TEAM - 09

Software Architecture and Project Scheduling

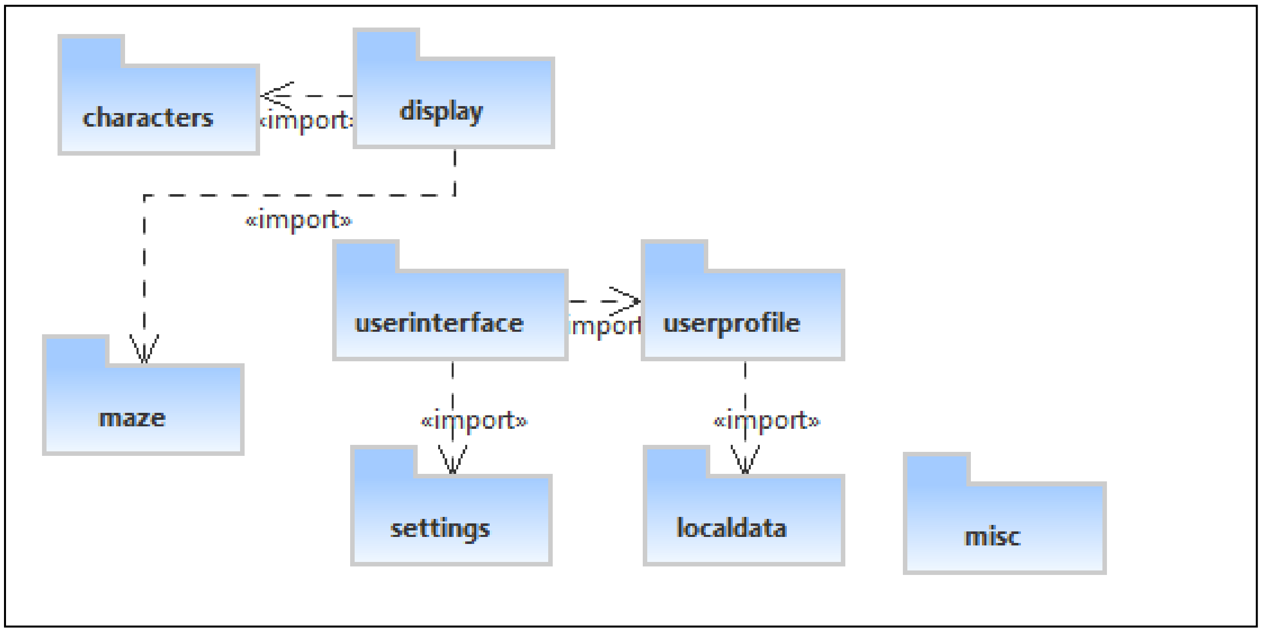
ECSE 321: Team Project

10/29/2012

Part 1: Meeting with TA

Part 2: Decomposition of System into Packages

System UML Package Diagram:



Package Description and Class Composition:

Package userinterface	**Description**: The user interface package handles the graphical display of the various menu windows of the application.
SettingsMenu	This class defines the layout of a settings display.
MainMenu	This class defines the layout of the main menu, including a login field.
UserProfile	This class defines the layout of the user profile display.
StatisticsMenu	This class defines the layout of the statistics display.
SaveLoadMenu	This class defines the layout of the save/load dialog.

Package userprofiles	**Description:** Player profiles, profile retrieval, profile storage, information/statistics retrieval
Class Profile	Constructed from () if guest, or (username, password) if signed in; hashes password and matches against storage.
Interface HashFunction	Used by Profile to hash passwords
PBKDF2 implements HashFunction	Implements hash(); uses a third party library (to be defined later)

Class ProfileStatistics	Instantiated by Profile; models data about past and current games.
Class GameStatistics	Models data about a single game

Package settings	**Description:** Package granting the ability to alternate between different video resolutions and sound styles for the game
Class Sounds	Possibility to alternate between three sound options (none, low, high)
Class Resolution	Possibility to alternate between two resolution options (300x400, 1024x768)

Package localdata	**Description:** Package responsible for storing information related to game settings, profile statistics etc.. locally on the computer
XML file Users	A XML file with usernames, hash addresses of passwords and statistics

Package display	**Description:** Package responsible for displaying instances of maze
Class Wall	Display information on the walls in the maze
Class Navigable	Display information for the navigable surface
Class CageDoor	Display information for the door of the cage containing the ghosts.
Class Fruit	Display information for the fruits
Class Powerpill	Display information for the power pills
Class MatrixMap	Map of the whole maze, in the form of a matrix
Class TextDisplay	Display statistics regarding number of fruits, life and score
Class PrintLife	Print the number of lives on the game board
Class PrintFruit	Print the fruits acquired on the game board
Class PrintMap	Print the whole maze, based on MatrixMap
Class PrintScore	Print the score above the maze (on the game board)
Class Board exteds JFrame	Displays a game board

Package characters	**Description:** Responsible for the storing of all character information and behaviour (such as pursuit algorithms)
Abstract class Character	Generic character, basic information as to how characters move around the layout
Class PacMan extends Character	In-game Pac-Man character, including display information, code for collision with items and ghosts
Abstract class Ghost extends Character	Generic ghost, includes rendering information and GhostPatterns as a class variable
Class Blinky extends Ghost	Red ghost main class, includes display methods
Class Inky extends Ghost	Blue ghost main class, includes display methods
Class Pinky extends Ghost	Pink ghost main class, includes display methods
Class Clyde extends Ghost	Orange ghost main class, includes display methods

Abstract class GhostPatterns	Methods for controlling ghost orientation and movement
Class BlinkyPatterns extends GhostPatterns	Red ghost behaviour when not frightened (includes scatter, pursuit and Cruise Elroy late game acceleration)
Class InkyPatterns extends GhostPatterns	Blue ghost behaviour when not frightened (includes scatter and pursuit)
Class PinkyPatterns extends GhostPatterns	Pink ghost behaviour when not frightened (includes scatter and pursuit)
Class ClydePatterns extends GhostPatterns	Orange ghost behaviour when not frightened (includes scatter and pursuit)
Class FrightenedPattern extends GhostPatterns	Behaviour for any frightened ghost
Class DeadPattern	Behaviour for any dead ghost
Class CagedPattern	Behaviour for any ghost that has yet to leave the cage at game start
Class GhostEyes	Display information and behaviour for ghost eyes (when part of a living ghost)

Package maze	**Description:** The Maze package is responsible for all of the classes that describe the maze tiles and their contents.
Tile	Generic describing a single maze tile
Emtpy	An empty tile describes a tile that doesn't contain any objects.
Wall	A wall describes a tile that cannot be traversed by any character nor contain any objects.
Edibles	An edible is a generic class describing the collectable objects that can be contained on a tile.
Dot	A dot is an edible that when eaten will increase the player score.
Energizer	An energizer is an edible that when eaten will change the state and behavior of the ghost characters.
Level	This class describes a game level difficulty, as well as the maze layout.

Package misc	Description: Package responsible for other miscellaneous classes and methods that could not be classified into any specific category
N/A Thus Far	

Software Architecture Explanation:

The user interface is the connection between the user and the game. The user interface package deals with creating a main menu which allows the user to "navigate" through and select any of the different functions of the game such as loading/saving a game, logging in, changing settings etc… The settings package defines main settings such as screen resolution and color, based on the system which the game is running on. It also allows the user to change certain specific settings such as sound volume among others. Therefore the settings of the game are system specific and are unrelated to the specific user logged on at any point in time. This is why settings does not need to access local data. This is in contrast with user profile, which deals with the creation and modification of accounts which will be linked to statistics and scores of users (or players). Only the user profile needs access to the local storage because any and all information that needs to be stored, retrieved or modified, must be linked with a profile or username.

The display package is a package that the user interface will utilize once a game is begun. Depending on the level, the display will import instances of characters and mazes and use the methods implemented within itself to display the graphics of the Pacman game. Moreover the display package will create a game board once a level has begun. The characters package will be responsible for defining the 4 ghosts, their behaviour throughout various stages of the game and of course pacman himself and the control of this character by the player. Maze on the other hand will be responsible for defining the stage in which the characters will "exist". The maze is responsible for creating the full level and will define the format of the whole maze, along with each individual tile and what it is populated with (empty, dot, energizer, wall etc…).

Each of these main packages will interact with each other in order to produce a fully functional Pacman game. Finally, there is a misc package, which could include any unclassifiable or optional classes. Although none have been found so far, this may change throughout the course of the project.

Part 3: Project Scheduling

List of Activities:

Tasks in user interface:

- Write the Window class: 4 man-hours
- Write the Button class: 2 man-hours (requires Window)
- Write the Slider class: 2 man-hours (requires Window)
- Write the Textbox class: 2 man-hours (requires Window)
- Write the Label class: 2 man-hours (requires Window)

Tasks in players (userprofiles):

- Locate, test and install a hashing library: 2 man-hours
- Build Hash matching code, including unit tests: 2 man-hours (doesn't depend on hashing library as we can use a dummy x -> x hash function)
- Write log-in front-end: 4 man-hours (depends on user interface)

Tasks in settings:

- Write the sounds adjustment class: 2 man-hours, depends on the abstract class responsible for the sounds

- Write the class to alternate resolution: 4 man-hours, depends on the display code in the userinterface package

Tasks in localdata

- Write the XML file and find the best way to store the data : 4 man-hours, depends on the log-in function and the hashing library

Tasks in mazedisplay:

- Write the code for the maze: 10 man-hours, depends on the matrix class of the maze
- Write the matrix class of the maze: 5 man-hours
- Write the display code for the edibles: 4 man-hours, depends on the maze display code

Tasks in characters:

- Write ghost patterns: 10 man-hours, depends on maze layout and cage implementation
- Write ghost display code: 4 man-hours, depends on maze layout
- Write Pac-Man control code, depends on maze layout
- Write Pac-Man display code, depends on maze layout
- Test ghost behaviour: 1 man-hour, depends on ghost patterns and display code
- Write GhostEyes: 2 man-hours depends on ghost display code, ghost patterns

Tasks in maze package:

- Write the Tile superclass: 6 man-hours
- Write the Empty class: 2 man-hours(requires Tile)
- Write the Wall class: 2 man-hours (requires Tile)
- Write the Edibles superclass: 4 man-hours(requires Tile)
- Write the Dot class: 2 man-hour(requires Edibles)
- Write the Energizer class: 4 man-hours (requires Edibles)
- Writing the Level class: 4 man-hours (requires Tile, Edibles, and characters package)

Writing out the input package/controller: 6 man-hours

Tasks in misc:

- Thus far, no miscellaneous classes have been identified but to be on the safe side it would be clever to leave 2-3 man-hours for this package

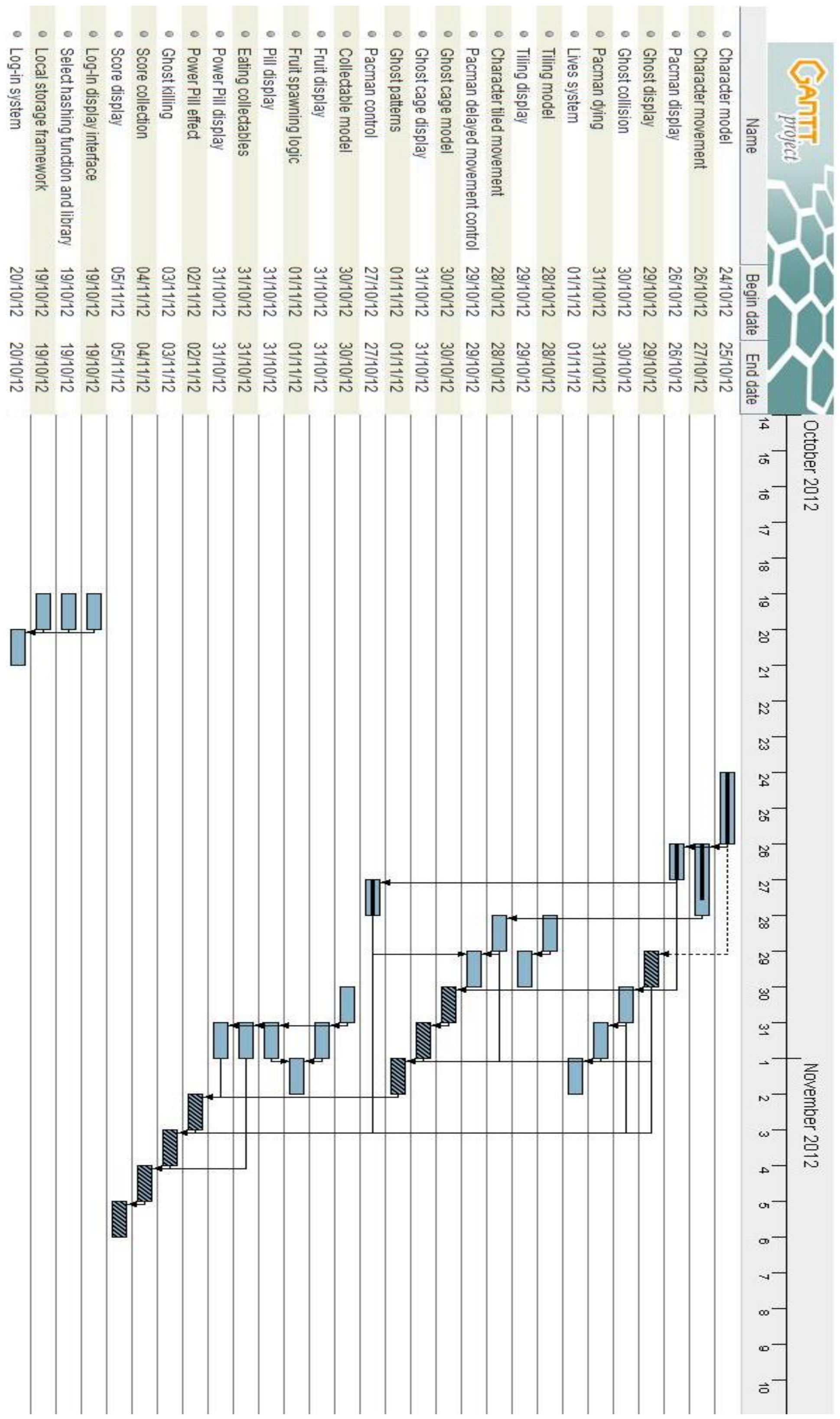
Activity Division and Estimated Time:

Activities were evenly among team members according to the following table: (all tasks related to the following subsystems must be completed fully by assigned team member(s))

Team Member	Subsystem
Antoine	User Input, User Interface, Game Management, Select Level
Carl	Ghost Algorithms, Login/Logout (player management)
Matthew	Ghost Algorithms, Statistics Collection/Calculation/Display,
Maxime	Game Display, Data Storage, Settings

NOTE: Gantt Chart on following page. Critical path identified by the diagonally shaded tasks.

Gantt Chart:



Name	Begin date	End date
Character model	24/10/12	25/10/12
Character movement	26/10/12	27/10/12
Pacman display	26/10/12	26/10/12
Ghost display	29/10/12	29/10/12
Ghost collision	30/10/12	30/10/12
Pacman dying	31/10/12	31/10/12
Lives system	01/11/12	01/11/12
Tiling model	28/10/12	28/10/12
Tiling display	29/10/12	29/10/12
Character tiled movement	28/10/12	28/10/12
Pacman delayed movement control	29/10/12	29/10/12
Ghost cage model	30/10/12	30/10/12
Ghost cage display	31/10/12	31/10/12
Ghost patterns	01/11/12	01/11/12
Pacman control	27/10/12	27/10/12
Collectable model	30/10/12	30/10/12
Fruit display	31/10/12	31/10/12
Fruit spawning logic	01/11/12	01/11/12
Pill display	31/10/12	31/10/12
Eating collectables	31/10/12	31/10/12
Power Pill display	31/10/12	31/10/12
Power Pill effect	02/11/12	02/11/12
Ghost killing	03/11/12	03/11/12
Score collection	04/11/12	04/11/12
Score display	05/11/12	05/11/12
Log-in display interface	19/10/12	19/10/12
Select hashing function and library	19/10/12	19/10/12
Local storage framework	19/10/12	19/10/12
Log-in system	20/10/12	20/10/12