

# BIRDMAN – FEEDBACK



## PRACTITIONER LEVELS

Talented

Skilled

Accomplished

Capable

Ineffectual

## GRADING

## TEAM: BIRB

### GAMEPLAY EXPERIENCE

- REPLAY VALUE

### TECHNICAL IMPLEMENTATION

- SCENE MANAGER

**PASS**

## COMMENTS

Well done; the game is very well presented and clearly a passion project. The graphics are nice and the way the scene plays out leads the player through a story. You need to incorporate clearer feedback to the user during the preview nights. At the moment they play out as interesting stories, but the gameplay is shallow.

With codebases, make sure they are not unnecessarily complex or obtuse to read. There is some extensive use of advanced features, but at the same time the game “feels” like it should be more data drive. Looking forward to the next game.

**GENERAL FEEDBACK**

<b>GAMEPLAY</b>	
101	Well done on creating an enjoyable Birdman inspired game
102	The game plays well but requires more innovation or depth
103	The game lacks replay value
104	Transitions between scenes work well
105	The game is buggy and/or difficult to play
106	The game is solid but requires more attention to detail
107	The menu system does not work
108	The progression or scoring system has not been implemented
110	The game or the UI lacks polish
112	It lacks gameplay testing and/or balancing
113	It is not possible to restart the game

<b>TECHNICAL</b>	
201	You should be pleased, there's lots of solid design and code decisions
202	Functions are short and designed well to reduce code complexity
203	OOP has been used well throughout the game's design
204	There is good usage of composition
205	The coding standard has been adhered to
206	Const correctness has been used for both functions and objects
207	The STL has been used effectively
208	Top use of dynamic memory and in particular smart pointers
209	The codebase is reasonably well structured, but could be improved
210	Data is not being properly encapsulated
211	Memory has been misused and/or there are leaks
212	Functions and conditionals could be made more succinct

213	Make use of keywords default and delete
214	The scene manager shows strong design decisions
215	Inherited classes do not have virtual destructors or use override
216	There is a general need to improve OOP design
217	The coding standard has been misused
218	Make sure to pass larger data structures by reference
219	The STL containers usage is not correct or non-optimal
220	The state system responsible for managing the game needs work