# BAA DigiPen Video Game Design 10

**District Name: Kamloops/Thompson**

**District Number: 73**

**Developed by: Keith Thomson and Reg Reimer**

**Modified by: Justin deVries**

**Date Developed: Spring 2010 (modified Fall 2011)**

**School Name: Sa-Hali Secondary**

**Principal's Name: Rick Kienlein**

**Board/Authority Approval Date:**

**Board/Authority Signature:**

**Course Name: BAA Digipen Video Game Design 10**

**Grade Level Course: Grade 10**

**Number of Course Credits: 4**

**Number of Hours of Instruction: 110**

## Prerequisite(s) recommended:

- Interest in Video Game Design
- C+ Final grade in Mathematics 9
- Art skills suggested

## Special Training, Facilities or Equipment Required:

Computer Science background, or experience in similar area along with training provided by DigiPen Institute of Technology. Facilities should include a computer lab with recent model computer, with reasonable video card and memory. Video projector, DigiPen training software (proprietary game development tools), graphics programs for 2D graphics (preferably PhotoShop CS3 or CS4, or Photoshop Elements). Printed support materials (workbooks, etc) are provided by DigiPen ï students retain their workbooks.

## Course Synopsis

This course is one of three courses that make up the DigiPen ProjectFun Workshops curriculum, targeted at high school students that show interest in video game development as a learning vehicle. The program is an introduction to video game design and programming, made up of three levels, each with unique guided games that students will produce. In addition to introducing students to game design, the material also covers topics about the gaming industry, design elements and production art.

## Rationale:

British Columbia is rapidly attracting a concentration of video game production companies. Employment and compensation opportunities provided in this industry are among the fastest growing in Canadaôs knowledge-based economy. The video game industry now vastly outsells the film industry in North America. This program will provide our students with an opportunity to participate in the curriculum developed by an internationally recognized video game university (DigiPen), while gaining an introduction to the field of video game creation. Students will also acquire the knowledge and skills to enable them to become successful candidates for the Digipen Academy course modules, as well as TRU credit for computer science. Students will be encouraged to develop cross-curricular knowledge and skills in disciplines such as Mathematics, Science, Music and Art using video games as the motivation to achieve these objectives.

The course offers an introduction the video game production process, programming, game loop concepts, mouse and keyboard input, animation, object behaviours, sound effects and music.

## Organizational Structure

This course outline follows the development of four guided games (simulations), each of which builds more concepts. The following table lists approximate teaching time for each unit:

| Unit 1 ï Introduction to Game Design | Concepts | Time |
|---|---|---|
| | Industry overview, History, size, value, demographics of video game industry. | 10 hrs |
| | Definition of a game ï examples of good and bad | |
| | Game design principles ï mechanic, dynamic, aesthetic Objectives, balance, meaningful decisions | |
| Unit 2 ï óThe Cageô | | 4 hrs |
| | Introduction to Project Fun Editor Software | |
| | Introduction to Image Editing Software | |
| | Defining user interactivity | |
| | Game loop | |

|  |  |  |
|---|---|---|
|  | Actors and Sprites |  |
|  | Sprite Groups |  |
|  | Maps for Games |  |
|  | Sprite Collision |  |
|  | Expansion ideas: animating sprites, resetting levels, multiple cages, changing ball animation |  |
| Unit 3 ï óPongô | Review Actors, Sprites, Collision Data, Game art<br>Introduction to state machines<br>Text objects, changing text object values<br>Splash screens, levels, win and lose screens | 6 hrs |
| Unit 4 ï óBrixô | Ghost collision | 10 hrs |
|  | Colour keying |  |
|  | Global Data |  |
|  | Object Functions (global) |  |
|  | My Functions (local) |  |
|  | Expansion ideas: other levels, multiple balls, multiple hits, particle engines, using state machines for paddle movements |  |
|  | Adding Sounds |  |
| Unit 5 ï óSide Scrollerô | Master Maps | 20 hrs |
|  | Wrapping Maps |  |
|  | Eight-way directional movement |  |
|  | View port / scrolling bounding box |  |
|  | Main character |  |
|  | Peg registering |  |
|  | Dynamic Sprite creation |  |
|  | Hot Spots |  |
|  | Movement patterns |  |
|  | Expansion ideas: secondary weapons, enemy power ups, bosses, weapon upgrades |  |

|  |  |  |
|---|---|---|
| Unit 6 ï ñPlatformerò | Game engines | 20 hrs |
|  | Horizontal / Vertical flipping |  |
|  | State machines |  |
|  | Artificial Intelligence |  |
|  | Expansion ideas: characters flying, levels, adding sprite to movement patterns, adding main character weapons, hidden doors |  |

| | | |
|---|---|---|
| Integrated Theory | (include during units as applicable) | 10 hrs |
| Pong or near beginning | Product Analysis ï  learn how to look at popular games critically. Evaluate production qualities of student and professional games | |
| Brix | Game Loop Theory ï  understand game loop theory, frame rates, updating vs. rendering | |
| | Planning a game ï  what makes game design fun, risk and reward concept | |
| Side Scroller | Pointers: dangers of improper use, addressing (ćthisô and óThisô) | |
| | Evaluation of good game play | |
| | Careers ï  defining various roles in the game industry | |
| | Technical Design Document ï  how it leads to assembly of skills | |
| | Careers ï  research current compensation and correlation between education and financial returns | |
| Platformer | Research Platform-style jumping games (both 2D and 3D) for comparison of core experience | |
| | Student Designed Platform Game within existing game engine, examine needs and possible solutions of Platform games | |
| | Assessment of personal skills & abilities, listing games that could be constructed using those skills. Compare lists of games theyôd like to play, and inventory skills required to make those games | |
| | Play a variety of games, and try to define the algorithms behind the game play | |
| Unit 7: | Summary Exercise: create your own game | 30 hrs |
| | Students use all acquired knowledge to synthesize their own games, solve their own problems, and foster effective creativity in this product-based unit | |
| | New problems encountered: requires additional student specific learning and use of additional online resources. | |
| TOTAL | | 110 hrs |

## Unit Topic Descriptions

**Unit 1 – Introduction to Game Design**
In this unit students will become familiar with some early instances of computer simulations, as well as the development of the use of the computer for gaming. There is an overview of the business of computer gaming, opportunities for careers in the industry, and discussions about the value of the industry to the local economy of British Columbia as well as the world. Students become aware of some of the market segmentation applied to video game creation, and how the various segments control the types of games being created.
Apart from market and industry concepts, students also begin to delve into the concepts of game design (which applies to both digital and non-digital gaming). With a central core ideology of mechanic, dynamic, and aesthetic (MDA), students better learn how to express their creativity in a design of a board game, complete with comprehensive rules.

Introductory Concepts
*It is expected students will:*

- Become familiar with a variety of game categories
- Be able to identify examples of games from various genre
- Be able to summarize the important business considerations that go into creating a game
- Work collaboratively through the iterative process of game design
- Evaluate the components of a game that are either enjoyable or not, and why
- Discuss, with detail and vocabulary, the essences (MDA) of a game and why or why not they work well

**Unit 2: The Cage**
In this unit students are introduced to their first ósimulationô called óThe Cageô.

Computer Science and Game Concepts
*It is expected students will:*

- Cover topics of real-time interactive simulation and game elements, arriving at a definition of a video game.
- Be introduced to game creation components such as the game engine, maps, actors and sprites.
- Be introduced to the concepts of collision between actors, and how the game engine can be used to detect collisions.
- Optional extensions could include how to animate the ball sprites, change ball spriteôs animation when it collides with another sprite or the level, add a reset button, and add levels and variations to the game.

Production Art:
*It is expected students will:*

- Use graphics software to create the shapes, colors, and textures needed for their game screens.
- Identify game appearances associated with various categories and genres of games

**Unit 3: “Pong”**
In this unit students are introduced to an historic video game called ôPongö.

Computer Science and Game Concepts
*It is expected students will:*

- Use game engine to create actors, sprites, and map collision data
- Explain behaviors attached to individual game components
- Explain a variety of ways to solve a behavior
- Understand how collisions are checked using systems of linear equations

Production Art:
*It is expected students will:*

- Use graphics software to create the shapes, colors, and textures needed for their game screens.

**Unit 4: “Brix”**
In unit 4, a new game is introduced. It is called ñBrixò. This game is similar to *Breakout* in many respects, and is the first true game created in this course. *Brix* features a paddle at the bottom of the screen, blocks at the top, and a ball in between. Once the ball hits a block, the block will disappear. Additionally, when the ball hits the paddle it is reflected back toward the bricks. The object of this game is to move the paddle in such a way that the ball will hit every single block on the screen. Once this completed, the game will either proceed to the next level or the winning screen.

Computer Science and Game Concepts
*It is expected students will:*

- Recognize concepts that make *Brix* a game rather than merely a simulation, and be able to identify ways to make it more challenging to play
- Review concepts such as sprites and actors, collisions, maps and levels from previous unit
- Recognize the discreet logic parts that make the game run, and how this information can be shared (Global Data) between logic parts. Example: a logic part that checks to see if a player collides with a switch to open a door.
- Recognize the need for behavior packets (Object Functions) to govern different aspects of the game, and how to attach them to the appropriate parts of the game.
- Recognize the value of globally available behavior packets (My Functions) for use by any part of the game.
- Optional extension: adding levels, adding multiple balls, adding resiliency to or animating certain bricks, using the particle engine to highlight ball interaction with bricks.

Production Art
*It is expected students will:*

- Continue their development of art design principles: balance, emphasis, contrast, movement, rhythm, pattern, variety and unity.

- Develop game maps, actors and sprite animations with cohesive graphic themes, according to the game category and genre of their choice

**Unit 5: Side Scroller**

The game *Side Scroller* introduces a number of new concepts that are fundamentally different than *The Cage* and *Brix.* It serves as a doorway to basic artificial intelligence in games, uses a scrolling map to give the illusion of a moving world, and finally it allows eight-way-directional movement. It continues to expand on previous project implementations of maps, actors and sprites, collisions, and individual behaviors.

Computer Science and Game Concepts

*It is expected students will:*

- Be able to accurately enter the code required to build the various game logic components involved
- Be able to make a map a wrapping map, and define its collision information
- Be able to identify fundamental eight directions of movement, as denoted by *x* and *y* values on a Cartesian grid.
- Be able to identify the coordinate systems of the screen, a master map, and how these interact with the size and positioning of the viewport
- Become familiar with the concept of a main character within the concept of viewport and scrolling bounding box
- Be aware of how *x* and *y* coordinates of a sprite can be used to explicitly define its location within the frame (Peg Registering)
- Identify how to generate copies of sprites dynamically within the game
- Create pre-defined movement paths for sprites
- Optional extensions: adding secondary weapons, adding enemy-dropped power-ups, adding a boss or new type of enemy to a level, making particle-based explosions, putting weapon upgrades within the game.

**Unit 6: Platformer**

The *Platformer* game is the final guided game project in this program. In this game, students will find all of the previous concepts integrated into a single project. The *Platformer* game also has the largest amount of code pre-written, so that it takes students less time to get the game running and on the screen. This game follows the previous three because it requires a higher level of maintenance to get all the components working and because the ñplatformingò effect relies on extra map information. Also, some of the core code is quite complex and out of the scope of an entry-level game programmer.

Computer Science and Game Concepts

*It is expected students will:*

- Understand the term ñgame engineò, and how all of the functionality of the game except for the content is derived from it
- Recognize the ways in which they can expand this game project within the scope of the game engineôs tools
- Learn to recognize the uses and value of logical states for triggering different actions, similar to artificial intelligence (State Machines).

- Optional extensions: adding a jet pack for the main character, giving the main character a weapon, placing the key on a movement pattern, creating more challenging ways to progress to the next level (such as finding a hidden door).

Production Art

*It is expected students will:*

- Continue their development of art design principles: balance, emphasis, contrast, movement, rhythm, pattern, variety and unity

## Instructional Components

- Direct instruction
- Indirect instruction
- Interactive instruction
- Independent, guided practice
- Modeling
- Group work ï principal component in assessment

## Student Expectations

- Ability to work cooperatively
- High level of classroom maturity
- Leadership in classroom activities, small group interactions
- Good Math skills
- Basic knowledge of computer operation

## Learning Resources

- Printed materials: DigiPen *Video Game Programming: Level 1, sample Game Design Document, Sample Post-Mortem document*
- Computer Lab
- Video projector
- Software: *ProjectFUN editor, graphics suite (Fireworks or Photoshop preferred)*

## Student Fees

- none

## Assessment

| **Learning Outcome** | **Value** |
|---|---:|
| **Introduction to Game Design** | 10 |
| **The Cage** | 5 |
| **Pong** | 5 |
| **Brix** | 10 |
| **Sidescroller** | 20 |
| **Platformer** | 20 |
| **Final Independent Project** | 30 |
| **Total** | 100 |

**Concept Quizzes and Class Work**

These assessments would be based on day-to-day progress in the course: meeting design deadlines and objectives, meeting the appropriate level of coding outcomes and maintaining pace with the class. Assessment techniques would include performance based assessment, completion checks, design portfolios, and in-class quizzes.

**Projects**

The bulk of the grade for this class would come from the proper demonstration of learning outcomes in product assessments. The students need to show that they have completed the specific learning outcomes for that module by the production of milestone projects, such as Pong, Brix, Sidescroller, and Platformer. For each project, students are required to demonstrate learning by going beyond the basic code demonstrated in class to show evidence of self-sufficiency with coding and design aspects.

**Final Project**

The final assessment for this class is a self-inspired, self-directed work that will take approximately 30 hours to create. The students create all art, code, and design themselves, truly demonstrating their abilities and their learning via product-based assessment. These games can be distributed to and played by anyone who runs a Windows-based computer.