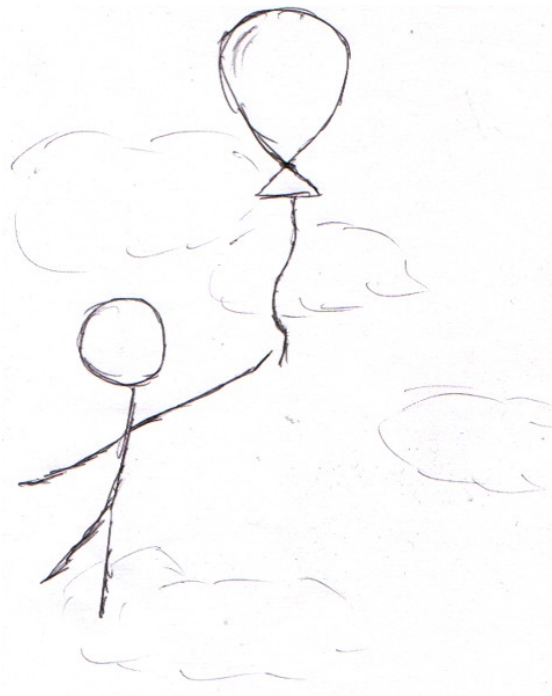


# Balloon Balloon

(can you swing on the moon?)

*Game Design Document*



*Balloon, Balloon*

*Can you swing on the moon?*

*Can you catch all the stars in the sky?*

*Can you keep up the pace*

*In this vertical race?*

*Can you float like a cloud up on high?*

# Contents

Concept.....	3
Abstract.....	3
Theory & Design.....	3
How It Works.....	3
Development Plan.....	4
Time Scale.....	4
Development & Delivery Formats.....	4
Considerations.....	4
XNA Game Studio 3.0.....	4
Silverlight.....	4
Game Design.....	5
The Environment.....	5
Gameplay Elements.....	5
Player.....	5
Balloons.....	5
Clouds.....	5
Stars.....	6
Moon.....	6
Winning & Losing.....	7
Scoring.....	7
Controls.....	8
Moving.....	8
Jumping.....	8
Code Issues.....	9
Physics.....	9
Custom Classes.....	9
BalloonController.....	9
Balloon.....	9
StarController.....	9
Star.....	9
CloudController.....	9
Cloud.....	9
Moon.....	9
Graphical Issues.....	10
Graphical Style.....	10
Required Art Assets.....	10
References.....	11

# Concept

## ***Abstract***

Balloon Balloon is a simple fun game in which players must leap from one floating balloon to another to see how high they can get.

## ***Theory & Design***

While books can bring us into the narrative and experiences of a person, and video can show us fantastic imagery we couldn't otherwise experience, it falls to interactivity to allow us to experience feelings and emotions we couldn't normally experience – and what experience is more common to all humans than that of the wish to feel the freedom of floating care-free through the air, like a balloon?

Part of the imagery of balloons is always this innocent, childish freedom. We like to release them, watch them float away, or carry them and see them above us. With Balloon Balloon, we seek to recreate some of that child-like fun and freedom.

Balloon Balloon will not be a tense or highly competitive experience. It will not be edgy or dynamic, and the player's pulse won't race, and they won't break a sweat.

There are multiple scoring mechanisms to Balloon Balloon, and no specific end conditions other than failure in the single player game.

Instead, players should feel a relaxing sensation of freedom and fun, and this should be reflected in all areas of the game.

## ***How It Works***

Multiple balloons fly up in a staggered and slightly random pattern behind the players.

Players use their controller to move left and right on the ground, and to jump for a balloon. When the player's character touches a balloon string with their outstretched hand, they grab it.

Hanging from a balloon, however, will slow the balloon's ascent. To ascend faster, the player must jump from one balloon to another.

# Development Plan

## ***Time Scale***

The aim will be to develop the game to a publishable state within 6-8 months, but there are no major restrictions on the time for development.

- 1. Model Balloon System – 1 months**
- 2. Model Player System – 1 month**
- 3. Model Player/Balloon Interactions – 1 months**
- 4. Model Clouds – 2 weeks**
- 5. Model Stars – 2 weeks**
- 6. Model Moon/Scoring System – 2 weeks**
- 7. Final Build Polishing/Bug Fixing – 1 month**

This time scale leaves plenty time for other work.

## ***Development & Delivery Formats***

### **Considerations**

While I would like to see a Flash version of the game developed at some point, the single delivery point and non-transferable code/asset format prevents it from being a viable option. Perhaps once the game is delivered on another format, Flash – or other in-browser code options – can be explored as a secondary delivery platform.

### **XNA Game Studio 3.0**

XNA Game Studio 3.0 allows for easy programming, and allows us to deliver the game on both PC and X-Box. For this reason, we will be developing in XNA Game Studio 3.0 on the PC.

### **Silverlight**

By coding for XNA and using Farseer Physics, there is a chance we could reduce the effort that would be involved in porting the game to Silverlight for use in browsers.

# Game Design

## *The Environment*

The play field begins with players on the ground. The sky behind them is a clear blue.

Balloons float upward in a basic pattern, with their position and frequency and position randomised just slightly.

The frequency of the balloons will decrease slightly as the players progress upward.

As players progress upward, clouds will appear, and then the sky will begin to darken, signifying them passing out of the atmosphere and into space, where stars appear.

Players can reach the moon, but the game does not end there. Players can continue upward if they please.

## *Gameplay Elements*

### **Player**

The players are represented by simple stick figures that are under the players' control. They can move left and right, jump and hang from balloons.

While hanging from balloons, players can swing from the balloon's string to move it slightly left or right.

While jumping left or right, they cannot change their direction.

There is a “skill shot” system for the jumping which allows the player to vary the distance that they jump based on their manipulation of the jump control.

The player's basic jump distance is equal to the distance between two adjacent balloons. The maximum jump distance should be the distance between three balloons.

If the player completes the parabola of a jump without grabbing a balloon, they begin to fall directly downward, gaining speed until they reach “terminal velocity”, at which point they will fall faster than the camera.

Should they hit the ground at this speed, or (more likely) drop off the bottom of the screen, the game is over.

### **Balloons**

Balloons float upward at a steady pace. This pace slows when you grab a hold of them, as it is determined by a weight/gravity algorithm, and the balloon is slowed by the weight of the player.

The balloons appear in a staggered pattern. The balloons in this pattern should be set apart by the distance of the player's basic jump. The pattern should be six balloons wide and repeat twice across this.

The position and weight of the balloon (and thus its rate of ascent) are randomised slightly to avoid too much repetition.

### **Clouds**

When the player reaches a certain height, they will come into contact with clouds. When a player hits them, they will dissipate in a little puff, and slow the player's ascent for a second.

## **Stars**

Once the player has left the atmosphere, they will see stars. Hitting a star will make the star “ting” with a musical and graphical effect, and will give the player a slight upward boost.

## **Moon**

Eventually, the player will reach the moon. The moon function like a giant balloon, only without a string. Players coming in contact with the edge of the moon will grab the edge like a balloon's string and hang.

## ***Winning & Losing***

There is no positive end to the single player section of Balloon Balloon – essentially, no way to “win” other than in multiplayer.

The single player game ends when the player reaches a fatal falling speed, at which point they will drop off the bottom of the screen.

The multiplayer game ends when all but one player has fallen off the screen.

When either game ends, a report is shown of the various scores of each player.

## ***Scoring***

While the main element of Balloon Balloon is to create a carefree environment to play in, previous experience has shown that games without a clear, obvious goal coupled with ways in which progress to this goal can be measured.

Balloon Balloon will feature multiple scoring methods, none of which can be considered the most prominent scoring method.

- Height  
The game tracks how high you have reached from the ground
- Balloons grabbed  
The game counts the number of balloons you've grabbed
- Clouds Hit  
The game counts how many clouds you've passed through
- Stars Touched  
The game counts the number of stars you've touched
- Swung On The Moon  
The game congratulates you if you've successfully swung on the moon.

## Controls

Move Left / Right  
Swing/Face Left/Right



Press - Begin Jump  
Release - Jump!

## Moving

On the ground, players can move left and right and jump.

Once hanging from a balloon, players can swing while facing left or right, and can jump from the balloon.

## Jumping

Jumping is a 2-stage process - holding the “jump” button will “charge” your jump, and releasing the button will begin the jump itself.

The power of the jump begins at a base level, and increases as the “charging” animation plays, but only to a “sweet spot” shortly before the end of the animation, after which the power of the jump decreases sharply. Releasing the button at the very end of the animation results in a jump that is only marginally more powerful than the starting power of the jump.

The exact position of the “sweet spot” is slightly randomised in each game cycle to prevent the game from becoming too predictable.

## **Code Issues**

### ***Physics***

The game will be coded using the Farseer Physics engine

### ***Custom Classes***

#### **BalloonController**

Designed to handle all the balloons in the game

#### **Balloon**

Consists of Bballoon and BalloonString, both physics object

#### **StarController**

#### **Star**

#### **CloudController**

#### **Cloud**

#### **Moon**

# Graphical Issues

## *Graphical Style*

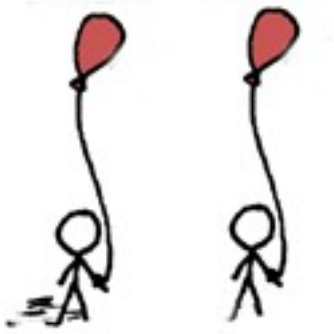
To reflect the innocent and carefree tone of the game, the art will be done in a hand-drawn style, with the players represented by stick figures.

## *Required Art Assets*

- Player - run cycle
- Player - turn cycle
- Player - jump charge-release (from standing)
- Player - swing cycle
- Player - hanging cycle
- Player - jump charge-release (from hanging)
- Balloon
- Star
- Star – ping anim
- Cloud
- Cloud – puff anim
- Moon

## References

- XKCD.com



- Zero Punctuation

