|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Extraction Form** | | | | | | | | | | | | | | | | | | | | | | | |
| **Title** | Detecting Video Game-Specific Bad Smells in Unity Projects | | | | | | | | | | **Authors(s)** | | Antonio Borrelli, Vittoria Nardone, Giuseppe A. Di Lucca | | | | | | | | | | |
| **Year** | 2020 | | | | | | | | | | **Venue** | | * **Journal** | | * Conference | | | | | * Other \_\_\_\_\_\_\_\_\_\_ | | | |
| **Quality Assessment criteria** | | | | * **QC1** | | | | * **QC2** | | | | | * **QC3** | | * QC4 | | | | | * **QC5** | | | * **QC6** |
| **Inclusion Criteria** | | | | * IC1 | | | | * IC2 | | | | | * **IC3** | | | * **IC4** | | | | * IC5 | | | |
| **Exclusion Criteria** | | | | * EC1 | | | * EC2 | | * EC3 | | | | * EC4 | * EC5 | | | | | * EC6 | | * EC7 | | |
| **Approach Used**   * Supervised Machine Learning algorithms * Unsupervised Machine Learning algorithms * Natural language processing * Deep Learning algorithms (Deep enforcement learning DRL’s) * Data mining based techniques * Statistical Method * **Other(Static method unity linter tool)** | | | | | | | | | | **Type of Solution** | | | | | | | | **Yes** | | **No** | | **Unclear** | |
| Novel Technique (Method, Tool, Technique) | | | | | | | | Check mark, Wingdings font, character code 252 decimal. | |  | |  | |
| Evaluation of existing techniques  (Evaluation framework, tool, platform) | | | | | | | |  | |  | | Check mark, Wingdings font, character code 252 decimal. | |
| Supporting techniques | | | | | | | |  | | Check mark, Wingdings font, character code 252 decimal. | |  | |
| **Review dataset** | | **Total number of apps** | | | | 100 open source unity projects | | | | **Evaluation Method Used** | | | | | | | | Feedback methods | | | | | |
| **Total number of crawled reviews** | | | | N/A | | | |
| **Year** | | | | 2020 | | | |
| **Research Type Facet**   * Validation Research * **Evaluation Research** * Solution Proposal * Philosophical Papers * **Opinion Papers** * Experience Papers | | | | | **Solution Type**   * **Single** * Hybrid/Integrated | | | | | | | **Contribution**   * Technique * **Tool (Unity Linter)** * Comparison * Model * Framework * Prototype * Taxonomy | | | | | **Evaluation Strategy**   * **Survey** | | | | | | |
| **Features used**   * Categorical * **Textual** * Both | | | | | | |
| **Factors Considered** | | | Unity project | | | | | | | | | | | | | | | | | | | | | |
| **Notes** | | |  | | | | | | | | | | | | | | | | | | | | | |
| **Limitations** | | | * No classification and categorization of bugs * Limited number of code smell * Static analysis is not particularly used to finding bugs in games | | | | | | | | | | | | | | | | | | | | | |
| **Description / Summary** | | | Unity Linter, a static analysis tool that supports Unity video game developers to detect seven types of bad smells they have identified as relevant in video game development. Such smell types pertain to performance, maintainability and incorrect behavior problems. Unity Linter is, in general, accurate enough in detecting smells (86%-100% precision and 50%-100% recall), and their study shows that the studied smell types occur in 39%-97% of the analyzed projects. | | | | | | | | | | | | | | | | | | | | | |