</talentlabs>

Lab 1 Building Mini Golf Game with MIT App Inventor

Instructions

1. Answer the below question in the boxes if needed.
2. Code on your computer and zip all your code before submission.
3. Please submit the assignment through TalentLabs Learning System.

Part 1 Finish the Simple Ball Bounce Game

Task 1.1 Setup your machine for developing apps with MIT App Inventor

Please follow the setup guide in the website below to set up your development environment with MIT App Inventor.

Option 1: With Android Phone (Recommended)
https://appinventor.mit.edu/explore/ai2/setup-device-wifi

Option 2: Without Android Phone (It might run slow on some computers)
https://appinventor.mit.edu/explore/ai2/setup-emulator

Make sure you can run an app and test your program based on the instructions in the link above.

Task 1.2 Finish the Ball Bouncing Game

In lecture video 1.4-1.5, we have demonstrated how to build a simple ball bouncing game using MIT App Inventor. Please follow the steps in the video, and replicate the same app.

The end product should have the following feature:

1. When the user fling the ball, the ball should move (Video 1.4)
2. When the ball reach the edge of the screen, the ball should bounce back (Video 1.5)

</talentlabs>

Part 2 Building Mini Golf Game

Task 2.1 Make Bouncing Game to Mini Golf Game

In this section, you are going to upgrade your ball bouncing game from Task 1.2 to a Mini Golf Game following the steps below. If you don't understand the code below, it's perfectly ok as we are going to cover the concept in the coming classes. This exercise is really just to give you a taste in developing software (build -> test -> modify/fix bug).

Steps:

1. Open up your project in Task 1.2.
2. We will need to make some modifications based on your previous work. Your app should have the following components, and properties. Please check and add/modify your components.

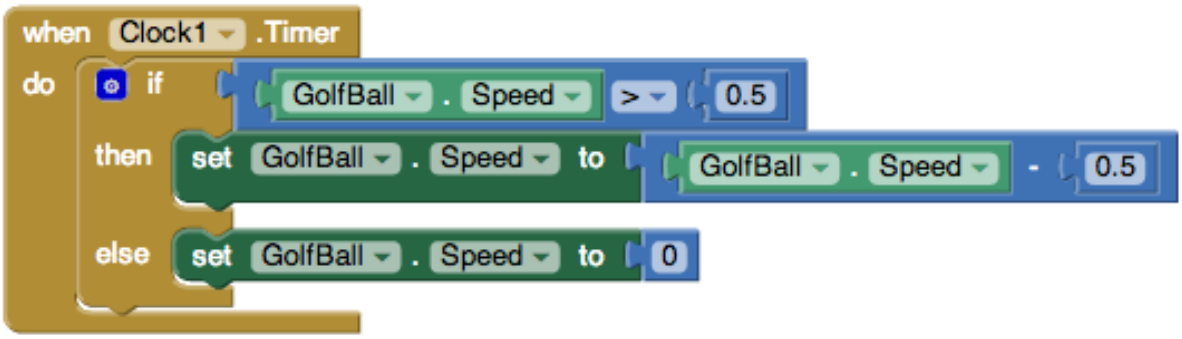
Component Type	**Palette Group**	**What You'll Name It**	**Purpose**	**Properties**
Canvas	Drawing and Animation	Canvas1	The canvas serves as the golf course	Height: 300 Width: FillParent BackgroundColor: Green (or whatever you like!)
Ball	Drawing and Animation	GolfBall	This is the ball the player will fling to try to hit the Hole	Radius = 10 Color: White (or your choice!) Speed: 0 Interval: 1 (ms) Z = 2 (when sprites are overlapping, the one with the higher z will appear on top)
Ball	Drawing and Animation	Hole	This will be the target for the GolfBall	Radius = 15 Color: Black Speed: 0
Clock	Sensors	Clock1	The clock will fire continuously to control the movement of the ball	Timer Always Fires Timer Enabled TimerInterval: 100

3. Once you have the components there, then you can start the coding part.

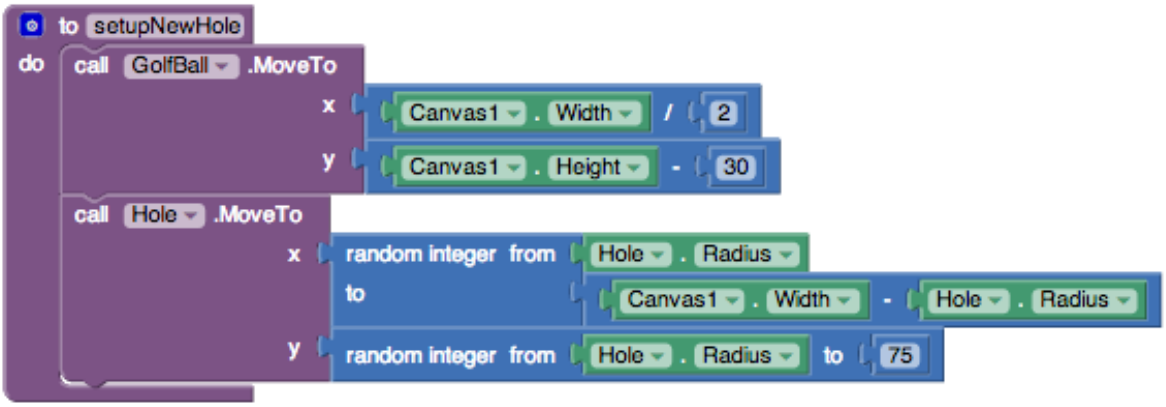
   Use timer event to slow the ball down so it doesn't bounce around forever.

   Each time the clock fires, it will reduce the speed of the ball slightly. Notice that if the ball is not moving then these blocks will do nothing. If you don't have this then the ball will just bounce forever.

   You'll need to use the if mutator function to change the if block into an if-else block.



4. Program a new procedure called SetupNewHole.
   This procedure will be called when a hole is scored and the ball has to be placed back at the starting point. Note that the Hole.MoveTo block sets the hole up in a new random location for the next play.



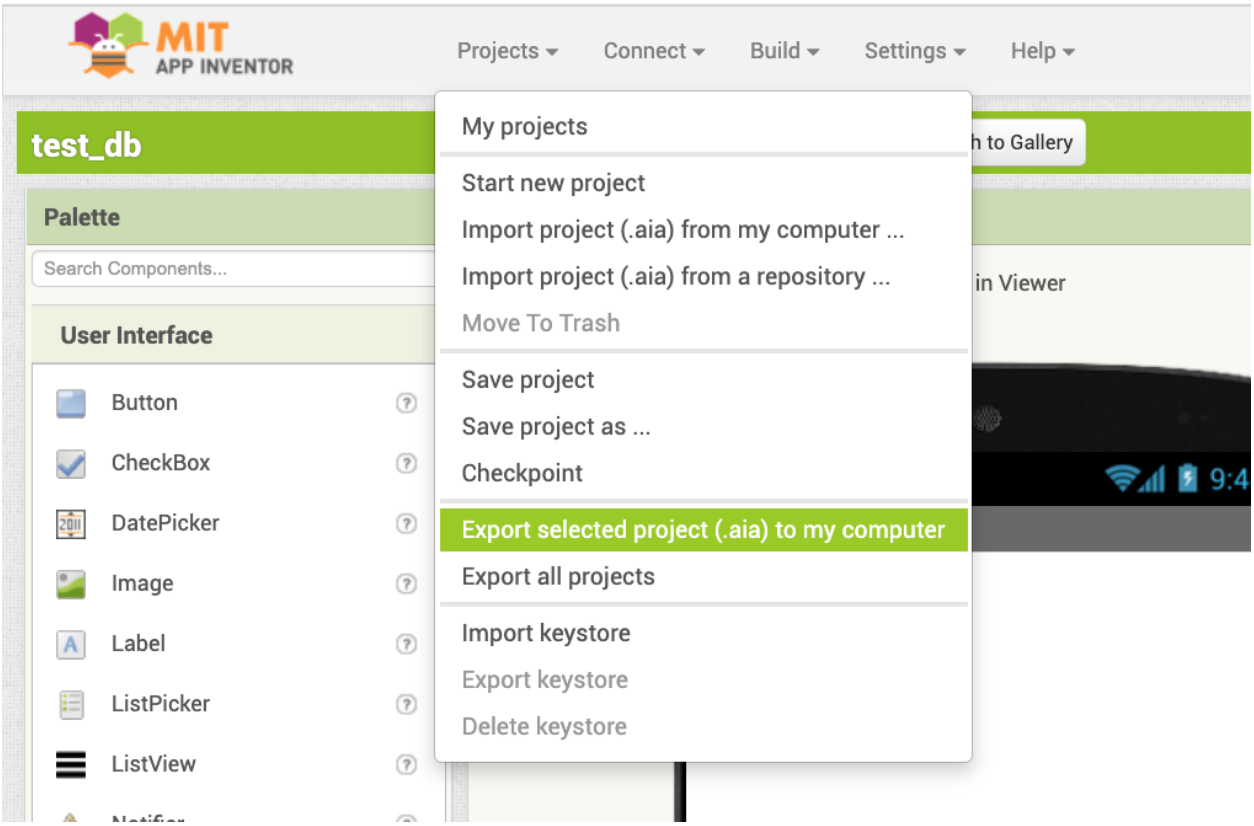
5. Program the Behavior of the Hole: When the ball collides with the hole, the ball disappears and resets at the bottom of the screen.
   Note: When you first drag out the GolfBall.CollidedWith event handler, the named parameter is called "other". Notice that the if then block tests to see if the object involved in the collision with the golf ball (other) is the black ball sprite representing the hole. You can't just put a text block with the word "Hole" in it, you must use the Hole block, that can be found in the drawer for the Hole image sprite. Do not use a text block here.

6. Test the App to make sure it works.

Task 2.1 Exporting the Project and Submission

Steps:

1. Export the project as .aia file by pressing “Projects -> Export selected project (.aia) to my computer”



2. Save the exported .aia file and upload it to TalentLabs Learning Management System.