

# Games Gone Wild

by Curry Fury Games

## How to run from Unity

Simply load the scene named “SplashScreen” and run the game. Nothing else needed.

## Trade-Offs

Not implemented:

- Drunk Filter. Originally we planned on putting a filter over the player’s field of vision which would indicate how drunk the player was and thus how close they were to losing.
- Credits - Have information about each person and how they help out in building the game. Also have favorite type of drink.
- Settings - Setting to change volume/mute for music, and game sound.
- Pause Menu - pause the game and pops up a window that will have setting, go to menu.
- Practice mode stars - earn stars for beating a certain set of level. Have 5 stars maximum you can earn
- Ads - Ads pop up every 3 set games.
- PlayStore / AppStore - Planned to have integration to show high scores.
- HighScore - Working on desktop but was broken on mobile at time of turn-in.

## Overall Design

We didn’t plan out any sort of design originally when starting the game. What this got us was a handful of different scripts for different scenes and no way to communicate between them. We had to fix this issue, and to do that we created a “Global Controller” that permeates through every scene in the game and manages values that are needed between mini-games and scenes such as high scores and where you are in the “rotation” of mini-games. This loosely follows the “Blackboard” design pattern, described as a “Generalized observer, which allows multiple readers and writers. Communicates information system-wide.” While probably not the most ideal way to handle the problem, it did solve it and we haven’t had too many problems because of it.

## Visual Effects

Beer pong cup rattle - Achieved through the use of Unity’s animation editor.

Beer pong ball shot - This was a hard-coded animation that is simply a translation each step as well as a size increase and decrease. The size increases by a hard-coded amount until the distance reaches half-way to the target, and then it decreases. The logic is handled by flags and timers.

Congratulatory text - At the end of most mini-games there is text that grows. This is handled manually, growing every few frames by a font size, but I would like to make this an animation eventually.

Save the Floors "Head" animation - On a hard-coded timer to simply change sprites.

Save the Floors "Head" rotation - Controlled with a randomized timer that goes anywhere from 0 seconds to 2 seconds long. During this timer, the head moves in one direction and then when it ends it chooses which direction to turn based on where it is in relation to 0 degrees.

Save the Floors "Puke", Flippy Cup, Dart - All based on the physics built in to Unity.

Arm Wrestling - We initially had still image and they would move left or right. We remade the arm in blender to make it 3D, and it came out really nice.

## Audio Effects

As for sound effects, we wanted to keep it to a minimal so it was not distracting nor annoying to the user while playing. As a group, we felt that some sort of playful background music would be appropriate for our game concept to keep it light and playful. All sounds effects were created by Colin. As we decided to create a collection of minigames for our semester project, we create each scene in 2D space so all sound effects are convey in 2D space.

Intro Music - elevator music that sets the mode.

Overall Music - Hardik asked his friend Garret Beelow. His website is <http://www.garrettbeelow.com/> and he made the music for our game.

Beer Pong - During our beer pong scene, sounds effects were added when the user either makes the pong into a cup or misses the cup but hits the edge of the cup.

Flippy Cup - During our flippy cup scene, when the user swipes up to initiate the movement of the cup, the sound effect of a cup being flicked is render. When the solo cup collides with the table for the first time, the same sound is convey to the user to add some sort of realism while playing the game. Initially, the sound effect was suppose to be activated when the solo cup and the table collided but after debugging, the collision time between these two object was so minimal that when the current sound effect was activated, the next collision would occur before the previous sound effect was finish activating another collision sound effect which then would give a muttering sound until the last collision occur.

Darts - When a user taps to keep the dart up, a "swoosh" noise follows, and when the user hits the dartboard it makes a sound.

Save the Floors - There are sound effects for throwing up, and landing in the trashcan.

## Facebook Integration

Facebook and Unity worked together to produce a Facebook SDK for Unity! Colin implemented this SDK and used the Facebook and Unity Developers' documents to write the code to use the Facebook API. So far, we successfully get the user's name and picture from Facebook. When a user loses to the play mode, his or her highscore is sent to facebook and then retrieved to

display his or her score along with friends' high scores. Also, a user can share his or her score in Play Mode once the game is over by click the "Share to Facebook" button. Future api calls still to implement are to retrieve a user's friends' pictures to display next to their high scores.