

How to survive Ludum Dare

By Taro Omiya (Omiya Games)
Thanks to Jamey Stevenson (Spoony Bird)

Most Important: Have FUN!

- Making games is fun!
- Walk in with excitement and anticipation
- No point going into it worried
- Which leads to...

Second Most Important: Learn!

- Just started?
 - Learn how to use a tool
 - Learn how to work towards a deadline
 - Learn to complete a game
- Already experienced?
 - Learn a new tool
 - Make something new and different
 - Experiment!

It's OK to fail!

- Ludum Dare is NOT a competition
 - It's a shared experience.
- It's OK to not finish game by the deadline
 - There's always the next Ludum Dare
 - Or Global Game Jam on January
 - Or any other game jams listed in Compohub.net

Jam-specific Points

Rules

- Work alone or in a team.
- Create a game in 72 hours.

Compo-specific Points

Rules

- You must work alone (solo).
- Your game, all your content (i.e. Art, Music, Sound, etc) must be created in 48 hours.
- Source code must be included.

Time Management

- Time management is the most important skill in any game jam
- Best way to save time is preparation
- Know what takes up time, and plan accordingly

Plan to Stay Healthy

- Account for 16 hours (or 24 for Jam) you'll be sleeping
 - Can't emphasize this enough: SLEEP IS MANDATORY, not an option
- Account for 5 hours (or 8 for Jam) spent on eating
- Speaking of which, consider take-outs or instant lunches

Plan for Bundling

- Account for the 1 hour taken from compiling for all platforms
- Always submit at least 1-hour before submission time
 - Sometimes, websites gets bog down by Ludum Dare traffic, and uploading doesn't quite work

Total Time

- Compo:
 - Account for $16 + 5 + 1 + 1 = 23$ hours.
 - Total dev time: $48 - 23 = 25$ hours.
- Jam:
 - Account for $24 + 8 + 1 + 1 = 34$ hours.
 - Total dev time: $72 - 34 = 38$ hours.

How to be Prepared

- Know/learn what tools you'll be using for Ludum Dare
- There's a lot of resources out there!
 - http://www.reddit.com/r/gamedev/comments/18e38t/pocketknives_for_game_developers/

Tools to look for

- Game engine (Construct 2, GameMaker, Unity, etc.)
- Art tools (GIMP, Paint.net, Aesprite, Blender, etc.)
- Sound creation (BFXR, Audacity, etc.)
- Music composer (LMMS, Garage Band, etc.)

Back to Time Management

- I divide my development phases by half-days
 - First half-day is brainstorming and prototyping
 - Second half-day is feature implementation
 - Third half-day is level construction
 - Last half-day is polish
 - Usually, level design takes up the majority of the time

Phase 1: Brainstorming

- Enter with an open mind
 - Enter Ludum Dare *without* a game idea
- Why?
 - Your game should be based off of the theme announced the minute the event starts
 - That game you wanted to make forever will most likely NOT fit with the theme
 - Also, it makes you open-minded to other games, like walking simulators

Phase 1: Brainstorming

- Scope properly!
 - Time AND resources are short, so start small!
 - Don't make games like Halo, World of Warcraft, etc.
 - Angry Birds is fine. Heck, mobile games are the perfect scope for Ludum Dare
- Aim to make a demo, minigame or experiment

Phase 1: Brainstorming

- I don't recommend writing a design document
- Game idea should ALWAYS mention how the game is going to be played
 - E.g. "platformer", "twin stick shooter", or "control bunch of synchronized characters at once while taking advantages of their strengths"

Phase 1: Brainstorming

Method 1: Brain Dump Method

1. Open Notepad++
2. Set a timer to one-hour
3. Write out as many single-sentence description of a game as possible
4. After an hour, put my keyboard down, and select my favorite idea

Phase 1: Brainstorming

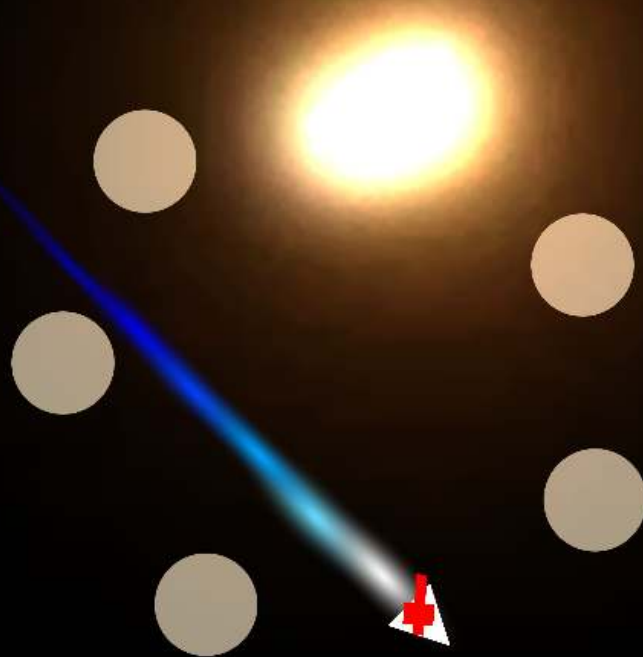
Method 2: Peter Molydeux Method

1. Ask a crazy, useless question
 - a. e.g. what will it be like to throw your own head?
2. Brainstorm on a game mechanic that attempts to answer this question

Phase 1.5: Prototyping

- Easily the most important 3 hours you spend in Ludum Dare!
- Learn to "fail faster"
 - Extra Credits video: <http://youtu.be/rDjrOaoHz9s>
- Basically, make a demo of your idea really, *really* fast
- Then play the demo, and see if it's fun

Phase 1.5: Prototyping



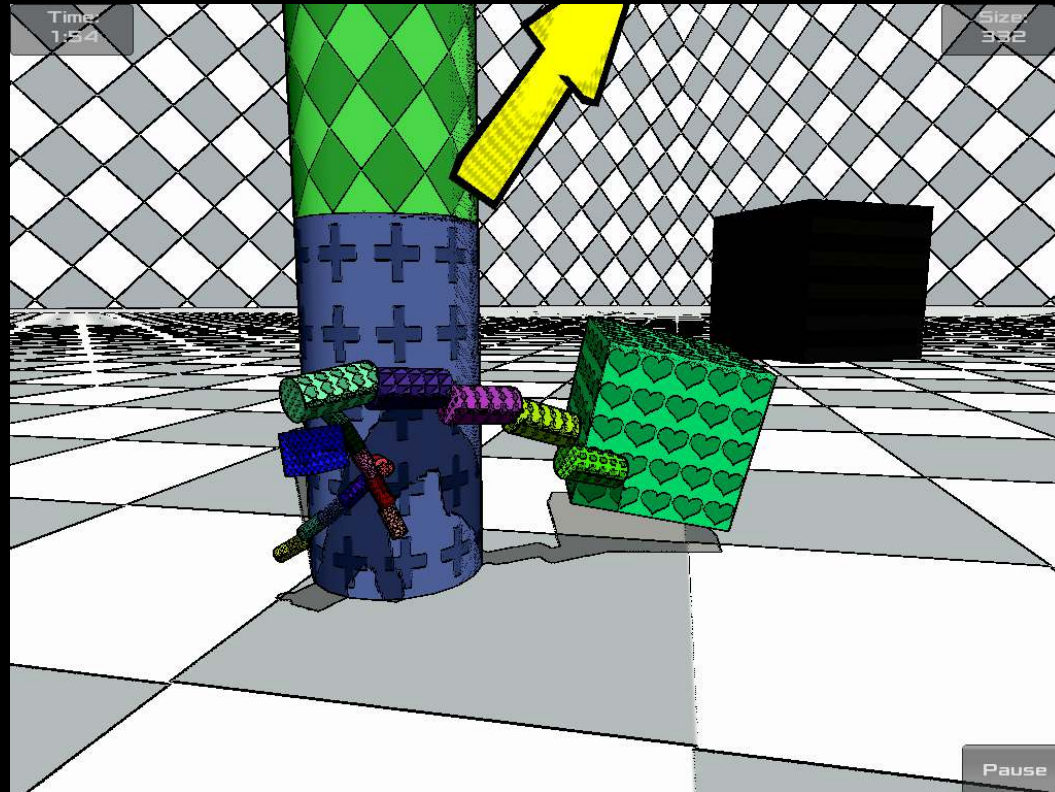
Phase 1.5: Prototyping



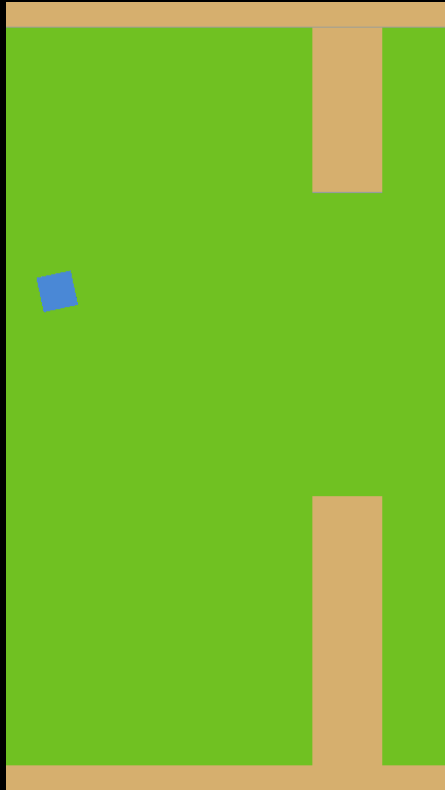
Phase 1.5: Prototyping



Phase 1.5: Prototyping



Phase 1.5: Prototyping



Phase 1.5: Prototyping

- Put something playable together fast!
 - Presentation quality isn't important
- Prototypes verifies whether your game is fun or not
 - Don't be afraid to throw out a prototype, and move on to the next game idea.
- Sometimes, the best games are created out of accidents!

Phase 2: Feature Implementation

- Write up a list of features to implement your game.
 - Recommend using a task tracker, like Trello
- Determine which features are going to be the most important
- Start implementing the highest-priority features within half-a-day

Phase 3: Level Construction

- For designing levels, I use pencil and graph paper
 - I usually sketch out either the aerial or profile view of the level, even if it's in 3D
- What is the "story/experience" you want to convey in this level?
- Design the middle and last levels first

Phase 3: Level Construction

For the first few levels:

- 1st level provides instructions on basic movement controls and complete a level.
 - Do NOT assume the player knows how to play FPS, platformers, etc.
 - Make the first level wide, simple, and focused

Phase 3: Level Construction

- 2nd level provides instructions on a game mechanic unique to this game
 - Also lets them practice movement a little more
- 3rd level is the selling point: "BAM! THIS is why you want to play this game!"
 - Make the first 2 levels short enough so they can get to this level within 5 minutes.

Phase 4: Polish

- Replace all placeholder sound effects and graphics
- Add juice!
 - Juice it or lose it: <http://youtu.be/Fy0aCDmgnxg>
- Adjust the level difficulty
- Bug fixes

Phase 4.5: Submission

- If it's fast, I recommend uploading to game portals
 - Itch.io, GameJolt, Kongregate, etc.
- If not, there's always Dropbox, Google Drive, etc.

Compo-specific Points

- Learn all the tools!
 - Focus on code, art, and music
- Super-easy to lose track of time
 - Pace yourself, and stay on a schedule
 - Setup a timer if you have to
- Always have graphics and sound
 - Placeholders are still better than no graphics or sound
- Still, only focus on polish on the last phase

Jam-specific Points

- Use game ideas that every team member agrees with
 - Best ideas are those that inspires more features to the game
 - Discuss the priorities of each feature
- Divide jobs based on category of work
 - e.g. one person focuses on programming, one on sound effects and music, and one on artworks

Jam-specific Points

- Make sure everyone is on-schedule
- Bookmark websites to get graphics, sound effects, scripts and music
 - Note that during voting phase, you must opt-out of a category where you used outside resources



Questions?

Stay tuned for list of super-useful
resources following Q & A

Game Engines

- Construct 2 (for 2D)
 - <https://www.scirra.com/construct2>
 - No programming!
- GameMaker (for 2D)
 - <https://www.yoyogames.com/studio>
 - GameMaker scripting language
- Unity (for 3D & 2D)
 - <http://unity3d.com/>
 - C#, Javascript, or Boo

Game Engines

- RPG Maker Ace Lite (for RPGs)
 - <http://www.rpgmakerweb.com/download/free-programs/rpg-maker-vx-ace-lite>
 - No programming!
- Ren'Py (for visual novels)
 - <http://renpy.org/>
 - Ren'Py's easy-to-learn scripting
 - Unless you want to muck around with its Python code...

Graphics (for 2D)

- GIMP (like Photoshop, all platforms)
 - <http://www.gimp.org/>
- Paint.net (like Photoshop, only Windows)
 - <http://www.getpaint.net/>
- Aseprite (great for sprites, all platforms)
 - <http://www.aseprite.org/>
- Krita (great tablet support, all platforms)
 - <https://krita.org/>
- MyPaint (great tablet support, all platforms)
 - <http://mypaint.intilinux.com/>

Graphics (for 3D)

- Blender (for...everything 3D, all platforms)
 - <http://www.blender.org/>
- 3DTin (simple 3D tool, online)
 - <http://www.3dtin.com/>
- MakeHuman (make humans, all platforms)
 - <http://www.makehuman.org/>

Audio

- BFXR (sound generator, anything Adobe AIR supports)
 - <http://www.bfxr.net/>
- Audacity (audio editor)
 - <http://audacity.sourceforge.net/>
- LMMS (music composer, all platforms)
 - <https://lmms.io/>

Jam-specific Free Resources

- Art (Kenney): <http://kenney.itch.io/kenney-donation>
- Art: <http://opengameart.org/>
- Sound Effects: <https://www.freesound.org/>
- Fonts: <http://openfontlibrary.org/>
- Music (Kevin MacLeod): <http://incompetech.com/music/royalty-free/>
- Music (DST): <http://www.nosoapradio.us/>