

31263 / 32004

# Introduction to Game Development

## Assessment 3 and 4

## Recreate a Classic Game

### Overview:

Assessment 3 and 4 are two parts of a linked assessment. In these assessments you will recreate a classic Atari-era game (from the list below) by developing nearly all the assets yourself. You will then innovate on the game to push your self-study skills and explore new game development topics on your own. These are **individual assessments** and it is expected that all the work done here is your own. Plagiarism checking software is used to check both the projects and the reports, as well as being evaluated by your ability to talk through your project during your demo.

### Primary Design and Development Constraints:

Regardless of the game that is chosen, there are some important constraints that must be adhered to:

1. The game must be made in 2D, using sprites.
2. Your final game (in Assessment 4) will have at least three scenes – one that is for the main menu, one that is a recreation of a level from the original game, and the last that shows off your design innovation. You have more scenes in your project structure if you wish.
3. All visual assets must be entirely your own creation.
4. You must have sprite animations in your game, even if there are no animations in the original game. You may also create your own particle system effects with 2D sprites.
5. Audio must be included for all major interactions in the game but the audio clips do NOT need to be your own creation.
6. All scripts in the game must be your own creation. You may use learning resources such as the Unity manual, forums, Unity Answers, and video tutorials, but you must not use any downloaded scripts from the Unity Asset Store, GitHub, or similar.
7. You may NOT use the Unity Rigidbody physics functionality, so all motion must be coded by you as frame-rate independent continuous movement (see Week 5 lecture) (unless you choose Tetris from the list below)

## Game List:

You may choose any of the games below to recreate. In recreating them, you are looking to have the same gameplay experience and functionality, though they should have more modern visuals, audio, and controls schemes.

Each game has an associated percentage level which represents the maximum grade in both Assessment 3 and 4. These are determined based on the relative difficulty of developing the corresponding games.

Each game also has some key constraints that must be followed. Otherwise, you should attempt to recreate the functionality of the original game as closely as possible and state any assumptions (such as version of the game or understanding of certain game functionality) in your report and demo.

### 80%

[Indy 500](#) – With two players. The cars always move at an increasing speed until they hit an object. The players can only steer left and right. Must include additional animations that are not in the original game.

[Missile Command](#) – With missiles following a randomly determined straight-line path. The player controls targeting with the mouse.

### 90%

[Tetris](#) – With randomly spawning bricks of the seven shapes in the original game. The bricks should move on a grid-based system (not continuous motion). Must include additional animations that are not in the original game.

[Asteroids](#) – With your own calculations for acceleration, combining multiple acceleration directions, and inertia. Asteroids should collide with and bounce off each other, with the size of the asteroid determining which asteroid is more affected. See “Constraints” regarding rigidbody physics.

### 100%

[Ms. Pac-Man](#) – With characters moving with continuous motion (not grid-based) and each color ghost having the following behaviours when the power-up pill has not been consumed.

- Green – Move randomly at each junction in the level
- Pink – Move clock-wise around the level
- Blue – Run away from Ms. Pac-man
- Red – Chase Ms. Pac-man

# Assessment 3

## Preparing your Assets and Planning Your Project

### Due Dates:

#### **Report and Unity Project Files:**

Due on UTSONline before Monday 2<sup>nd</sup> September

#### **Demo to Tutor:**

During your lab in Week 7

### Deliverables:

#### **Report:**

<student\_number>\_A3\_report.pdf

This must be a single PDF document, with the naming style mentioned here, with all of <student\_number> (including the brackets) replaced with your student ID. This will be submitted through TurnItIn on UTSONline, separate from the Unity project files below. The details of the contents of this report are specified below.

#### **Unity Project Files:**

<student\_number>\_A3\_project.zip

This is your entire Unity project folder, zipped up (as a .zip, no other format), with the specified naming convention. This will be submitted to UTSONline separate to the above report PDF file. The contents of this project for Assessment 3 are specified below.

#### **Demo:**

During your lab time, you will demo your project so far to your tutor in a one-on-one conversation. Your demo time will be a **maximum of 5 minutes**. You will need to talk about the work you have done so far in a way similar to the contents of your report. You do not need to prepare presentation slides.

## Task and Grade Overview:

In Assessment 3, you will be assessed on the first half of the semester worth of lecture and lab content by preparing **visual and audio assets** for your game, setting up and beginning to use your repository, specifying a novel design for your game in Assessment 4, and planning for your future development. In this way, you will demonstrate not only foundational knowledge and skills in development, but also your ability to analyse an existing game, decompose it a set of required assets and coded systems, and identifying your own knowledge gaps for future development plans.

Below is an overview of the grading, with an indication of what should be included in your Unity project and what should be written into your report.

Criteria	Details	Marks
Visual Assets	<p>You must create all of <b>your visual assets yourself</b>. You may use any tool you feel comfortable with. If you do not have much visual design background, it is recommended that you use <a href="https://www.piskelapp.com/">https://www.piskelapp.com/</a>.</p> <p>You do not need to recreate the assets exactly as they appear in the original game, you may use your own style. These also do not need to be works of art (we are the Faculty of Engineering and IT after all).</p> <p>However, there should be a created sprite/<b>animation</b> for each one that appears in the original game (and in some cases more) and it should be clear what each of your sprites represent. You will be graded on your ability to have an appropriate sprite and animation created to properly visuals the premise and activity within the game. Some visual assets that you should consider are:</p> <ul style="list-style-type: none"> <li>• Character (player and enemy) sprites and animations.</li> <li>• Object sprites (e.g. projectiles, obstacles, collectables, etc.)</li> <li>• Effects sprite animations or particle systems (e.g. explosions, collisions, movement accentuators, etc.)</li> <li>• Level boundary and platform sprites</li> <li>• Level background sprites</li> </ul>	3

	<p><u>Project:</u> All visual assets should be imported into Unity and animations should be established so that they can be previewed in the Animation Window.</p> <p><u>Report:</u> Show all sprites that you have created and give a 1-2 sentence description for what each of them is used for in the game.</p>	
Audio Assets	<p>You must source all audio that you will need for your game. You do not need to create the audio yourself (but can if you want to). You should only use “Royalty Free” or completely free audio from the internet and not copyright protected audio. Some of the audio files that you may need include:</p> <ul style="list-style-type: none"> <li>• Menu background music</li> <li>• Game background music</li> <li>• Sound effects for when objects move</li> <li>• Sound effects for when objects collide</li> <li>• Sound effects for when characters are destroyed</li> </ul> <p><u>Project:</u> All audio assets should be imported into Unity so that they can be previewed in</p> <p><u>Report:</u> List all the audio files you have sourced and a 1-2 sentence description for what each of them are used for in the game.</p>	3
Recreated Level Scene	<p><u>Project:</u> With the created visual assets, create an initial layout for the recreated level. This can be the starting state of the level or an example of the level during play to show how each visual asset would be placed in the level.</p> <p><u>Report:</u> Provide a screenshot of the level so far and a 1 paragraph description of what you will need to do</p>	3

Initial Movement Development	<p><u>Project:</u> Some form of movement should be in your demo scene. This should not be the player controlled movement or register collisions between object (as these are covered after the assessment due date). It should include example movement of the player's character (without input controls), enemies, projectiles, or other objects.</p> <p><u>Report:</u> Provide a 1 paragraph description of how the movement in your game has currently been coded.</p>	3
Project Organization	<p><u>Project:</u> The Project Window in Unity should have a logical and organized layout of folders and sub-folders. Folders and assets should be appropriately named such that other potential team members could easily navigate that project and find the asset they are looking for.</p> <p>The Hierarchy Window of the Recreated Level scene should also be well organized, with use of parenting to form groups of gameobjects and each gameobject with an appropriate (short but clear) name.</p>	2
Git Repository	<p><u>Project:</u> You must setup a Git repository, hosted on Bitbucket, GitHub, or GitLab. You must use a <a href="#">Git Workflow</a> with at least a Master, Development, and multiple Feature branches (you may use a more in-depth branching strategies if you wish). You should begin to use this repository for Assessment 3, and then use it extensively in Assessment 4.</p> <p><u>Report:</u> Screenshot and paste the Git commit history / branching snapshot plus a paragraph describing your branching structure.</p>	2
Plan for Coded System	<p><u>Report:</u> Provide a list of the major functionality within the game that you believe you will need to code to recreate the classic game. Provide a description of each of these from a design perspective – i.e. you do not need to provide</p>	2

	<p>sample code, rather talk about what the scripts will do overall in relation to the player experience. This may include:</p> <ul style="list-style-type: none"> <li>• Movement</li> <li>• User-input</li> <li>• Object behaviours (characters, weapons, etc.)</li> <li>• Game rules and resources (e.g. health, ammo, scoring, win/lose checks, etc.)</li> <li>• Results of collisions</li> <li>• Playing of animation and audio</li> <li>• User-interfaces</li> <li>• Menu screens</li> <li>• Etc.</li> </ul>	
Design Proposal	<p><u>Report:</u> Provide a maximum of 2-pages describing how you plan to innovate on the classic game with your own design. This design should change the gameplay experience, such that it is still identifiably similar to the original game but with a unique twist.</p> <p>The design changes should also show that you will need to conduct self-learning to achieve your objective and that you are aware of where your knowledge gaps are. You should describe what skills you will need to learn to implement the design change and how these skills will benefit your knowledge of and future in games development overall.</p>	2