



# ASSESSMENT ITEM COVER SHEET

Student Name: YAT CHUN LEUNG  
Last name (USE BLOCK LETTERS) First Name

Student Number: 

3	0	9	5	0	3	2
---	---	---	---	---	---	---

 Email: c3095032@uon.edu.au  
(Your studentmail account)

Course: 

I	N	F	T	6	3	0	3
---	---	---	---	---	---	---	---

Computer Games Production  
Course Code Course Title

Campus of Study: GradSchool

Assessment Item Title: Game Prototype 1 – Puzzle Game

Tutorial Group:  Word Count (If applicable):  Due Date/Time: Oct 18, 9.00am

Lecturer/Tutor Name: \_\_\_\_\_

Extension Granted:  Yes  No Granted Until: \_\_\_\_\_  
Please attach the approved copy of your extension approval.

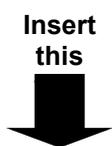
Include here any instructions / checklist for submission

I declare that this assessment item is my own work unless otherwise acknowledged and is in accordance with the University's plagiarism policy available from the Policy Library on the web at <http://www.newcastle.edu.au/policylibrary/000608.html>  
I certify that this assessment item has not been submitted previously for academic credit in this or any other course. I certify that I have not given a copy or have shown a copy of this assessment item to another student enrolled in the course.

I acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:

- Reproduce this assessment item and provide a copy to another member of the Faculty; and/or
- Communicate a copy of this assessment item to a plagiarism checking service (which may then retain a copy of the item on its database for the purpose of future plagiarism checking).
- Submit the assessment item to other forms of plagiarism checking.

I certify that any electronic version of this assignment item that I have submitted or will submit is identical to this paper version.



Turnitin ID: (if applicable)

Signature: Joseph Leung Date: Oct 7, 2010

## Ambassador from the wild (AOTW)



The prototype in assignment 2 is actually part of the working game, which Dr.Mad goes mad and throw stuff (represents by cards) off the window, as well as assemble the ultimate weapon. This part included both Matching cards and Jigsaw puzzle, it's fully working except it doesn't include multi-player support.

### Features

- Throw to window animation
- Disaster
- Popup notification box
- Puzzle undrag animation

### Contribution in this prototype

Most code is based on my work, except I may use the logic learnt from the textbook (*Gary Rosenzeig, 2007*) as well as some of my previous experience. Most resources like Cards/Puzzle images and Sound effects are based on the textbook. (*Gary Rosenzeig, 2007*).

Please refer to [Appendix 1](#) for a list of contribution:

### Source control

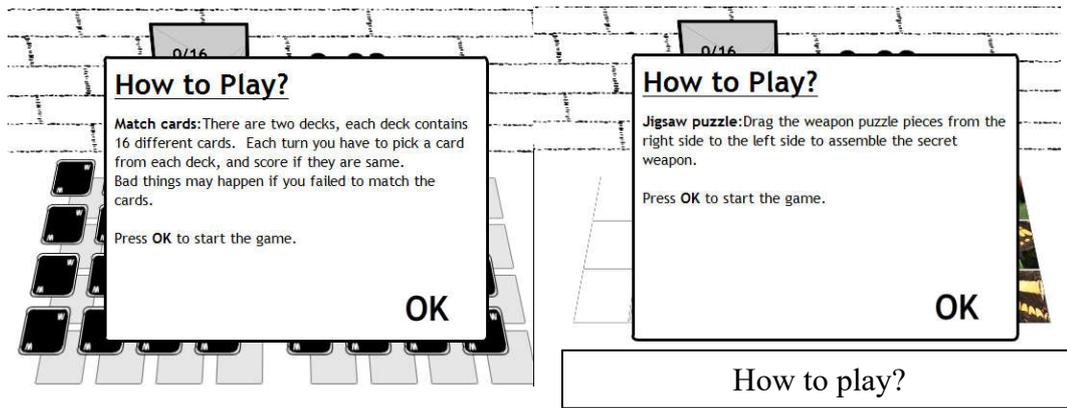
The project uses Mercurial Distributed SCM for source control to monitor changes, the source control files is located in .hg directory.

### JavaScript

The project uses JavaScript to change html background and to enable cheat mode, if you run the html file locally please add the flash file to trusted locations, as described here:

[http://www.macromedia.com/support/documentation/en/flashplayer/help/settings\\_manager04a.html](http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manager04a.html)

## How to play?

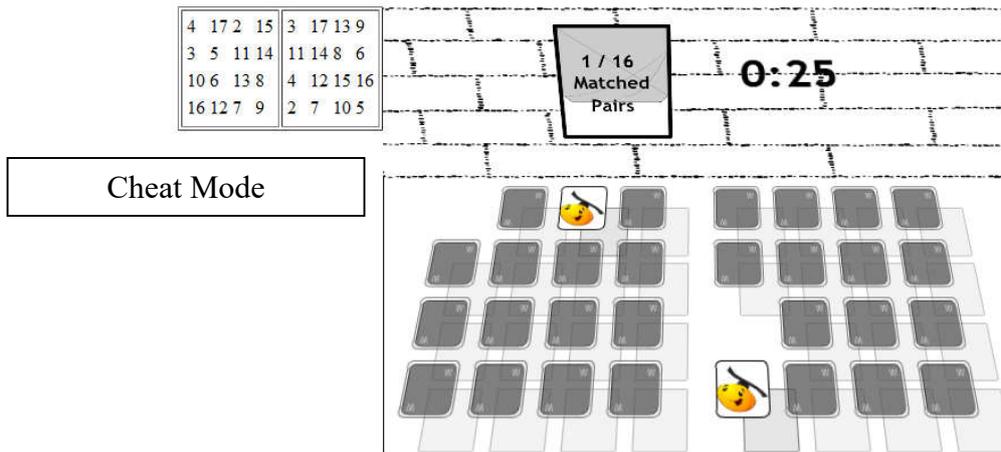


Please run the [CardGame.html](#) in the output directory.

The prototype can be started by selecting [Multi Player \(One Vs One\) \ Server \(Puzzle\)](#). Selecting other options will display a Notification screen mentioned that they are under construction.

To win the game, you have to play Matching cards, then Jigsaw puzzle. Both games included how to play screen when started.

Cheat mode is available in both games; it shows the game data as html table, press "Cheat" button to enable this mode.



## References

Gary Rosenzweig. (2007). *ActionScript 3.0 Game Programming University*: QUE Publishing.

Bryan O'Sullivan (2009). *Mercurial: The Definitive Guide*: O'Reilly Media

Elizabeth Castro (2003). *HTML for World Wide Web visual quickstart guide: with XHTML and CSS, 5th ed.* : Pearson Education Inc.

Shelley Powers (2008). *Learning JavaScript* : O'Reilly Media

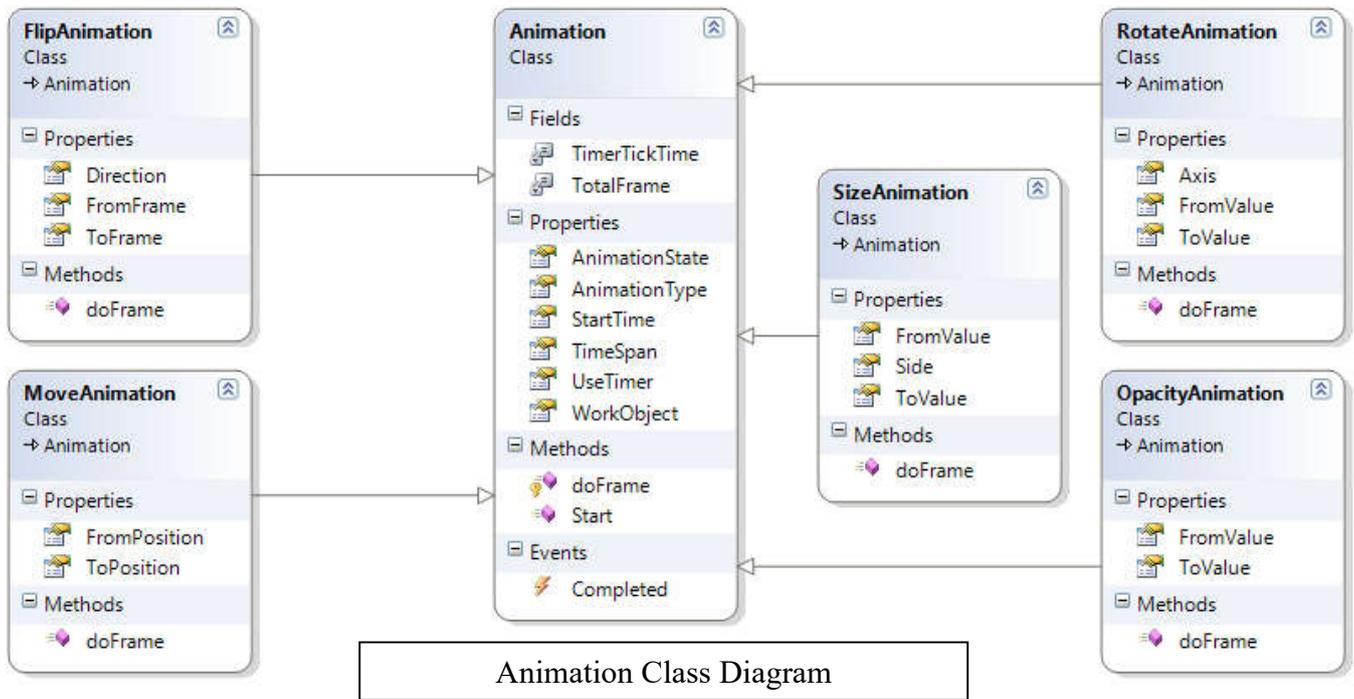
*Coding data exchange between JavaScript and Flash Player.* (2010).

Retrieved October 9, 2010, from

[http://kb2.adobe.com/cps/156/tn\\_15683.html](http://kb2.adobe.com/cps/156/tn_15683.html)



## Animations



The main class is *Animation*, which does all the control, it support two modes.

If *UseTimer* is *false*, it will render the animation based on *TotalFrame* (which is 10). If *true*, it will recognize the *TimeSpan* and *StartTime* property, and do smoother animation.

Either way is based on handling *ENTER\_FRAME* event, which then send a *doFrame()* method, the *doFrame()* method takes a parameter valued between 0 to 1, and is implemented by the inherited class.

Using this method, extending *Animation* class becomes very simple, for examples, for *OpacityAnimation*, you only have to implement *doFrame()* method:

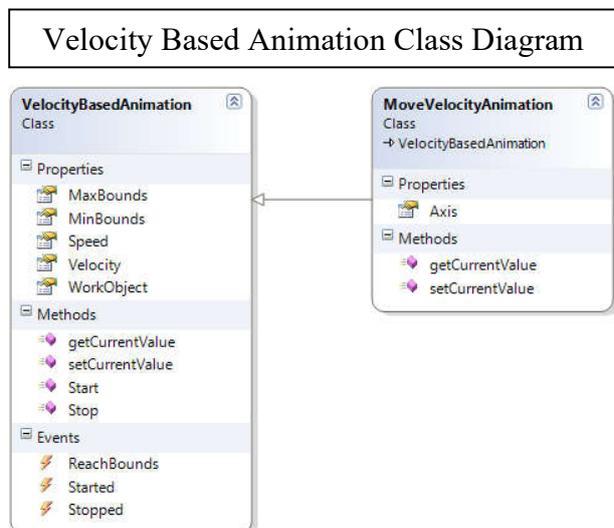
```

protected override function doFrame(frameRatio:Number)
{
    WorkObject.alpha = FromValue + ((ToValue - FromValue) * frameRatio);
}
  
```

## VelocityBasedAnimation

*VelocityBasedAnimation* is mainly designed for next assignment, it's currently control the card movement in the main screen, *MoveVelocityAnimation*, when started, controls the movement of a *MovieClip* in a specific axis for a specified speed.

I have added also *AngleMoveVelocityAnimation*, which move a *MovieClip* based on a specific angle, this class is intended for future use.



# Appendixes

## Appendix 1 : Contribution Chart

Classes / Interfaces / Events	Contribution		Description
	Others	My own	
<b>Animation</b>			
- Animation		<input type="radio"/>	Base class for all Time-Limited animations.
- FlipAnimation		<input type="radio"/>	Flip from one frame to another.
- MoveAnimation		<input type="radio"/>	Move from one location to another.
- OpacityAnimation		<input type="radio"/>	Set Opacity from a value to another.
- RotateAnimation		<input type="radio"/>	Rotate x, y or z axis of a MovieClip.
- SizeAnimation		<input type="radio"/>	Change the size of a MovieClip
- SequentialAnimation		<input type="radio"/>	Run multiple Animation(s) in sequential.
<b>Velocity animation</b>			
- VelocityBasedAnimation		<input type="radio"/>	Base class for all Velocity (Speed) Limited animations.
- MoveVelocityAnimation		<input type="radio"/>	Move a MovieClip for x px per second.
- BoundEvent		<input type="radio"/>	Raised when the movieclip hit the bounds.
<b>Drag and Drop</b>			
- DragDropTools		<input type="radio"/>	Generic class for Drag and Drop support.
- ISupportDrag		<input type="radio"/>	Implemented by controls to support drag.
- ISupportDrop		<input type="radio"/>	Implemented by controls to support drop.
<b>Allocation</b>			
- WrapPanel		<input type="radio"/>	Organize the child from left to right, and move to next line when out of space.
- Grid		<input type="radio"/>	Organize the child based on child's col/row.
- IGridItem		<input type="radio"/>	To be implemented by Grid's children.
<b>Misc</b>			
- NotificationBox		<input type="radio"/>	Display tips in the middle of the screen.
- TimerTextField		<input type="radio"/>	Display time elapsed.
- LoadingControl		<input type="radio"/>	(Unused) Show progress when loading.
- Window11		<input type="radio"/>	Show how many cards matched.
<b>Card related</b>			
- CardDeck		<input type="radio"/>	Holder of one set of cards (16)
- Card11	<input type="radio"/>	<input type="radio"/>	The original art is based on the textbook. The code is animation based and is done by myself.
- CardEvent		<input type="radio"/>	Event used when a card is selected.
<b>Puzzle related</b>			
- Puzzle		<input type="radio"/>	Holder of one set of puzzle (16 PuzzlePiece)
- PuzzlePiece		<input type="radio"/>	A piece of puzzle, draggable.
<b>Main controller</b>			
- MainScreen		<input type="radio"/>	Display Title screen and Introduction screen.
- MatchGame		<input type="radio"/>	Main controller of both puzzle games.
- GameOverScreen		<input type="radio"/>	Display game over and time elapsed.
- GameStrings		<input type="radio"/>	Included most game text for localization.
<b>Resources</b>			

- Wall		<input type="radio"/>	Background of the MatchGame
- Symbol11	<input type="radio"/>		The art is based on the textbook.
- Sound Effects (3)	<input type="radio"/>		The sound effects is based on the textbook.
- BasicButton		<input type="radio"/>	Button
- OKButton		<input type="radio"/>	OK Button
<b>Support files</b>			
- CardGame.js		<input type="radio"/>	Added to support background changing/cheat mode.
- CardGame.html	<input type="radio"/>	<input type="radio"/>	I have modified the file generated by Flash.
- Diagrams	<input type="radio"/>	<input type="radio"/>	Generated by Microsoft (c) VS 2010