Darryl Runyan
Arts, Digital Media, and Communications Division
Revised Spring 2023
Implemented Fall 2023

COURSE OUTLINE
Game Design 2

Course Description
ID 218. Game Design 2. 3 credit hours. Prerequisite: ID 118 with a C or better. This course will enable the student to design, create, and deploy, using the Unity 3D game engine, complex video games and interactive experiences. The student will use the tools needed to create advanced interactive 3D experiences, through 3D manipulation and the creation of state machines.

Required Materials
For complete material(s) information, refer to https://bookstore.butlercc.edu

Portable storage device (Portable hard drive recommended)

Personal earbuds/headphones for lab use

Butler-Assessed Outcomes
The intention is for the student to be able to do the following:
1. Design and create competitive game levels using the Unity 3D game engine.
2. Create and use state machines and advanced scripts to create interactive systems.

Learning PACT Skills that will be developed and documented in this course
Through involvement in this course, the student will develop ability in the following PACT skill area(s):

Technology Skills
- Discipline-specific technology - Through the use of the Unity 3D game engine, the student will use design and scripting skills to an advanced level.

Major Summative Assessment Task(s)
These Butler-assessed Outcome(s) and Learning PACT skill(s) will be demonstrated by the following:
1. Creating and scripting a space station game.
2. Creating and scripting a multiplayer game.

Skills or Competencies
These actions are essential to achieve the course outcomes:
1. Create advanced stand-alone and multiplayer games.
2. Script advanced stand-alone and multiplayer games.
3. Deploy advanced stand-alone and multiplayer games.

Learning Units

I. Animation
   A. Mecanim state machine
   B. Importing animations
   C. Creating animations
   D. Playing animations

II. Setting up a state machine
   A. Creating C# state classes
   B. Creating a static game manager
   C. Moving between levels

III. AI Navigation
   A. NavMesh
   B. NavMesh agents
   C. Patrol scripting
   D. MoveTo scripting

IV. Building the state machine logic
   A. Adding GUI elements to control states
   B. Handing control off to the current state
   C. Interfaces

V. Physically-based rendering
   A. Baking lightmaps
   B. Image-based lighting and HDRI maps
   C. Final gather

VI. Holding objects
   A. OnTriggerEnter
   B. Instantiate
   C. Creating game objects with scripting

VII. Shuriken particle system
   A. Lights
   B. Noise
   C. Rendering
   D. Sub-emitters

VIII. Scripting user interfaces
   A. User Input
   B. Sliders
   C. Programmatic creation of UI

IX. Presentation
   A. Adding juice

B. Best practice design
C. Exposing quality settings
D. User-configurable controls

X. Controller Input
A. Local multiplayer
B. Rewired
C. Split-screen cameras

XI. Saving player progress
A. WWW Variable
B. Asset bundles
C. PlayerPrefs
D. Data serialization

XII. Procedural generation
A. Parent classes
B. Hexagon maps
C. Marching squares

XIII. Advanced AI
A. Finite state machines
B. Goal-oriented action planning

XIV. Intelligent level design
A. Creating a history of your level
B. Adding choke points

XV. Industry expectations
A. Resume requirements
B. Portfolio generation
C. Entry-level jobs
D. Networking

Learning Activities
Learning activities will be assigned to assist the student in achieving the intended learning outcomes through lectures, class discussions, team research, individual research, readings, viewing tutorials and study material, quizzes, tests, and other activities at the discretion of the instructor. These activities may be either face-to-face or online.

Grade Determination
The student will be graded on the learning activities and assessment tasks. Grade determinations may include the following: class participation, projects, team and individual participation, research assignments, quizzes, tests, and other methods of evaluation at the discretion of the instructor.