The Learning Triangle

Use Case Specification: update view

Version <1.0>

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 31.10.2016 | 1.0 | First set up | LearningTriangleTeam |
|  |  |  |  |
|  |  |  |  |

# 

# **1. Update View**

## **1.1 Brief Description**

The view is important for a user to follow the things happening in our game. It should be able to react on the things that happen in our game. This Use Case describes the different events and how the view react on these events.

# **2. Flow of Events**

## **2.1 Basic Flow**

Activity diagram:

# 

Mockup:

To see an overworld mock-up, just look at the TLT\_UC\_SET\_GAME\_RULES.pdf. The view shows the movement of the triangles, the energy value and you can also see when a triangle dies.

Feature File:

**Feature:** Update View

In order to update the visible gameworld

As a view

I want to react on events which influences the UI

**Scenario:** Randomly Created Overworld

*Given* I started the game

*When* I finished creating a random overworld

*Then* I draw this overworld on my UI

**Scenario:** Create Triangles

*Given* I created the overworld map

*When* I created my triangles

*Then* I display the triangles on my UI

**Scenario:** One Triangle Moves

*Given* I display the triangles

*When* one triangle moves

*Then* I redraw the triangle on his new position

**Scenario:** One Triangle loses or consume energy

*Given* I display the triangles

*When* one triangles energy changes through a field event

*Then* I redraw the energy value of this triangle

**Scenario:** One triangle dies

*Given* I display the triangles

*When* one triangle dies

*Then* I delete this driangle from my UI

# 

# **3. Special Requirements**

# n/a

# **4. Preconditions**

The game started.

# **5. Postconditions**

## Until the game ended, the overworld changes the whole time. After the game ended, the overview must not be updated anymore.

# **6. Extension Points**

## **n/a**