CET1012 - Programming Methodologies: Java - Practicum 01

Topics Covered: Computational Thinking, Problem Definition, Abstraction, Algorithms, Flowcharts, Pseudocode

Learning Objectives:

- Apply computational thinking to solve complex problems
- Apply decomposition to break down the problem into smaller problems
- Apply the concepts of algorithm flowchart

Deliverables:

- Submit a single Scratch project file called `CET1012_P01_<Your_Name>.sb3` (e.g. `CET1012_P01_John_Doe.sb3`)

Overview

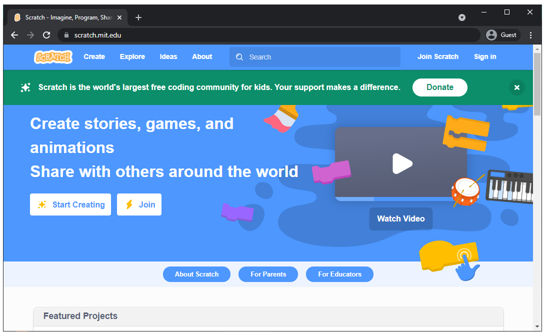
When solving problems, a step-by-step approach would be ideal. However, real world problems tend to be big and complex. Not all problems can be solved systematically. Fortunately, there are strategies, heuristics, and good practices that previous problem solvers have used. In this practicum, you will be using computational thinking to create a very simple top-down perspective game.

This practicum is made up of several problem sets. Each problem set comprises of a small feature that will be used in your game. You are expected to use Scratch, a simple visual programming language to solve the problems. Scratch will also serve as an introduction towards other formal programming languages such as Python, Java, or C/C++, that you will be using throughout the course.

Introduction

Note that this practicum is not an introduction to Scratch. There are many resources available online for you to get started. You may find it surprising that scouring the internet to find answers to programming-related questions is a very common thing that developers do. In fact, most of the time, it is also common that the questions you seek answers for, have already been asked (and answered) by developers before you. However, it is not as simple as asking, "how to create a game?". Like building a house, the problem often can be broken down into smaller questions like "how to build a room" and in turn will ask even more fundamental questions like "how to build a wall", and so on. As you have learnt, this is known as *decomposition*. Breaking down the problems and asking the right questions is an acquired skill and we hope that this practicum will serve as a good platform to get you started.

The first thing you would want to do is to access Scratch using https://scratch.mit.edu/.

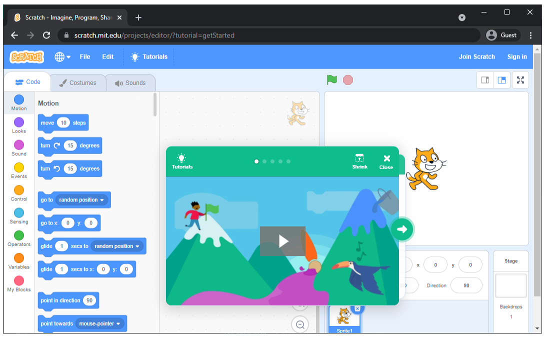


Scratch is the world's largest free coding community for kids. Your support makes a difference.

Create stories, games, and animations
Share with others around the world

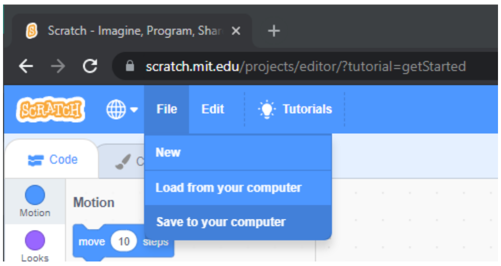
At this stage, it is recommended that you create an account in Scratch. This gives you the option of accessing your saved projects online. For those that opt to not create an account, you have to ensure to save your projects locally to your computer. Regardless of which option you choose, it is always considered a best practice to regularly save your projects. The last thing you want is to lose all the progress you have made.

Next, you would want to create a project. Click on `Create` at the top of the page. This will bring you to the editor page.



This will be the page that you will use to develop your game. You are encouraged to explore the tutorials provided and play around with the tools and features in Scratch.

At the end of the practicum you are expected to save your project. Rename the file to `CET1012_P01_<Your_Name>.sb3` before submission. Failure to do so will result in a penalty.

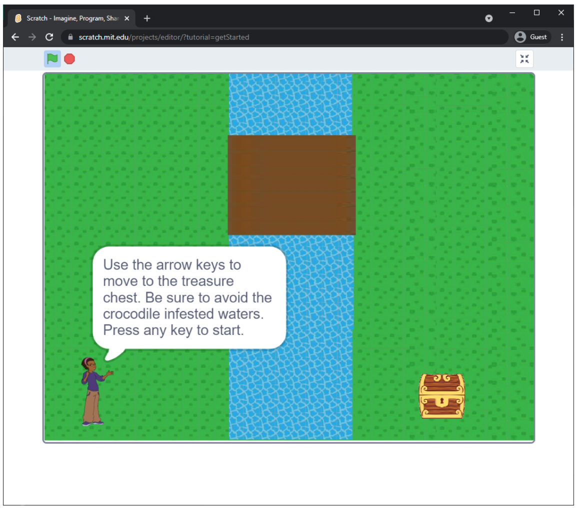
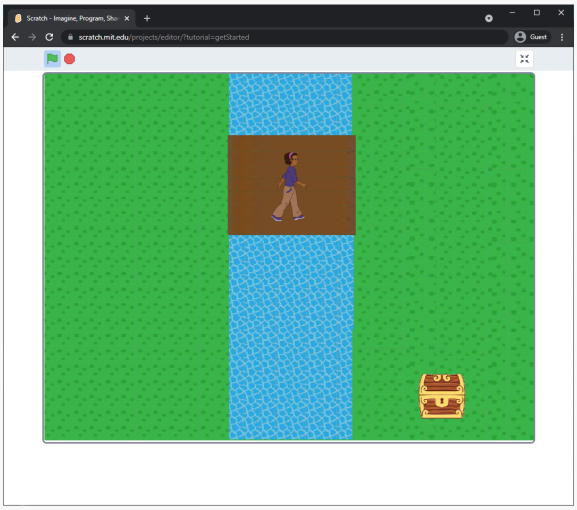


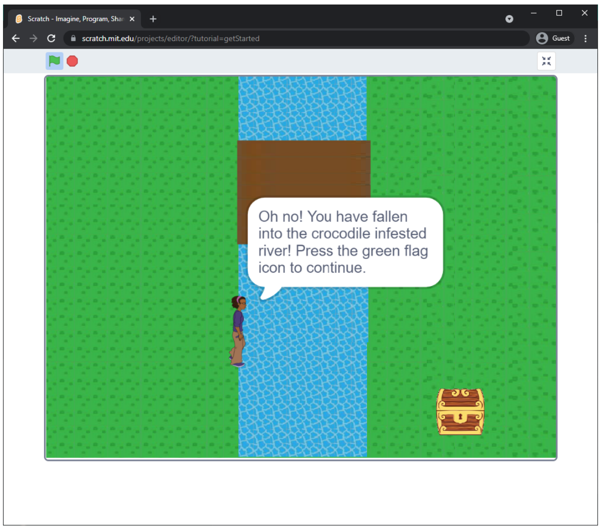
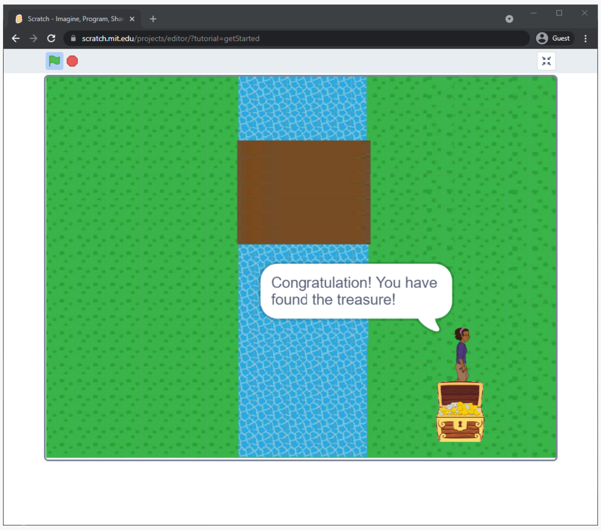
Define The Problem using the Game Objectives and Features

When given a complex problem, the first step is to define the problem. If you are to build a house, some of the problem definition includes identifying type of the house you want to build and the number of rooms required. If you are to bake a cake, some of the problem definition includes identifying the flavor and the serving size of the cake. In the case of a game, the problem can be defined in the objectives and also the features that is needed in the game.

The objective of the game is to reach the treasure chest. Players will control the movement of the character using the arrow keys. However, the treasure chest is located across a crocodile infested river and the only way to cross the river is over a bridge.

The table below shows some of the features available in the game.

Screenshot	Description
 Scratch - Imagine, Program, Sha scratch.mit.edu/projects/editor/?tutorial=getStarted Guest Use the arrow keys to move to the treasure chest. Be sure to avoid the crocodile infested waters. Press any key to start.	This is the start of the game. To start the game, click on the green flag. At the start of the game, the character will be positioned on the bottom left of the map. A simple message will be shown to tell the players of the objective of the game.
 Scratch - Imagine, Program, Sha scratch.mit.edu/projects/editor/?tutorial=getStarted Guest	Controlling the character using arrow keys to cross the bridge.

Screenshot	Description
 Oh no! You have fallen into the crocodile infested river! Press the green flag icon to continue.	If the character falls into the crocodile-infested river, a message will be displayed to indicate the game has ended. This marks the end of the game.
 Congratulation! You have found the treasure!	Upon reaching the treasure, the treasure chest will open and a message will pop up to congratulate the player. This marks the end of the game.

As you can see, while the game may seem very simple, it is in fact made up of many features. In this practicum, you are to implement the above features. To assist you, we have broken down the problem into multiple parts called problem sets. For each problem set, you will tackle a small feature of the game. You will then combine all the different features to complete the game as the final problem set.

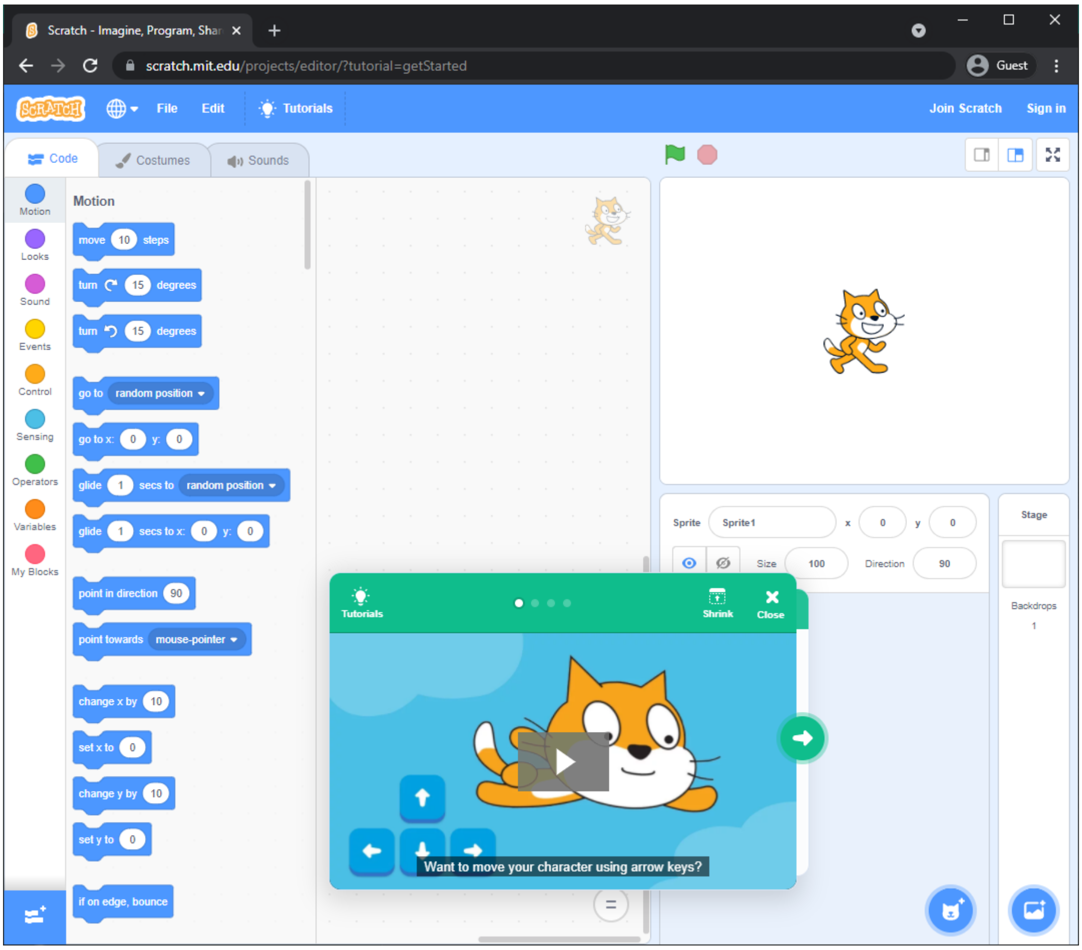
Also note that there are multiple ways to solve the problem sets. While there may be more optimal ways to solve a problem, it is not part of the requirements of this practicum.

Problem Set 1 - Move a Sprite

For your first problem set, you are expected to move a sprite using the arrow keys.

A *sprite* is an object that performs actions in a project. You may use the default cat sprite to solve the problem. Note that the type of sprite is irrelevant to the problem you are trying to solve. This is also a form of *abstraction*. If you are having difficulty, the solution is already available as a tutorial in Scratch, you just have to find it. Also, take this opportunity to familiarize yourself with the Scratch interface.

There is no submission for this problem set.



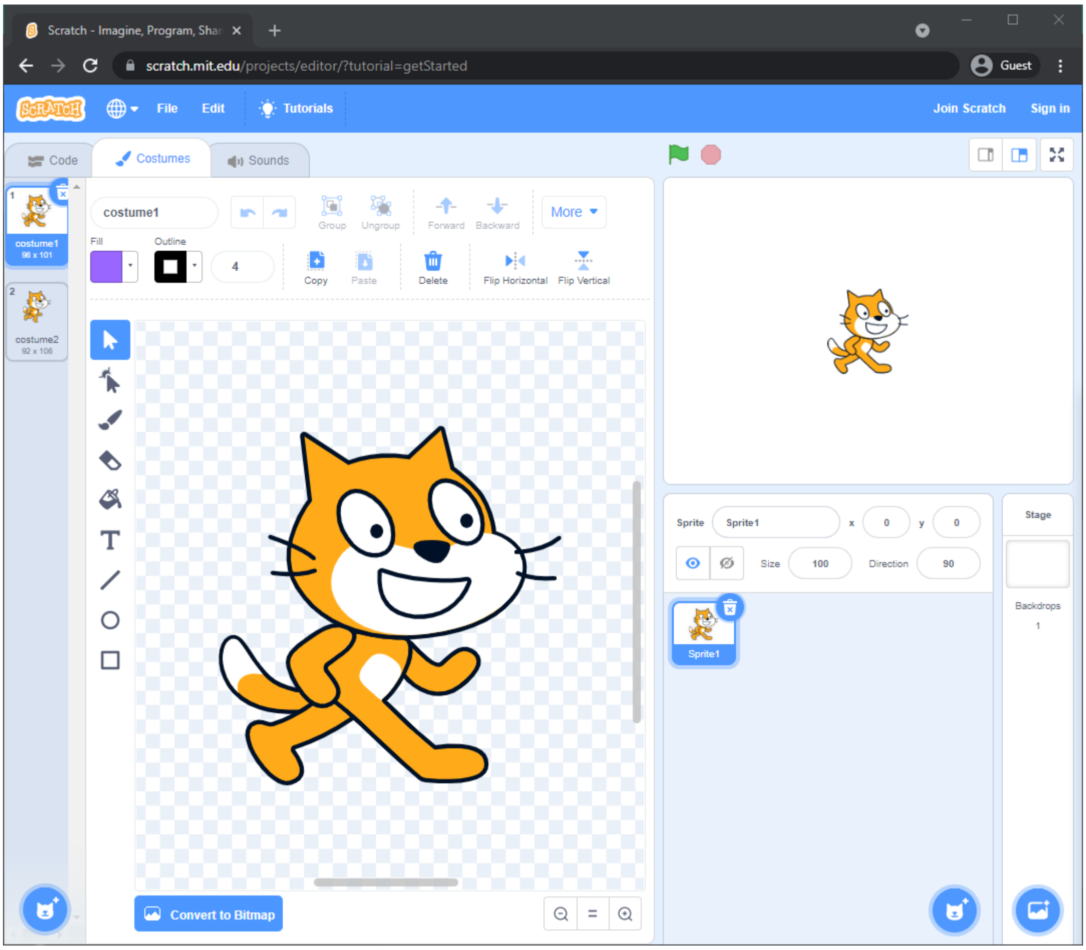
Want to move your character using arrow keys?

Problem Set 2 - Walking/Running Animation

For this problem set, you will learn how to animate a running cat by changing the costume of the sprite.

A *costume* is one out of many "frames" or alternate appearances that a sprite may take. Sprites can change their look to any of its costumes. They can be named, edited, created, and deleted, but every sprite must have at least one costume. One of the most common uses of costumes is to make an animation for a game.

Navigate to the "Costumes" page, you will see that the cat sprite has two costumes: `costume1` and `costume2`.



By toggling between costumes, we can animate a running cat. Similarly, there is a tutorial to animate the sprite by changing costumes.

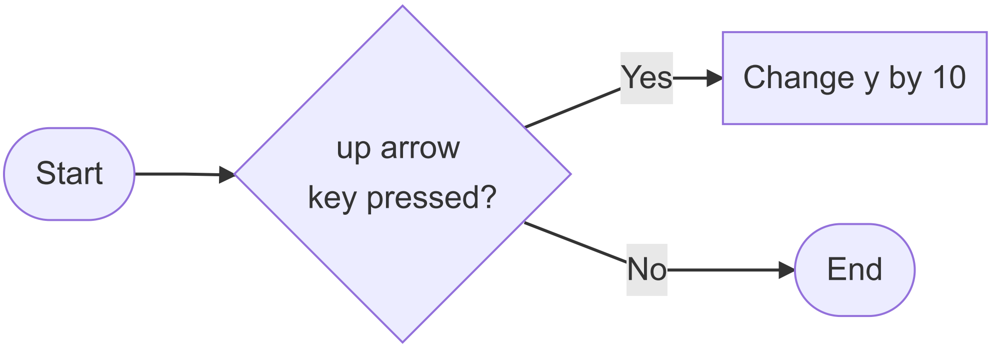
Now, combine the running animation with the moving sprite from problem set 1. You are not required to change the orientation of the cat based on the direction it is going i.e. it will still face to the right even if it is moving left. Once you have completed, your sprite should appear running as you control it with the arrow keys.

There is no submission for this problem set.

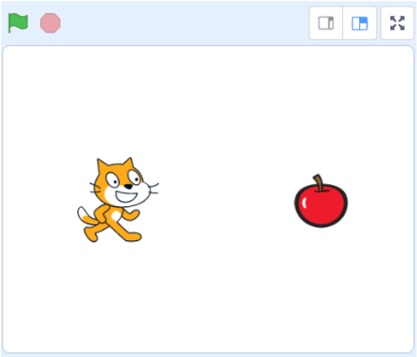
Problem Set 3 - Conditional Event

For this problem set, you will learn how your sprite can interact with its surroundings such as other sprites or even the backdrops. This can be achieved through a conditional event where an action is taken based on the decision made. Conditional events are a staple tool in a developer's toolkit. A common example is the `if-else` statements in programming languages. Conditional events are very useful for creating events that trigger based on different conditions such as when the sprite touches another object.

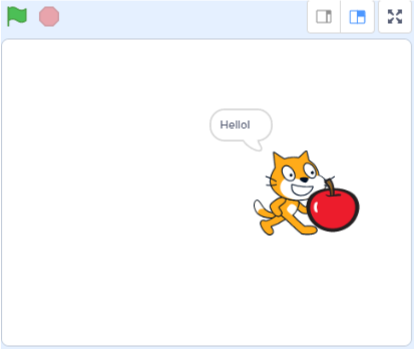
In fact, you have already encountered an example of a conditional event. Moving a sprite in problem set 1 is an example of a conditional event - the sprite will move in the direction based on which arrow key is pressed. We can represent the sprite moving process in an *algorithm flowchart* as shown below.



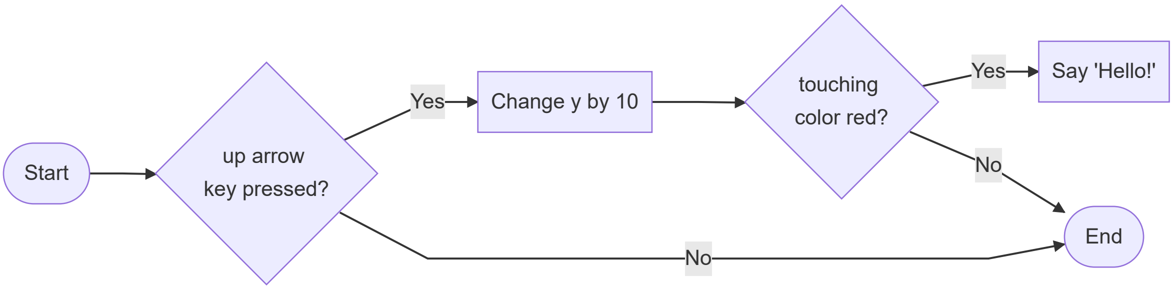
Building on top of the previous problem sets, you will choose the apple sprite from the default sprite library. You may reposition the apple anywhere on the Stage. An example is shown below.



The main task of this problem set is for the cat to say a message when it touches the apple.



To assist you, refer to the following flowchart. It is an extended version of the moving sprite flow chart.



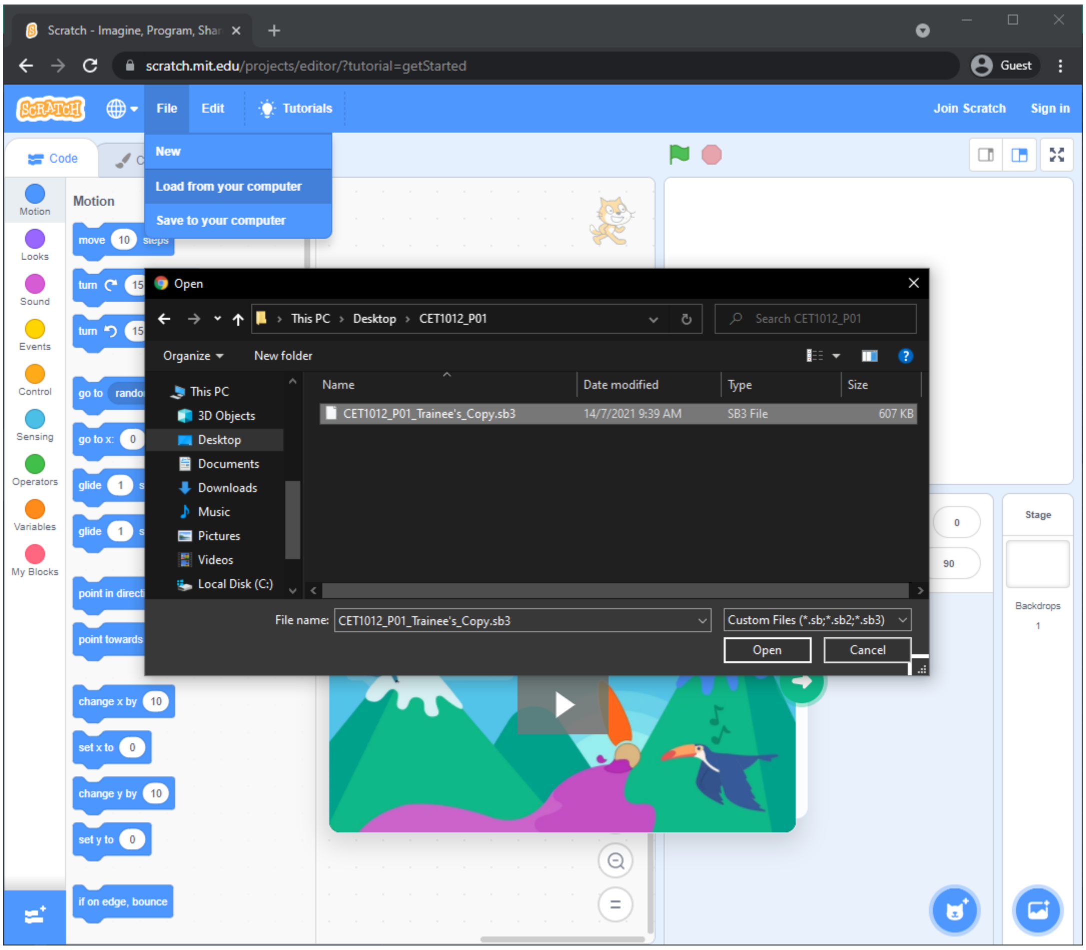
One of the ways to achieve this, is to use the 'broadcast' code block under the 'Events' code type. Using the 'broadcast' code block is also a known solution from errors that may arise if multiple of the 'When Green Flag clicked' block is used. This may be useful for the final problem set

There is no submission for this problem set.

Problem Set 4 - Creating the Game

For this problem set, you will combine all concepts learnt from the previous problem set to be applied to the game.

Open the `CET1012_P01_Trainee's_Copy.sb3` file in Scratch.



Name	Date modified	Type	Size
CET1012_P01_Trainee's_Copy.sb3	14/7/2021 9:39 AM	SB3 File	607 KB

File name: CET1012_P01_Trainee's_Copy.sb3

Custom Files (*.sb;*.sb2;*.sb3)

In the file there are 3 sprites,

- Avery (1 costume)
- Avery Walking (4 costumes)
- treasure (2 costumes)

and 1 backdrop.

You are only allowed to use the sprites and backdrop provided. Do not modify or delete the sprites and backdrop.

Each sprite have its own set of code blocks. You may add new code blocks but you are not allowed to delete the existing code blocks.

They are mostly to initialize the sprites when the game starts. The game starts when the green flag is pressed.

There are also the 'say' code blocks. You are to use the 'say' blocks provided. You may refer back to the screenshots to see their usage.

For the maximum allocation of marks, refer to the table below.

Description	**Marks (%)**
Smooth and controlled movement of Avery Walking using the arrow keys	20
Smooth Avery Walking animation	20
Successful 'say' triggers upon falling into crocodile-infested river and upon reaching the treasure.	20
Successful opening of treasure animation upon reaching the treasure.	10
Successful hide/show of sprites.	10
Successfully ends the game upon falling into crocodile-infested river and upon reaching the treasure. A game is considered to have ended if pressing any key does nothing.	10
Game operates smoothly with no major errors or bugs.	10

Once you have completed, save and rename the file in the following format `CET1012_P01_<Your_Name>.sb3` e.g. `CET1012_P01_John_Doe.sb3`.