**Project Documentation**

**CS308 Group PR\_W3**

McAuley Paul, Crawford Robin, Jack James,

Lejhanec Filip, Prohorovs Glebs

**First Meeting**

Wednesday 17th January 3.30pm

During the first meeting, the following points were discussed:

1. **Project specifications** – the playing area and all the elements on it should be scaled to the variable L. The playing area should be a square with the side 20L. Gizmos have their own specific size expressed in terms of L.
2. All the gizmos should extend the abstract class Gizmo.
3. **Triggers**. Triggers have many-to-many relationship with gizmos. Gizmo objects can be passed other gizmo references to build triggers.
4. **Flippers**. Flipper transitions should consider the move time to move synchronously with the ball. Angular velocity of the flipper movement will add energy to the ball. The amount of energy depends on a point at which the ball hits the moving flipper. Hence, certain momentum will be added to the ball. Velocity and reflection angle will be affected. Flippers are constructed from the two straight lines of the same length and two circles. There should be two types of flippers that should take 4 squares with L sides.
5. Zero-radius circles should be added to the end of the lines to imitate realistic collisions.
6. **Version control** – it was said that Git repository would be provided for the project.

**Lab work and task allocation for next week**

GitLab repository was set up to share the code and files. It was decided to use JavaFX for GUI instead of Swing.

**TODO**

Paul – write use cases,

Robin – write project specifications,

James – write trigger control description,

Filip – code initial GUI design and take screenshots;

Glebs – write physics loop description.

The group arranged to meet every Monday outside of the allocated lab slot to check the overall progress. The group has met for the first time on Monday 22nd January. All the related files and initial code were pushed to the gitlab repository.

**Second Meeting**

Wednesday 24th January 3.30pm

Use cases were demonstrated to the TA and the following points were discussed:

1. **Build mode**. A better design would be introducing different modes such as ‘add mode’, ‘edit mode’, ‘delete mode’, ‘rotate mode’ etc. as opposed to clicking on a corresponding button every time a specific action needs to be performed. Both design choices are acceptable.
2. **Saving**. Use toString() to write gizmo instances to a file (type, alias, location coordinates). Saving requires parsing the board twice. On the first pass, write all the gizmos. On the second pass, write all the trigger-action pairs (connects) and other actions. The file reader shouldn’t care about the order of command statements, but objects should come before any actions/connections because all gizmos need to be there before relating them to each other. Introduce checks to make loading and reading from file safe.
3. **Loading**. Possible approach: create gizmos in a factory class (check factory design pattern) when parsing gizmo lines, and push all actions on the stack/queue data structure when parsing actions/triggers lines.
4. **Trigger implementation**. A gizmo object has references to other gizmo objects stored in a collection. Each gizmo instance has its implementation of trigger method. Observer pattern was an acceptable solution.
5. **Friction equation**. The constants represent surface friction and air resistance. The former increases linearly with velocity, while the latter increases exponentially until the terminal velocity is reached. I.e. the friction increases exponentially with velocity.
6. **Flippers**. Each flipper instance should have a key attached to it.

**Lab work**

Class relationships were discussed and an initial class diagram produced. A Gantt chart for the workflow of the following weeks was created. Tasks were allocated to each group member for the next week.

**TODO**

Paul –

Robin – Gantt chart

James –

Filip – front-end development

Glebs – complete physics loop description, create model