Game Jam Checklist

These are suggestions and may help as a sanity check before and during a jam.

A. Materials before the jam

1. Program installations (Game Creation tools, IDEs, Audio, Graphics).	
2. Extensions, USB drives/storage, headphones, adapters, cables, fans.	
3. Tablets, microphones, extra equipment like controllers or other inputs.	
4. Mobile devices, tablets or consoles as necessary.	
5. A (web) camera or mobile phone for photo & video documentation.	

B. Logistics on Friday

1. Form a team and pick someone who is project-responsible.	
2. Find a place for your team, sort out (wireless) internet access.	
3. Plus points if you sort out whiteboard access, team schedule and roles.	
4. Know where to find support and where to find food and supplies.	
5. Prepare or make it easier to submit (e.g. a project page on itch.io).	

C. Game Idea on Friday

1. Brainstorm/ideate on game ideas. Look over the brief and any supplements.	
2. Rapidly prototype ideas through paper, whiteboard or digitally.	
3. Evaluate your ideas and find ways to vote on them.	
4. Keep the idea simple, you can make it complex later. Or way, way later.	
5. Always have a backup plan. Have a good sleep schedule.	

D. Game Prototyping on Saturday

1. You should be well on your way with game creation.	
2. Check that people are well fed and drinking enough water (or choice of drink).	
3. If people have not slept Friday night, a nap or a walk may be good.	
4. Update the project page with title, description, names and roles.	
5. Take photos of your playtesting/progress and start forming your documentation.	
6. Game jams sometimes have some kind of activity/talk during the afternoon.	

E. Deliverable on Sunday

1. Create a zip file for the source, resources and print&play files as needed.	
2. Create an executable for public play for digital games.	
3. Update the project page with all the necessary details to get the game running.	
4. Upload screenshots/action shots on the project page.	
5. Upload source files and executable to the project page.	
6. (Optional) A gameplay video which is also uploaded (e.g. to YouTube/Vimeo).	
7. Submit before the deadline at (what usually is) 15:00!	

F. Last few things on Sunday (for local game jams)

1. Clear and clean up your area and work space. Leave things as you found it.	
2. Set up your game in the demo space. Usually there’s a space dedicated.	
3. One team member should host the demo, but make time so everyone gets to check out everyone else’s games at the game show.	

Practical troubleshooting tips

A quick reference of things to look out for during the course of the game jam.

A. Division of different tasks

(not to be confused as roles; everyone may contribute)

1. Coding/programming/version control
2. Artwork/assets including character design
3. Game Mechanics
4. Level/map Design
5. Music/sound and audio
6. Quality Assurance/feedback and user testing
7. Documentation/copywriter

B. When stuck or struggling

1. Check if there are other priorities, the details may be minor and disregarded.
2. Reassess if making it technically work is worth it, and how much time it will take.
3. Attempt to divide the problem into smaller ones, and among team members.
4. Look for external feedback and accept critique to change the design.
5. Look for creative alternatives or easier solutions and playtest them.
6. Take a short break and revisit the issue later. Napping, sleeping or eating may help.

C. Recommended habits

1. A shared folder to collaborate together on. Google Drive works fine, for example.
2. A form of version control (e.g. GitHub, Bitbucket, Mercurial, script, or manually copying).
3. Put up a schedule where everyone can easily see it. Preferably on a whiteboard.
4. Shared working area, and a quiet (maybe dark) isolated area for sleep.
5. Save early and often, playtest early and often. Take breaks early and often.
6. Build-test-repeat and be sure to always have something playable.
7. Rotate working hours so the work process and flow is more natural (e.g. when one coder sleeps another can take their place to avoid the need to merge code).
8. Rotate hours for assets and graphics only to keep sanity and sleep deprivation at bay.
9. Keep a positive vibe and remember the time dedicated to this is for fun.

D. Quick overview checklist

1. (Philosophical) What defines a game?
2. Do the core mechanics work?
3. Does it relate to or challenge the theme?
4. Is this a game of quality?
5. Is the gameplay fun?
6. Does it have good game flow, challenge and feel?
7. Is the game compelling or intriguing?
8. Has there been at least three different playtest sessions?
9. How long does gameplay last?
10. Is it understandable, and can players get it?
11. Does the game have music or audio?
12. Have you added juice?

E. Game specifics

1. (Recommended) A minimum resolution of 1024x768
2. Player controls, mechanics and gameplay work
3. Game challenge and game flow are balanced
4. Menus for starting and ending the game, and a help screen if needed
5. If the game ends, you should be able to easily start/retry the game
6. Audio (especially music) add to the game and it's presentation
7. External user testing or feedback needs to be conducted
8. The deliverable exists (optimally a zipped file for ease of download):
    a. A readme with a short description, credits/contacts, help to get started and links to any dependencies
    b. Executables and libraries
    c. Source files
    d. Screenshots (3 are optimal)
    e. (Optional) Presentation, Gameplay Video, Gamelog, Design document and other documentation