Implement background graphic effects (10 points):

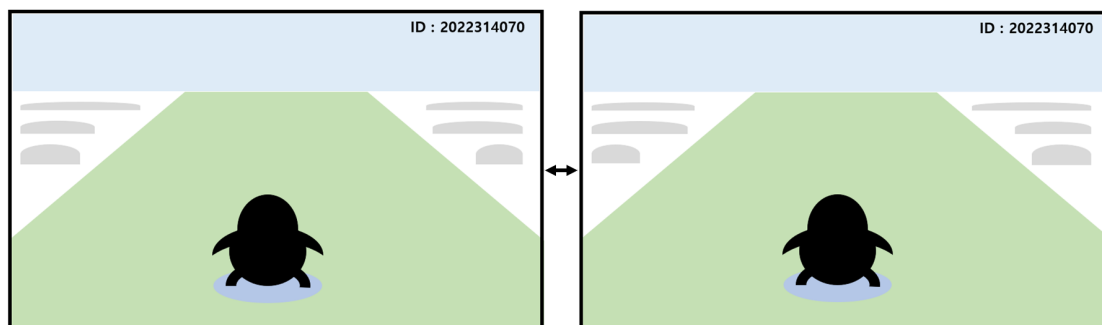


Figure 2: Example for Step 2

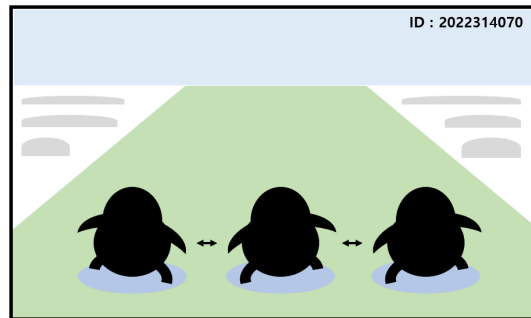
### ☐ Step 3. Make the runner move (20 points)

In this game, the runner runs forward on a fixed horizontal line with the ability to move left and right. It can also jump to overcome obstacles. When a push button is pressed, the corresponding movement should be performed. You are allowed to make the runner run either on any position of the line or on just three positions (left, right and middle). When the runner jumps, you should keep displaying the shadow on the line to make it look natural.

#### To-Do:

- Implement movements of the runner (10 points):
  - Run: Run on three positions: left, right and middle
  - Jump: Jump to the upper point of a fixed distance and fall down to the ground in a constant time.

- The shadow should be displayed on the line following the horizontal position of the runner.
- Use different images for each movement (10 points)
  - For running, switch at least two images on the runner's position repeatedly.
  - In total, you should use at least three images.



Run	
Jump	

Figure 3: Example for Step 3

#### □ Step 4. Make moving obstacles and coins (20 points)

The goal of this game is to earn more coins in a limited time while dodging incoming obstacles. As the runner runs on a fixed horizontal line, obstacles and coins should come from the end of the road to the runner's position. Make the obstacles and coins come on three lanes: left, right and middle. Obstacles and coins from the left and right lane should move in a diagonal direction to create a sense of perspective. The size of incoming objects should be increased as they come closer to the runner. The obstacles and coins can have any shape as long as they can be distinguished from each other. For example, you can represent obstacles as blue rectangles and coins as yellow rectangles.

#### To-Do:

- Display obstacles and coins which approach the runner in three lanes (10 points)
- Make the obstacles and coins get larger as they come closer (10 points)

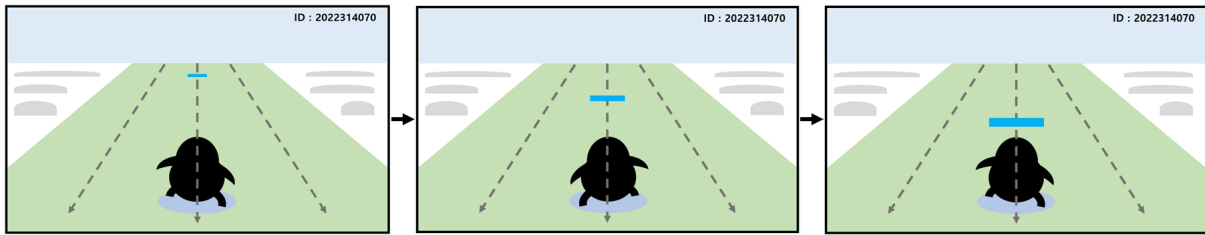


Figure 4: Example for Step

#### ☐ **Step 5. Make effects of coin (10 points)**

As the runner takes coins, the player should be able to check how many coins have been earned so far. Count the score as a number of obtained coins and display the current score on the left of your student ID. Also there should be some effects to inform the game player about what happened. You need to implement a sound effect for taking coins and a graphic effect for taking a coin. Here, the definition of graphic effect is displaying a new image layer, not just removing an existing image.

##### **To-Do:**

- Display current score on the screen
- Taking a coin increases the score by one
- A sound effect for taking coin
- A graphic effect for taking a coin

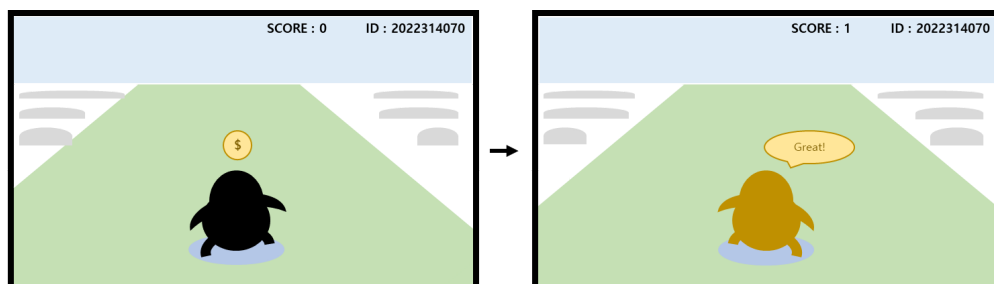


Figure 5: Example for Step 5

#### ☐ **Step 6. Make effects of obstacles (10 points)**

You need to count how many times the runner has collided with obstacles. If the runner collides, the number of lives is decreased by one. When the runner collides with an obstacle, there should be some effects to inform the game player about what happened. You need to implement a sound effect for taking coins and a graphic effect for colliding with obstacles.

##### **To-Do:**

- Display current number of lives
- Collision with obstacles costs one life
- A sound effect for obstacle collision
- A graphic effect for obstacle collision

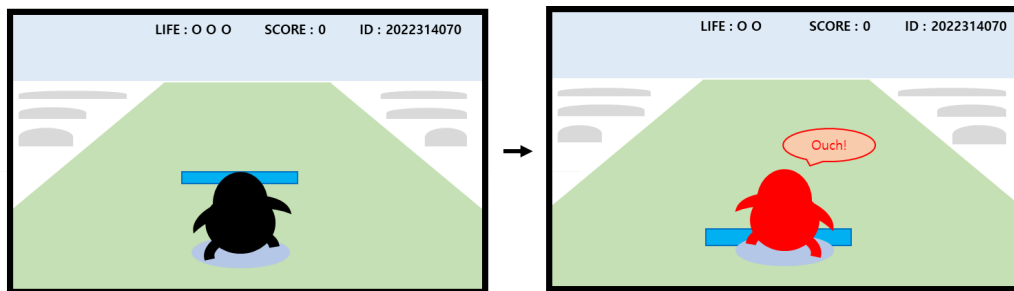


Figure 6: Example for Step 6

#### ☐ Step 7. Make end point (10 points)

You may already have implemented the life system in step 6. If the number of lives becomes zero, the game becomes a finish-state.

Also, the game is played with a fixed distance(hundreds of meters). The distance decreases as the runner runs. If the remaining distance becomes zero, the game also becomes a finish-state.

In the finish-state, all objects stop, and a message “FINISH” is displayed in the center.

##### To-Do:

- Display the current remaining distance
- Decreasing distance as the runner runs
- Stop all objects and display “FINISH ”when the distance or life becomes zero

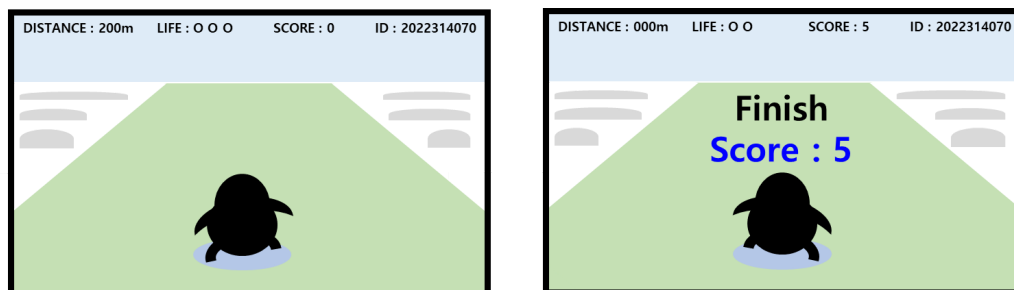


Figure 7: Example for Step 7

#### ☐ Step 8. Make background music (10 points)

Background music is essential for games! Play background music when the game is started. You can use any music.

##### To-Do:

- Play background music

### 3. How to write a report

#### ☐ Basic Rules

- within ten pages  
(not strict constraints but ten pages are enough with essential elements)
- write the report in English or Korean

#### ☐ Format of contents

You may not independently implement each step in your project, but you should describe each step independently with corresponding subheadings in your report. For example, a subheading can be like 'step 1', 'step 2' . . 'step 8'. Under the subheading, you should describe the code and algorithm you implemented for the corresponding step. Please do not simply copy and paste your code and explain it line-by-line, most of which are meaningless. It is better to explain your code in a larger scope. Since a template or skeleton code is not provided for this project, you may start by explaining how you structured the code and how you implemented such components. If you want to explain with code, it is recommended to use code snippets with corresponding source file names and code line numbers. In addition, if you first made or modified the block diagram, you have to show the block diagram, describing which part you added to the block diagram and explaining what block you used and why you added such blocks for the diagram. For steps with several states, you should draw FSM and explain the rationale of your design.

### 4. Hand-In Instructions

You need to submit three things on LearnUs

#### 1. Exported Vivado and Vitis Project

- 1) Vivado:
  - A. Click File > Project > Archive
  - B. Include all files and click OK
  - C. Submit the generated “~.xpr.zip” file
- 2) Vitis:
  - A. Click File > Export
  - B. Check the system and platform, and click OK
  - C. Submit the generated “~.ide.zip” file

#### 2. Demonstration Video

<student\_id>\_video.mp4 (<student\_id>\_video\_<number>.mp4 in several files)

#### 3. Result report in pdf form

<student\_id>\_report.pdf

As this project is designed to implement several components to a single project, you only need to submit your final version (everything-implemented version) of the source code.

You should properly demonstrate the functionality of each step of the project in the video. If you want to submit multiple videos, you must add "\_<number>" at the end of the video name. The total length of videos should be within 5 minutes.

## **Caution!**

### **Do Not Lie!**

We will regenerate the bitstream with the source files in your project, and check the behavior of your design manually. If the functionalities of the submitted project and the result report do not match, or the project cannot be launched on the PYNQ board, you will receive a zero point.

### **Do not copy others' code! Plagiarism is strictly prohibited.**

In case of cheating, all involved students will receive a zero point in the project, and some may get an F grade for this course. In addition, there may be further punishment at the department or college level concerning its severity.