CSE 231 Problem Set 04

# Problem 32.1: Alignment of Missile Command

Consider the following problem domain:

The 1980 game Missile Command consists of three types of entities: targets, assets, and inert objects. The targets are things that the player can shoot at, the assets are things the player attempts to protect, and inert objects are non-interactive elements designed to contribute to the look and feel of the game. Among the targets are incoming ICBM missiles, smart-bombs, bomber planes, and satellites. Each can be destroyed by counter-missiles controlled by the player. Assets include the anti-missile batteries from which the counter-missiles are fired and cities. All assets can be destroyed by missiles and smart bombs. The inert objects include the score, number of missiles left, high score, and other status messages.

Identify the level of alignment from the following class diagram:



Rationale for whether this design is **Extraneous**:

This could be considered Extraneous as the class of Statistics should not inherit from static, but instead be placed under its own inheritance.

Rationale for whether this design is **Partial**:

This could be considered partial because we are missing a construct for Battery and City. There should be a construct for ASSESTS that can be interacted with but are not mobile, btu are also still important to the look and play of the game.

Level of alignment:

The level of alignment for this problem is considered PARTIAL.

If the class diagram does not exhibit complete alignment, provide a class diagram that does:

# 



# Problem 32.2: Alignment of Pac-Man

Consider the following problem domain:

The 1980 game Pac-Man consists of a maze and several game entities: ghosts, pellets, and Pac-Man himself. The maze is a matrix of sides, each of which can be opened or closed. The ghosts are four flavors (Blinky, Pinky, Inky, and Clyde). These ghosts are in one of three modes: edible (blue), slow (when a power pellet is eaten), normal, and dead (when the ghost returns to the base). The pellets have three flavors: normal, energizers, and fruit.

Identify the level of alignment from the following class diagram:



Rationale for whether this design is **Extraneous**:

This could be considered Extraneous as the relation for the states of the Ghosts is missing from the class diagram. The ghost require the states of Edible, slow, normal, and dead, none of which are represented.

Rationale for whether this design is **Partial**:

This could be considered Partial as there is a relation that is needed between the Ghosts and the Pellets

Level of alignment:

The alignment would be considered PARTIAL, There is a large amount of inheritance missing from the tree mainly related to the Ghost and pellet relation, as sum functionality for the Ghosts are reliant on information from the pellets

If the class diagram does not exhibit complete alignment, provide a class diagram that does:

mountiHome (245 Davison Dr)



# Problem 32.3: Design Dig Dug

Consider the following problem domain:

The 1982 game Dig Dug consists of enemies, Dig Dug, and a playing field. There are two types of enemies: Pookas and Fygars. Both can move through tunnels, can ghost through solid ground, or can be in various stages of inflation. The ground can be dirt, a tunnel, or a rock.

Create a class diagram exhibiting complete alignment.

Justify why you think your design’s alignment is complete.

This diagram is complete as it each enemy type inherits from a base enemy, each type of surface that they ground can represent is shown, and the and the enemies that require movement are also shown.



Enemy

Pookas

Fygars



movement

Ground

Dirt

Tunnel

Rock

# Problem 32.4: Design Centipede

Consider the 1981 arcade game Centipede. Research how the game was played:

Provide a class diagram exhibiting complete alignment:

Justify why you think your design’s alignment is complete:

This class diagram references all entities that are contained within the game, such as darts, entities, etc. This diagram also references static information such as score, time as well as static but interactive objects like mushrooms and launchers.



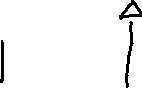
Launcher



Statistics

Static

Game



Mushrooms

Assets

Scorpions

Movement

Environment

Dart

Spider

Entities

centipede