

Software Design Phase

Problem Description

You are required to design and implement an application model for a platformer video game. The video game consists of a character on the screen, can move left, right and has the ability to jump. The character will eventually interact with the world, having a floor to stand on, obstacles to interact with (bounce, hit, ..) and maintain a score.

You will need to ensure that your application is configurable by a JSON text file, allowing configuration of the main character's size, background cloud velocity. You will be given an example JSON file format that you must follow.

Prior to writing any code, you must produce a UML class diagram, visualising the design of your application and providing a rationale to your design decisions.

Implementation Details

You will use the java programming language to implement your UML model and bind it to an existing view module. You may use the documentation supplied with view module to help with binding.

To simplify the construction of the model, we are providing a simple implementation of the view. This view requires your model to implement interfaces and classes so it is able to be rendered.

- Configuration file in a JSON file format
- Background objects (clouds) must move at a constant speed, your character must be able to move around the stage.
- The character can be created in four sizes which is defined within your configuration file. The size of the stickman as a string. The options are: tiny, normal, large, and giant.
- Stickman's starting position as an x coordinate.
- You will select a pre-existing coding style to use, and provide a reference to it in the accompanying README.txt or README.md.

Refactoring

Any changes to your design after your diagram is submitted must be noted in your final submission. You will need to rationalise changes to your initial class diagram submission.

Template

You are able to access the assignment template from Ed and import into your IDE. You are encourage to maintain a backup copy using a git repository which will maintain hold snapshots of your applications with your commits.

Algorithms and Functions

2D AABB (Axis-Aligned Bounding Box)

A simple collision/box intersection detection function allows your for program to detect when two rectangular shapes have intersected. This is a general algorithm that checks for an overlap between two rectangles, returning `true` if an overlap has occurred.

```
aabbintersect(box1, box2):  
    return (box1.x < (box2.x + box2.width)) and  
    ((box1.x + box1.width) > box2.x) and  
    (box1.y < (box2.y + box2.height)) and  
    ((box1.y + box1.height) > box2.y)
```

Submission Details

You are required submit all assessment items by the due date listed on Canvas.

Your code and tests must be submitted using Canvas. You can upload your files in the assessment page of the appropriate assessment. You are encouraged to submit multiple time, but only your last submission will be marked.

Your report must be submitted onto Canvas using Turnitin, you are allowed a maximum of 1000 words which must concisely provide a rationale to your design.

Marking Criteria (10%)

- 2 Marks for a UML class diagram of your model submitted prior to the final implementation.
- You are required to submit a UML class diagram.
- 4 Marks for the final UML class diagram of your model, providing details about any changes from your first submission.
 - You must submit your final UML class diagram with a report, rationalising your design decisions.
- Reflect on any changes you have made from your initial design.
- 4 Marks for the implementation of the model, once you have designed your model, implement it using the Java programming language. You will be assessed on logical categorisation of your model,
 - Ensure your program can compile on the university's lab machines.
 - Ensure your program implements the view's interfaces and can render some simple elements - this will be validated by a unit test suite we will run it against.
 - Ensure your program implements the logic required to move the character.
- Ensure your program is able to load a configuration file and set game specific environment variables.

You are currently not being assessed on any design patterns at this stage of development.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire

assignment. Negative marks can be assigned if you do not properly follow the assignment specification, or your code is unnecessarily or deliberately obfuscated.

Academic Declaration

By submitting this assignment you declare the following:

I declare that I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgement from other sources, including published works, the Internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Computer Science, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program, and that a copy of the assignment may be maintained by the service or the School of Computer Science for the purpose of future plagiarism checking.