# *Heslington Hustle* Assessment 1 Presentation

## ENG1 Group 23 (Cohort 3)

Oliver Dixon George Cranton Praj Dethekar

Denys Sova Shivan Ramharry Albara Shoukri

Rafael Duarte

Department of Computer Science, University of York

Semester 2, 2024

# Extensible Architecture

- The software architecture has been designed specifically for the Assessment 2 requirements.

# Extensible Architecture

- The software architecture has been designed specifically for the Assessment 2 requirements.
- *Functionally minimal; architecturally maximal.*

# Extensible Architecture

- The software architecture has been designed specifically for the Assessment 2 requirements.
- *Functionally minimal; architecturally maximal.*
- Case-in-point: the MVC architecture employed for the management of player and world metrics. Adding metrics and modifying their behaviour and visualisation schema requires minimal changes to the constructors.

# Extensible Architecture

- The software architecture has been designed specifically for the Assessment 2 requirements.
- *Functionally minimal; architecturally maximal.*
- Case-in-point: the MVC architecture employed for the management of player and world metrics. Adding metrics and modifying their behaviour and visualisation schema requires minimal changes to the constructors.

*Live demonstration: Adding an energy metric (as per Assessment 2)*

# Simple and Modular Area Definitions

- Areas are composed of a tilemap background, the player-controlled character, and various interactable objects.

# Simple and Modular Area Definitions

- Areas are composed of a tilemap background, the player-controlled character, and various interactable objects.
- As such, adding, removing, or modifying on-screen entities is trivial and supported by an `AreaFactory`.

## Simple and Modular Area Definitions

- Areas are composed of a tilemap background, the player-controlled character, and various interactable objects.
- As such, adding, removing, or modifying on-screen entities is trivial and supported by an `AreaFactory`.
- Interactable actions are described by lambda functions, hence any interactable object can execute arbitrary code upon player interaction.

# Simple and Modular Area Definitions

- Areas are composed of a tilemap background, the player-controlled character, and various interactable objects.
- As such, adding, removing, or modifying on-screen entities is trivial and supported by an `AreaFactory`.
- Interactable actions are described by lambda functions, hence any interactable object can execute arbitrary code upon player interaction.

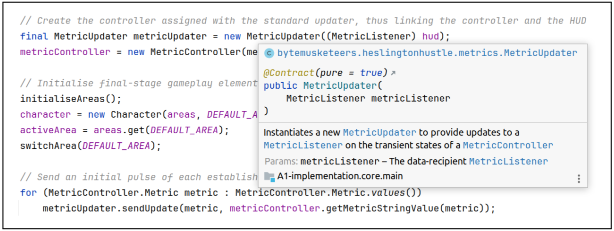
*Live demonstration: Adding an interactable*

# Comprehensive and Consistent Documentation

- Every package, class, interface, method, attribute, and constant has been annotated with consistent and accurate JavaDoc documentation.

# Comprehensive and Consistent Documentation

- Every package, class, interface, method, attribute, and constant has been annotated with consistent and accurate JavaDoc documentation.
- This can be accessed during development with any IDE, or browsed interactively on-line: https://www-users.york.ac.uk/~od641/ENG1-website/javadoc/index.html



```
// Create the controller assigned with the standard updater, thus linking the controller and the HUD
final MetricUpdater metricUpdater = new MetricUpdater((MetricListener) hud);
metricController = new MetricController(me

// Initialise final-stage gameplay element
initialiseAreas();
character = new Character(areas, DEFAULT_A
activeArea = areas.get(DEFAULT_AREA);
switchArea(DEFAULT_AREA);

// Send an initial pulse of each establish
for (MetricController.Metric metric : MetricController.Metric.values())
    metricUpdater.sendUpdate(metric, metricController.getMetricStringValue(metric));
```

bytemusketeers.heslingtonhustle.metrics.MetricUpdater

```
@Contract(pure = true)
public MetricUpdater(
    MetricListener metricListener
)
```

Instantiates a new MetricUpdater to provide updates to a MetricListener on the transient states of a MetricController

Params: metricListener – The data-recipient MetricListener

A1-implementation.core.main

# Any Questions?

Byte
*Musketeers*

ENG1 Group 23 (Cohort 3)

https://www-users.york.ac.uk/~od641/ENG1-website