

Project Development Proposal

For

Cow Panic!

Instructor: **Marcello Balduccini**
Team Members: Drew Graham, Jake Carfagno, Josh Karmel, Hugo Armella,
Dan Sipe, Jonah
Cycle: 1
Date Submitted: 4/6/16

Document template copyright 2015, CCI Faculty

Grading Rubric – Project Development Proposal

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading.

Achievement	Minimal	Exemplary	Pts	Score
Project Request			10	
Scope of Effort	Students need to describe (3) levels of scope with a value of 10 points each.	Clear distinction between the (3) different scope levels with details.	30	
Approach	Technical, 5 points Management, 5 points		10	
Major Deliverables	Students must describe a minimum of (3) the critical tasks for success, 10 points. Students must identify and describe risks that relate to their project, 5 points.	Students describe a more than (5) critical tasks for success, 10 points.	15	
Plan and Schedule	At least 30 tasks listed in their plan.	At 35-50 tasks in their plan all with dates and resources assigned.	20	
Grammar and Spelling	Many serious mistakes in grammar or spelling	Grammar, punctuation, and spelling all correct	10	
Tone and Presentation	Difficult to follow, but can be understood. Deduct 4 points if submitted document still contains items related to the template. (< fields >, Word comments)	Clear and concise. A pleasure to read.	5	
Late Submission	Deduct 10 points if 24 hours late. Deduct 25 points if 48 hours late.			
Total			100	

Project Development Proposal

Project

Project Request

We would like to make a game called *Cow Panic!*, a game to be played by casual gamers. There will be no funding necessary; we will be working in Unity, GameMaker Studio, and SourceTree, which are all free applications. The game will be made in Unity, with scripts written in C#. Artwork, which is currently limited to pixel art, will be made in the GameMaker Studio sprite editor. If we need to make anything beyond pixel art, we have Photoshop available to us. Our work will be shared in a repository managed through SourceTree. We plan to have the game finished by Week 9, which gives us an extra week of time if more work is necessary. In our original scope, we had planned to make the game Cow Panic for Android devices; however, we have changed our scope so that we are developing the game to be played on PC, with an Android port included as a stretch goal.

Scope of Effort

Critical Priority:

We will deliver a functioning, easily playable game that can be run on a PC. The game will have animated sprites for different types of cows (however many there end up being), as well as a pixel art background. It will also have retro 8-bit music playing in the background, and a heads-up display that includes the player score and life count. We will also improve the abduction script to make it function properly, and implement a "ground" for the cows to walk on so that when the tractor beam begins to pull a cow up but then disappears before they reach the UFO, the cow will fall to the ground and continue walking to the right.

“Nice to Have” Level:

We would like to include many different types of cows (chocolate cow, sickly cow, strawberry cow, soy cow, coconut cow, golden calf, etc.). This will be very easy for the pixel artist, but implementing them may not be as easy. We will try to break the code into modules as we are learning in CS-172 to make this process easier. We would also like to make a main menu that includes the sub-menus Play, Options, and Quit. If the artist has enough time, it would be nice to have some more music for the game (maybe menu music and gameplay music). Sound effects for things such as ambient cow noises (ex. *moo*) and the tractor beam extending and retracting (ex. *nyomomomomo vrrrrm*) would be great as well.

Future Consideration:

A port to Android will likely be difficult, but we would really like to make it happen. We aren't experienced in developing for Android in Unity, so we don't know what it entails. The likelihood of producing an Android version of the game by Week 10 is slim, but we would like to work on it in the future beyond CI-103. Porting the game to Android will probably take at least 1-2 weeks, when taking into account the time required to learn the technology and implement what we learn into a port. If we do manage to bring *Cow Panic!* to Android, it would increase our player base significantly. The portability brought by Android would allow players would be able to enjoy the game in the best possible setting: when they are bored (on the subway, waiting for food, etc.)

Approach

Technical:

Development will take place in Unity and GameMaker Studio. Unity is the game engine that we will be designing the game in. In Unity, we will write scripts in C# that will interact with game objects to make the game function (ex. the CowSpawn.cs script makes cows spawn at a steadily random rate and move to the right). Gamemaker Studio is also a game engine; however, we are only using it to make sprites with its sprite editor.

Management:

The art for the game (mostly consisting of pixel art) will be done by Josh Karmel and Hugo Armella. However, if it proves to be too much work for them, I will lend a hand. I have experience in Adobe Photoshop and Illustrator, so if my skills are necessary I will gladly help out. The development (programming and Unity) will be done by myself (Drew Graham), Jake Carfagno, and Dan Sipe, while Jonah will take care of quality assurance.

Major Deliverable

The first few tasks will apply to the gameplay, rather than the game as a whole. We have to create a ground for the cows to walk on. Its secondary function will be that when a cow walks into the tractor beam and start to move up, but then the beam disappears, the cow will fall back down to the ground and continue moving forward. We also have to implement screen borders, so that when a cow walks to the right and off the game screen, it will be deleted from the game space to save memory. Our next tasks are to begin creating the HUD, the main menu, and a score saving mechanism. The work required for the main menu consists of creating options for the user to choose (Play, Options, Quit) and sound effects for when the user selects an option. After the gameplay is mostly functional, the final task is to test the game for bugs.

Plan and Schedule

CI 103 Project Plan

