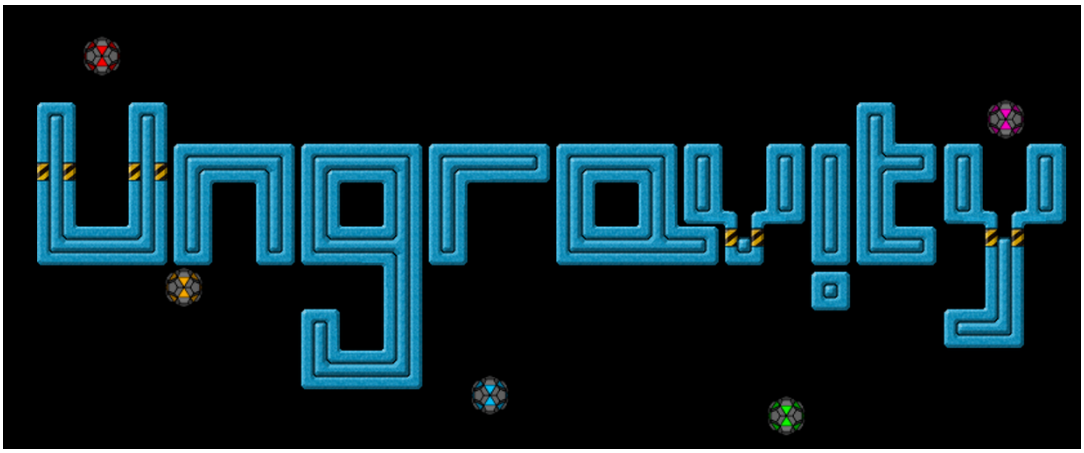
Author: David Curras
Version: 1.0



Version # 1.00

July 1st, 2013

Index

Introduction

Ungravity is an HTML5 casual game where the player has to lead the balls to the goals and collect as many stars as he can by changing the gravity of the world. Each level is a maze with at least one goal, many stars and at least one ball. There are many levels grouped by episodes, and each episode has its own graphics and difficulty.

Game Play

The player objective is to make the balls reach the goal collecting as many stars as he can. To do this, the player will click or tap in the playable zone of the game and it will change the gravity of the world randomly. For instance, if the gravity is pulling down the objects in the world, after click there is a ⅓ chance that the gravity starts pulling object to the left.
Each level is a maze with balls, stars and goals. There are many levels grouped by episodes, and each episode has special features. A fixed <amount of stars will be distributed in each level.
Collecting stars is not required to win the level, but stars increases the player score. One star means one point, and the score will be increased when all player balls in the level reaches the goal. If the player aborts the current game before win the level, the stars collected on that level will be discarded and points won't be added to the score.
The player can play the same level as many times as he wants, but only the best score for that level will be recorded. The total player score is the sum of the recorded score of each level.
The first time the player enters the game just the first level of each episode will be unlocked. When a player wins a level the next level will be unlocked.

Graphics

Ungravity is a 2D web game with a "top-down" perspective. The whole world map is visible as well as the stars, wall, goals and balls. The player can choose rendering the game in HTML5 Canvas mode or HTML5 and CSS3 without canvas. This option is available under "Main Menu" → "Options" → "Graphics".
The maps backgrounds were made with "Tiled Map Editor" and the sprite sheets for animations were made with the free version of TexturePacker. All images has PNG format and each tile has 32 pixels of width and height.
The game container default dimensions are 800 x 480 pixels (25 x 15 tiles), but the game will fit the Browser dimensions by automatically scaling the images keeping the aspect ratio.

Controls

The only action that the player can perform is changing the gravity. There is a control panel at the right of the playable zone with buttons to pause, play again, play next, play previous, mute and back to menu.

- ***Touchable Displays:***
  - Tap the playable zone to change the gravity.
  - Tap the control panel button to execute the action.
- ***Non Touchable Displays:***
  - Left or Right click in the playable zone.
  - Left or Right click on the button to execute the action.

Keyboard shortcuts such as P → Pause, N → Next, B → Previous, M → Menu and S → Toggle Sound will be added in the second release.

Plot

The game takes place in a near future where Dr. Ungravity is doing experiments with gravity. He has small boxes of many kinds of materials and colorful balls to perform his experiments. One day, Dr. Ungravity's child comes to the lab, finds the boxes and start playing with it.

To explain the plot to the players, some draws will be added at the beginning of the first level of each episode in the Release 2.

At the end of episode 4 the kid will break a box and the whole lab will be under the "Ungravity Experiment" effects. This will be the starting plot of Ungravity 2, a platform game that I want to develop in the future.

Languages

English is the only available language in the Release 1. Spanish will be added in Release 2 and the player will prompt for choose language when the game starts and also will be able to change the language from "Main Menu" → "Options" → "Language".

Assets

- **Tile Sets *(png images):***
  - Level Background. *path: assets/maps/tileset.png*
- **Sprite Sheets *(png images):***
  - Balls. *path: assets/sprites/ballanim.png*
  - Star. *path: assets/sprites/staranim.png*
  - Control Panel Buttons. *path: assets/sprites/controlpanel.png*
- **Menu Texts *(png images):***
  - Ungravity. *path: assets/texts/ungravity.png*
  - Play. *path: assets/texts/play.png*
  - Options. *path: assets/texts/options.png*
  - Credits. *path: assets/texts/credits.png*
- **Episodes and Level Thumbnails *(png images):***
  - Episode. *path: assets/thumbnails/episode[NUMBER].png*
  - Level unlocked. *path: assets/thumbnails/level[NUMBER].png*
  - Level locked. *path: assets/thumbnails/level[NUMBER]-locked.png*
- **Level Maps *(tmx files from Tiled):***
  - Level. *path: assets/maps/level[NUMBER].tmx*
- **Animation Scripts *(json JavaScript files):***
  - Balls. *path: assets/sprites/ballanim.json.js*
  - Star. *path: assets/sprites/staranim.json.js*
  - Control Panel Buttons. *path: assets/sprites/controlpanel.json.js*
- **Sounds *(mp3 and ogg files):***
  - Presentation. *path: assets/sounds/presentation.mp3 / .ogg*
  - Level starts. *path: assets/sounds/*start.*mp3 / .ogg*
  - Ball against Ball. *path: assets/sounds/ballCollision.mp3 / .ogg*
  - Ball against Wall. *path: assets/sounds/wallBounce.mp3 / .ogg*
  - Star collected. *path: assets/sounds/star.mp3 / .ogg*
  - Ball reaches the goal. *path: assets/sounds/win.mp3 / .ogg*
- **Fonts *(woff files from Google Fonts API):***
  - Quantico. *link: quantico.woff*
  - Permanent Marker. link*: permanentmarker.woff*
  - CSS API Link link: *Google API Fonts*

Project Schedule

Release: 1 **Year:** 2013 **State:** Done **Deadline:** July 1st

Learning and Researching *(2 weeks)*: *May 14th to May 26th*
- **Tiled:** half day
- **TexturePacker:** half day
- **LimeJs:** 4 days
- **Box2d:** 5 days
- **Closure Library:** 2 days

Update Box2d JS under Closure *(1 week)*: *May 27th to June 2nd*
- **Researching:** 1 day
- **Development:** 5 days

Level 101 development *(1 week)*: *June 3rd to June 9th*
- **Create private git repository for version control:** 1 hour
- **Level Design in Tiled:** 1 day
- **Show the level map from tmx file using LimeJS:** half day
- **Add and show objects from tmx file using LimeJS:** half day
- **Add physics objects and handle collisions with Box2d:** 2 days
- **Detect collision between different objects:** 1 day
- **Change gravity on click / tap**: 1 day

Add game scenes and levels 102-104 *(1 week)*: *June 10th to June 16th*
- **Levels 102, 103 and 104 Design in Tiled:** 1 day
- **Create Main scenes in LimeJS (including artwork):** 4 days
- **Implement levels 102, 103 and 104 from tmx files:** 1 day

Add control panel and finish all levels *(1 week)*: *June 17th to June 23rd*
- **Create Control Panel in Tiled:** half day
- **Add Control Panel to Levels 101-104 in Tiled:** half day
- **Create levels 105-109 and 201-209 in Tiled:** 3 day
- **Implement Control Panel in levels 101-104 from tmx file:** 1 day
- **Implement levels 105-109 and 201-209 from tmx files:** half days
- **Handle control panel actions:** 1 day
- **Create Options and Credits scenes in LimeJS:** half day

Sounds, documentation and publish *(1 week)*: *June 24th to July 1st*
- **Create GitHub public repository and commit:** 1 hour
- **Create Documentation (Spanish Version):** 1 day
- **Create Google Doc and translate Spanish Documentation:** 1 day
- **Get Art Assets from Renzo Gustavino and Implement:** half day
- **Register domain, assign DNS, do SEO and publish the game:** 1 day

Release: 2 **Year:** 2013 **State:** Non scheduled

Release 1 - Features out of deadline

- **July 2nd**
  - Basic Documentation
  - Canvas / HTML render switch option
- **July 3rd**
  - Save user preferences in cookies
  - Improve maps background
- **In progress**
  - Add Spanish texts and language selection
  - How to play scene
  - Full Documentation

Release 2 - New features

- Architectural improvements
- Episodes 3 and 4
- Plot illustrations and scenes
- Users ranking
- Back-end features:
  - SQL Data Base
  - Users CRUD features
  - User authentication

Udacity Classes

Unit	Used	From library	Description
Canvas	**X**	**X**	Using LimeJS to handle canvas / html renders
Atlases	**X**	**X**	Using LimeJS to parse atlases
Input	**X**	**X**	Using Closure Library for click / touch events
Entities	**X**	**X**	My own implementation (must improve)
Physics	**X**	**X**	Using Box2d for physics calculations
Sound	**X**	**X**	Using LimeJS for handling sounds
Asset Loading	**X**		My own implementation