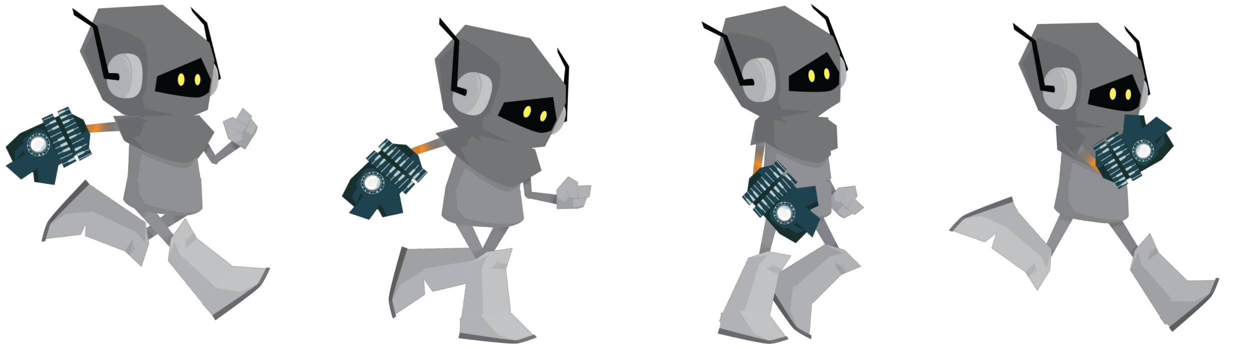
High Level Design Document

Basic Info

Game Name: Chromacore
Team Name: Team Bonanza
Overview: 2D musical platformer with dark, black & white aesthetics with crisp edges. The world progressively becomes more colorful and happy through successful gameplay.



Project Team Plan

All team members will act as Game Designers, in addition each member will have a more specific role. Though our individual schedules vary, all of us have Fridays off, and thus we will have weekly meeting Friday at 2PM to go over progress, assign weekly goals, and use as a working session. Additionally we will use the Sunday meetings of the Game Development Club as a working session as needed.

Jeff

Role: Programmer
Schedule: Mondays busy until 5:40. One class Tuesday from 11:45-1:25 and free after that. Wednesday free before 2:50 and after 5:40. Thursday same schedule as Monday. Fridays Off. So I can do between

Jen

Role: Artist
Schedule: Monday busy from Noon onward. One class on Tuesday, one class on Wednesday. Thursdays are busy all day. Fridays are free, except for work from 8pm-12am. I also have a freelance job which will occur intermittently (namely work on weekends). So, generally speaking, work for this class will be done on Friday day, weekends, and Tues/Wednesday night.

Deniz

Role: Programmer, music & sound
Schedule: Fridays off, one class on Tuesdays. Many assignments for classes due on Wednesdays. Rush week for my fraternity is from 09/22 - 09/29 and I will be stretched thin. I've been completing work ahead of schedule in order to compensate for this week.

Korbin

Role: Take point on music & sound, contribute to programming and art as needed
Schedule: Mondays Free after 5:40pm, Tuesdays free from 5-7:30pm, Wed/Thursday Free before 2:50pm and after 5:40pm, Fridays free no classes.

Vision Statement

Our vision for Chromacore is an industrialized, grody, dark environment, *initially*. Color, music, and happiness are parts of society that have been long forgotten; acid rain rules the skies. Teli, our hero, was born with an innate happiness - a longing for music and color has been his lifelong dream. After meeting an old, lonely hermit, Teli was given an amazing gift - a glove with the ability to return color and happiness this sad, gloomy world. Through our 2D musical platformer gameplay, Teli will share this gift with the world.

Chromacore's gameplay is a purely singleplayer experience on the home PC. Using exclusively the keyboard, players will simply control Teli's jumping (and, by extension, falling via gravity) to grab Notes and stay on the level, as well as control Teli's punch attack to destroy obstacles in his path. As Teli successfully picks up Notes and survives the hostile environment, his gift will progressively turn the world into a happier, more colorful environment.

Features of the Game

Core Mechanics

- **On-rails platforming** - The player is not given control of the character's horizontal movement. However, the player will control when the character jumps. This decision was made in order to sync gameplay with the audio feedback of the game.
- **Note Collection** - 'Notes' are collectibles placed in the level that the player must run or jump through to collect. Upon collision with a Note, audio and visual feedback is provided to the player.
- **Audio Feedback** - Gameplay is closely linked with audio feedback. Upon collecting a Note, a sound will be added to the background music.

Secondary Mechanics

- **Visual Feedback** - Gameplay is closely linked with visual feedback. Upon collecting a Note, the background will become more colorful in addition to an extra burst of color around where the Note was.

Tools

Development: Unity3D - Allows for easy world creation, frequent build runs, and rapid development of a working prototype
Art: Adobe CS5 - To create & edit art assets, Spriter (for animation), Texture Packer (to create sprite sheets)
Music: FL Studio, GarageBand - To create & edit music assets
Collaboration and File Sharing: SugarSync, Google Drive, GitHub - To simplify collaboration and asset management

Plan & Milestones

A. Development Process Model

We are going to use an Agile model in our development of Chromacore, specifically a method very similar to a scrum. However, we will not have daily meetings nor the formal roles assigned in a true Scrum. With crazy class schedules to balance and only 4 members, these constant check-ins are not really possible. Instead, we will be operating under weekly deadlines and meetings which essentially count as sprints, besides our meetings during class. Should any technical difficulties arise, we will make these known to everyone as quickly as possible. The weekly goals of an Agile model allow for fixing any of these issues as we develop, which makes it ideal. Because our development cycle is so brief, we need an easily adaptable production method so we can produce a playable product as quickly as possible. The first prototype will serve as our proof of concept, while the second will be a more polished and evolved iteration.

B. Milestones (by week)

Milestone 1: 10/4

Basic setup processes completed
Art: Character run/jump and item design completed. At least one background and some foreground elements completed. Effect animations started.
Programming: Basic game set up completed. On rails character movement, 2D orthographic camera following character, physics for running/jumping/falling, item pickups.
Sound: Music choice solidified and work has begun on properly cutting it apart for our purposes.
Design: Level Design sketches

Milestone 2: 10/11 - 10/18 - **First Prototype**

The first steps towards a fully completed game, which is why more time is budgeted here.

Art: Color transitions occur with item pickup. Item pick-up animation complete.
Art: More foreground assets developed and possible further development of background. Further tweaking of animations as needed.
Programming/Sound: Every action timed to the music. Score system in place.
Design/Programming: Timing is our key element to pin down in this game, so at this point, we should have a good idea of when and where actions should be occurring (or where we want the player to be taking specific actions). This means a close to finished level design, mapped out so we know when jumps, falls and path splits occur.
Sound: Music is properly cut up and integrated into the game.
Design: Begin user playtesting.

Milestone 3: 10/25 - **Second Prototype**
Essentially the finished game.
Art: All effects and animations are completed. Background work is finalized. Essentially, any and all art is done and integrated with the game.
Programming: Scoring is in place to properly correspond to item pickup/percentage of level colored, if not already developed in the past milestone. Further tweaks of game physics and timing of elements are implemented as needed.
Design: Level design finalized.
Design: Integrate results from user playtesting.

C.Task Dependency

As a note, most of these steps are multi-disciplined and require more than one expertise. For example, the visual feedback for the level changing color requires a programming element to actually implement the transition. The labels are just which discipline is more responsible for accomplishing the desired effect.

KEY

Programming (namely Deniz and Jeff)	Art (namely Jen)	Music (namely Korbin)	Design (everyone)

World Evironment created with character ---->	character and camera can move/be controlled --->	item pickup functions --->	Auditory Feedback (for item pickup)

		item pickup functions --->	visual feedback (for level changing color)
		item pickup functions --->	visual feedback (for pickup animation)
		item pickup functions --->	Scoring
		Character Run animation --->	jump/falling animation
Music is chosen/properly broken down --->	Level design (platform placement, jumps, items) --->	Jumps timed to music --->	auditory feedback (for jumps)
	Level design (platform placement, jumps, items) --->	Item placement/Pickup timed to music	
	Level design (platform placement, jumps, items) --->	Lose conditions implemented (ie falling off a platform)--->	Reset of level/checkpoints
	Level design (platform placement, jumps, items) --->	Foreground Platforms created --->	Implemented/randomly generated
	Level design (platform placement, jumps, items) --->	backgrounds created --->	Moving/properly implemented

Risk Management Plan

In order to create the appropriate audio, we need the individual instrument tracks, called stem files, from the songs we choose. We contacted the artists, but there is a high probability we won’t hear back from them. To minimize this risk, we have backup songs that we would rather not use, but we have the stem files available.

Another risk is that we will not be able to properly sync up the music to the gameplay. We will combat this by extensively playtesting.

Testing Plan

We plan to test every stage of development. First, once we have a barebones prototype, we will test basic platforming and jump controls to ensure that the character responds correctly. Once this is done, we will then build the next iteration for collectible items, and test to make sure that items correctly disappear when the player collides with them. Next comes matching sound to the objects. Once this is implemented, we will test to make sure that objects produce the correct sound on pickup. Finally, we will match the visual effects to ensure that the environment changes upon collection. We will test this feature to make sure it works properly. Once all of these mechanics are in place, we will then test to smooth out the gameplay.