COMP1023 Software Engineering

Spring Semester 2019-2020

Dr. Radu Muschevici

School of Computer Science, University of Nottingham, Malaysia

Lab 1

2020-02-11 & 12

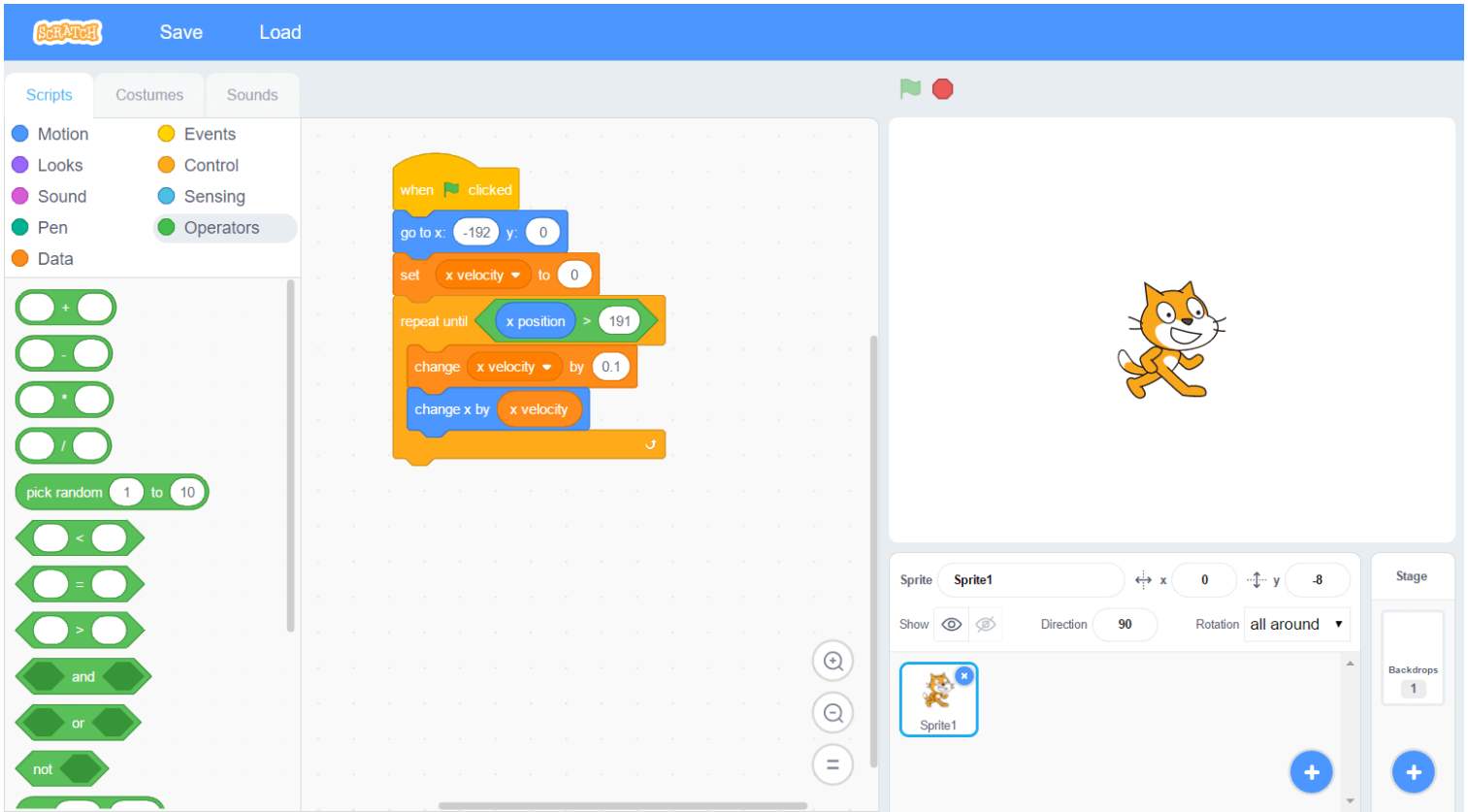
The Lab

- Each week, an exercise will be posted on Moodle, which you should complete that week.
  - Exception: for the first exercise (starting today), you have time until end of the **following** week (Week 24).
- Towards the end of the semester (Week 32), each group will submit a portfolio of their lab exercises.
- The mark for this constitutes 50% of the module mark.

- **Topic:** develop a game in Scratch.
- **Objective:** practice applying software engineering methodologies and processes.

What is Scratch?

- A visual programming language.
- Runs in a Web browser (also available as standalone app for Windows, MacOS).



1st Exercise – Lab 1 – Week 23

Individually…

- Create a Scratch account & log in: `https://scratch.mit.edu`
- Watch the Getting Started with Scratch video.
- Play around in the development environment to get a feel of programming in Scratch.

As a group…

- Decide whose account you will use for the class project.
- Discuss & agree on what type of game you want to develop together.
- Work on a set of **requirements** for your software project.

Requirements Engineering – a Short Introduction

- Detailed introduction of Requirements Engineering in the lecture.
- Today only the basics, so you can get started.

Requirements Engineering – Types of Requirements

Functional

- Services (features) the system should provide; how the system should react to particular inputs; how the system should behave in particular situations. **What the system will do.**
- Example:
  - An email client supports the IMAP protocol for fetching emails from a server.

Non-Functional

- Qualities of (constraints on) the services or functions offered by the system (related to timing, runtime environment, development process, standards, …) **How the system will be.**
- Example:
  - The system is able to handle 1000 service requests per second.

Requirements Engineering – Activities

1. **Elicitation:** gathering and discovery of requirements from stakeholders and other sources.
2. **Analysis:** logical breakdown and understanding of each requirement.
3. **Specification:** representing and storing requirements in a consistent, well-organized form.
4. **Validation:** making sure the requirements define the system that the customer wants.

Requirements Specification: User Stories

- A **User Story** is a requirement expressed from the perspective of an end-user goal.
- User Stories are a popular way of expressing requirements in Agile software development.

User Story Format

"As a `<type of user>`, I want `<some goal>`, So that `<some reason>`."

"As your lab instructor, I want to be able to play your game in a current web browser, so that I don't have to install any additional software."

"As the user of a hotel booking site, I want to be able to search for available rooms within a configurable radius from a certain landmark."

Requirements Specification: User Stories (2)

A User Story…

- Focuses on the viewpoint of a role who will use or be impacted by the solution.
- Defines the requirement in language that has meaning for that role.
- Helps to clarify the true reason for the requirement.
- Helps to define high level requirements without necessarily going into low level detail.

1st Exercise – Lab 1 – Week 23
Deliverable: The Software Requirements Specification

See the exercise sheet in Moodle or the handout during the lab session.

Further

- Explore the Scratch video tutorials:
  https://scratch.mit.edu/help/videos/
- and project ideas:
  https://scratch.mit.edu/ideas